Mario R. Eden, Marianthi Ierapetritou and Gavin P. Towler (Editor)
Proceedings of the 13th International Symposium on Process Systems Engineering PSE 2018
July 1st - 5th, 2018, San Diego, California, USA.

Parallel, multi-scale, mechanistic model for high shear granulation using coupled DEM-PBM models on high performance computing systems.

Chaitanya Sampat[a], Yukteshwar Baranwal[a], Ioannis Paraskevakos[b], Shantenu Jha[b], Marianthi Ierapetritou[a] and Rohit Ramachandran[a,*]

[a]*Department of Chemical and Biochemical Engineering, Rutgers University, Piscataway, NJ-08901, USA*
[b]*Electric and Computer Engineering, Rutgers University, Piscataway, NJ-08901, USA*
**rohitrr@soe.rutgers.edu*

Abstract

A multiscale model combines the computational efficiency of a macro-scale model and the accuracy of a micro-scale model. It is preferred over a fully micro-scale model for its speed advantages while maintaining the physics of the problem. A less accurate way to perform such a simulation is touse data from a precomputed microscale model in a macroscale model. With the current cyberinfrastructure resources available, using more computationally intensive and concurrent multiscale models are morefeasible. This study proposes to use Discrete Element Method (DEM), a microscale model, and a Population Balance Model (PBM), a macroscale model, in a concurrent manner to model the granulation process of a pharmaceutical product inside a high shear granulator. The granulation between the components of a pharmaceutical blend is governed by the collision in between the particles. This leads to increase in their size, due to physical bonds in between them. The DEM provides the collision data while the PBM helps in predicting the macroscale phenomena like aggregation and breakage. This work attempts to couple these two models using a controller program, which triggers the DEM first, to give initial seed data to run the PBM. Then, the controller uses the data generated from the PBM continuously to determine the change in the physical properties and trigger the DEM from its last known state. The controller does the same with the DEM data to trigger the PBM. This occurs iteratively until a steady state is reached. A workflow diagram of the procedure followed is provided in Figure 1. The execution of each of the components is governed by a multilevel job scheduler which allocates resources rather than waiting for each simulation to run on a normal job scheduler on a cluster. The DEM is parallelized using Message Parsing Interface (MPI) while the PBM is parallelized using a faster hybrid approach which is a combination of both MPI and Open Multi-Processing (OMP). Since the DEM is computationally heavy, an algorithm is developed to utilize the idle cores during the PBM execution to run multiple instances of the PBM such that parameter estimation of the kernels of the PBM occurs on the fly as well. This method of using shorter bursts of each simulation led to faster simulation times as well as a more accurate model of the high shear granulator. The Quality by Design (QbD) approach is addressed using such a modelling framework and it also helps us understand the granulation process in a quantitative as well as in a mechanistic manner.

Keywords: Multi-scale Model, Population Balance Model, Discrete Element Method, High-performance computing, MPI

1. Introduction

Particulate materials are products or intermidiates of around 60% of the chemical industry(Ingram and Cameron, 2005). The modelling of these particulate processes pose a challenge when compared to modelling if uniform liquid systems. These solid systems usually comprise of solids which co-exist with various sizes and shape configuration, which have a large effect on the final composition and the performance of the product. This variation in the configuration of the product is unaccepatable by the pharmaceutical industry. It could lead to differences in the dissolution rates and bioavailability of the product thus, affecting the quality of the product. The unit process that helps control the size of a granule of the powder in the pharmaceutical product is granulation. During this unit operation, fine powders are converted to the larger particles by the process of aggregation. This occurs due to addition of a liquid binder, making the particles agglomerate into larger granules. This process is also referred to as wet granulation as a liquid binder is added. Due to high restrictions set by the Food and Drug Administration (FDA) these products undergo various batch rejections or have high recycle ratios(Sen et al., 2014). One of the major aims of the industry currently is to reduce this waste, thus creating a need for better modelling of this process.

