NSF CDSE DRAFT 02

Thor, Gibbs, Zeus, the guy who invented sticky notes, Santa Claus*

North Pole department of ice and snow Fwding address - Department of Chemical and Biochemical Engineering, Rutgers, The State University of New Jersey, Piscataway, NJ, USA 08854

Abstract

abstract text goes here ...

Keywords: Population balance model, heteroaggregation, alginate, chitosan, oppositely charged

1. Introduction & Objectives

Half of all industrial chemical production is carried out using particulate processes (Seville et al. (1997)). Important products that are produced using particulate processes include detergents, aerosols, fertilizers, and pharmaceuticals. Particulate processes have complex phenomena due to the chaotic microscale behaviour of individual particles in the system that determine the bulk behaviour of the system. Due to the complex nature of these processes there is a lack of governing equations which can accurately describe them (Sen et al. (2013)). As a result industries such as pharmaceuticals use expensive heuristic studies and inefficient operational protocols with high recycle ratios to meet strict regulatory standards (Ramachandran et al. (2009)). In an effort to cut costs and improve efficiencies there is a high demand for accurate models that can be used for process design and control systems.

Discrete element methods (DEM) have proven to be an accurate way to model a bulk material using microscale particle level interactions. The downside to these models is that they take large amounts of time to solve. Another more computationally efficient, but less accurate, technique used for these systems are population balance models (PBM). To achieve high levels of accuracy with out the long simulation times of DEM there have been numerous works on hybrid PBM-DEM methods.

in the past when faced with comp heavy tasks scientists have used parallel computing. divide big prob into small pieces that are simultaneously solved. time to solve is time to solve small piece.

potential benefits this could have 1. unprecedented accuracy 2.wide adoption of using these models since they are so fast 3. cheaper drugs 4. MPC accurate but in real time 5. parameter estimation for QbD 6. etc.

1.1. Objectives

The main objective of this study was to develop new techniques to parallelize our model and use high perfomance computing (HPC) equipment in order to reduce the computation times for the linked DEM-PBM simulations. Specific objectives of each individual study are listed below:

- Developed a 4-Dimensional PBM that is parallelized using hybrid techniques (MPI + OpenMP) for optimal utilization of modern high performance computing equipment.

[*] Corresponding author
Email address: `santaclaus` (Santa Claus)

- LIGGGHTS$^{®}$ has been used to perfrom Discrete Element Model (DEM) simulations to model the micromechanics of the Lodgitech granulator.
- These 2 techniques were linked uni-directionally to provide an accurate model of the granulator.

2. Background & Motivation

Particulate processes are ones in which a system of discrete species exist, such as granules or catalyst pellets, that undergo changes in average composition, size, or other pertinent properties. Such processes are prevalent in the pharmaceutical industry. These processes have great oppurtunities in better equipment design, process efficiency and scale-up (Ketterhagen et al. (2009)).

2.1. Modeling

The quality by design (QbD) concept i.e, theoritically modelling the process for better quality of the product is being implemented by the pharmaceutical industry due to its cost benefits. Also, the paradigm shift of the industry towards continuous manufacturing, emphasizes the need to model these processes more accurately. This further helps develop better control stratergies for the process.The modelling of these processes is more time consuming and computationally expensive when compared to a fluid systems since the particles are considered as individual entities rather