The most popular way to model the granulation process is to use a population balance model (PBM). This model tracks the number of particles with certain set of charactersitics such as size, porosity and liquid content. It also takes into account the changes that may occur in these properties due to growth, breakage, consolidation, and other phenomena. This model is discussed in more detail in section 2.1. PBM fails to account for the particle level data, thus being inaccurate at times. This particle level data needed to improve the PBM can be obtained from discrete element methods (DEM). DEM help obtain the velocity of each particle and the number of times it collides with other particles by solving the Newton's laws of motion. This information is vital to the PBM as it helps develop a more accurate model with a higher physical significance. As these methods are complementary to each other, they are usually coupled whille building a model for the granulation process. These processes can be coupled serially like in **(cite current paper here)** or in parallel as discussed in this work. DEM is always executed at initiation and the particle level data is then transferred to the PBM. The DEM simulation is computationally very high as it needs to solve large amounts of simultaneous equations at every time step. The parallel execution of this type of coupling requires various instances of simulation of each of these models to be initiated, requiring high computational power. The most efficient way to make these simulations run faster is to take advantage of large number of CPU cores present on modern day high-performance computing systems (HPCs / supercomputers).

This work focusses on the building of physically accurate model for a high shear granulator. Section 2 discusses the model equation used and the method of parallelization implemented on the PBM and DEM to enable them to run on various cores of HPCs. The resulting particle properties from the coupling and the speed improvements achieved when the number of cores used for the simulation are changed, is discussed in section 3.

2. Methods

Figure 1 represents the workflow followed for the two-way coupled model that was used to model the high shear granulator. A two-dimensional PBM was developed for the solids present while a lumped model was used for the liquid and gas content. The PBM tracked particle flow, aggregation, breakage, consolidation and liquid addition. This PBM used the particle velocities and collision data from the DEM simulations making it mechanistic in nature. The DEM simulation was used to otain the particle residence time, velocities, inter- particle as well as particle-geometry collisions. These simulations were monitored using python scripts to switch in between the two simulations. The details of each of these components are discussed in this section.

2.1. Population Balance Model (PBM)

A detailed, mechanistic two dimensional PBM was developed in *C++*, which tracked the particles as they undergo the aggregation, breakage, consolidation and liquid addition. The governing population balance model is shown in Equation 1.

$$\frac{d}{dt}F(s_1,s_2,x) = \Re_{agg}(s_1,s_2,x) + \Re_{br}(s_1,s_2,x) + \dot{F}_{in}(s_1,s_2,x) - \dot{F}_{out}(s_1,s_2,x) \quad (1)$$

where, $F(s_1,s_2,x)$ is the number of particles with an active pharmaceutical ingredients (API) volume of s_1 and an excipient volume of s_2 in the spatial compartment x. The rate of change of number of particles with time in different size classes depend on the rate of aggregation $\Re_{agg}(s_1,s_2,x)$ and the rate of breakage $\Re_{break}(s_1,s_2,x)$. Also, the rate of particles coming into, $\dot{F}_{in}(s_1,s_2,x)$ and going out, $\dot{F}_{out}(s_1,s_2,x)$ of the spatial compartment due to particle transfer affect the number of particles in different size classes.

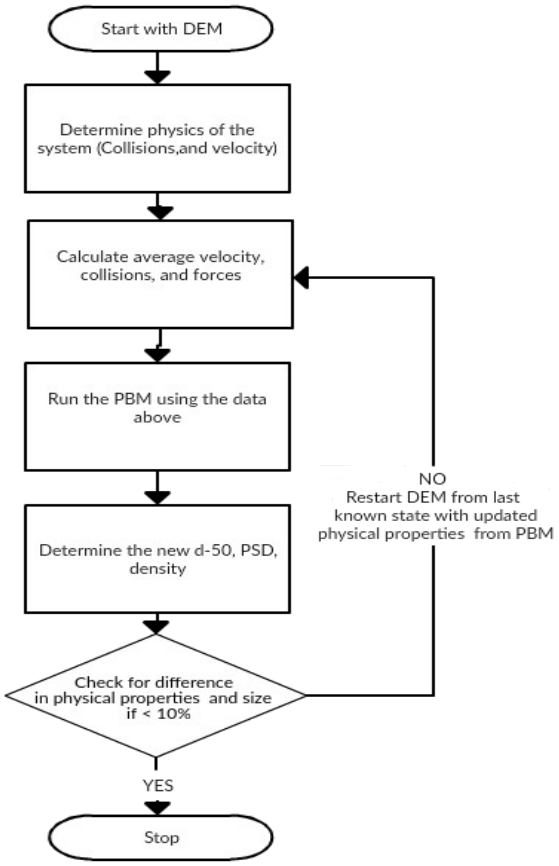

Figure 1: Schematic of the bi-directional approach used to model the high shear granulator.

The aggregation and breakage kernel as defined in Equations 2 and 3 respecively. These mechanistic kernels used the DEM inter-collision data as well as the impacts with the geometry to determine the collision and impact rate. The velocity of these particles were used to create a velocity distribution which decided whether the particle agglomerated with other particles or had to undergo breakage due to the impact on the geometry. The further details about the aggregation and breakage kernels can be found in Barrasso and Ramachandran (2015). Details about the lumped model for the gas and liquid present in the system can be found in **(Cire current paper)**.

$$\Re_{agg}(s_1,s_2,x) = \frac{1}{2}\int_0^{s_1}\int_0^{s_2}\beta(s_1',s_2',s_1-s_1',s_2-s_2',x)F(s_1',s_2',x)F(s_1-s_1',s_2-s_2',x)ds_1'ds_2' - F(s_1,s_2,x)\int_0^{s_{max_1}-s_1}\int_0^{s_{max_2}-s_2}\beta(s_1,s_2,s_1',s_2',x)F(s_1',s_2',x)ds_1'ds_2' \quad (2)$$

where, the aggregation kernel, $\beta(s_1,s_2,s_1',s_2',x)$ is expressed as a function of collision rate coefficient (C) and probability that collision results in agglomeration(ψ)

$$\Re_{br}(s_1,s_2,x) = \int_0^{s_{max_1}}\int_0^{s_{max_2}} K_{br}(s_1',s_2',x)F(s_1',s_2',x)ds_1'ds_2' - K_{br}(s_1,s_2,x)F(s_1,s_2,x) \quad (3)$$

where, the breakage kernel $K_{break}(s_1,s_2,x)$.

The number of solid bins slected for this simulation were 16 for each type of solid. The granulator was divided in 16 axial compartments to help with the parallelization of the *C++* code. Since each compartment needed to perform identical calculations which were independent each other, they could be sent different CPU cores without any decrease in the speed as there will be minimum communications between these processes for a given time step. Thus, each of compartment was sent to a single process (core) using message parsing interface (MPI). The PBM code was made to dump the d50 and number of particles at an interval of $0.2s$ such that the conrroller could monitor the execution.

2.2. Discrete Element Model (DEM) setup

The DEM simulation was performed on an open-source DEM software known as LIGGGHTS (Kloss et al., 2012). The DEM simulation has to be initiated using a input file which was generated using a python script which was developed. The python script also helped generate the multiple restart files which would be required to restart the DEM simulation during the parallel multi-scale simulation. LIGGGHTS natively supports parallelization using MPI, which helps divide the geometry into various smaller simulation boxes thus, making them run faster compared to a serial code.
The particles used in the DEM simulation were of diameter $2mm$ and were considered to be granular in nature. A non-cohesive elastic contact model was used to model the collisions between the particles. The particles were introduced at the entry of the geometry with a constant mass flow rate of 15 kg/hr. The impeller of the granulator was set to rotate at 2000 rotations per minute. This was kept so high since this would lead higher number of particles colliding with the walls of granulator leading to breakage of the particles. Physical properties of the partciles required for the simulation were taken from **(cite current paper here)**. The collision and velocity data of the particles from the DEM simulation was printed at pre-decided intervals such that it could be monitored.

2.3. Controller development

The controller scripts to monitor the DEM and PBM simulations were written in python. Each of these scripts was executed along with its respective simulation. The DEM controller script monitored the collision and velocity data being dumped by the DEM simulation. The data was first stored and the average of the velocities for each type of particles and the total number of collisions and impacts weere determined. If there was a change in either of these properties of more than 15%, an exit status was sent which led to killing the execution of the DEM and indicated to start the PBM simulation. The DEM controller script also printed out various data files required for the the PBM execution as well as for the restart of the DEM simulation if needed. In the case, the properties did not showing a variation of more than 15% over the span of $5s$ of DEM simulation time, it was consisdered to be in steady state and that all the simulation were halted.
The PBM controller read the d50 and number of particle files which were printed by the PBM after a constant time interval. The change in each of these properties in each compartment was monitored and an exit signal was sent if they varied by more than 15%. The halting of the PBM was accompanied by the printing of files required to restart DEM with new diameters and data for restarting the PBM if needed after the DEM simulation.

2.4. RADICAL-Pilot (RP)

A major challenge faced while running large number of simulations was the communication between various componenets of the system. Since each of these simulations were run on a high-performance computing (HPC) resource, each component has to be submitted as an individual job and sually leading to a mismatch in the sequence of execution of the jobs. This leads to each job halting before completion until other componenets finish execution. This communication and job

scheduling can be overcome using pilot job abstraction. RP is a pilot job abstraction container developed at Rutgers' electrical and computer engineering department, which supports scalable and efficient launching of heterogeneous tasks across different platforms. RP helped create a link between each of these components. The pilot also handled halting of the executions of the components when indicated by the controller as well as the restart of the DEM and PBM whenever required. This also helped bypass the waiting in multiple job-queues that would have been required to run multiple instances of these simulations.

3. Results

This coupled DEM-PBM simulation were tested on the Stampede2 supercomputer with the Knights Landing configuration. Each of the compute node of the cluster consists of Intel Xeon Phi 7250 ("Knights Landing") which has 68 cores on a single socket clocked at 1.4 GHz, with 96 GB of DDR4 RAM. Each of the cores consists of 4 hardware threads. The simulations were run on various configuration of the cores used for each of the component of the system. The number of cores used for the DEM simulation varied from 16 to 256, while the PBM used cores from 1 to 16. The DEM and PBM controller scripts used 1 process each for their respective executions.
The coupled system took **X** DEM and **Y** PBM simulations to reach steady state. The total simulation time of DEM execution was **A** seconds and PBM was **B** seconds.

3.1. Scaling results

Cores used for DEM and PBM and how scaling occured

3.2. Improved results over one way coupling

Better d50 plots particle count

4. Conclusion

References

D. Barrasso, R. Ramachandran, 2015. Multi-scale modeling of granulation processes: bi-directional coupling of PBM with DEM via collision frequencies. Chemical Engineering Research and Design 93, 304–317.

G. D. Ingram, I. T. Cameron, 2005. Formulation and comparison of alternative multiscale models for drum granulation. Computer Aided Chemical Engineering 20, 481–486.

C. Kloss, C. Goniva, A. Hager, S. Amberger, S. Pirker, 2012. Models, algorithms and validation for opensource DEM and CFD–DEM. Progress in Computational Fluid Dynamics, an International Journal 12 (2-3), 140–152.

M. Sen, D. Barrasso, R. Singh, R. Ramachandran, 2014. A multi-scale hybrid cfd-dem-pbm description of a fluid-bed granulation process. Processes 2 (1), 89–111.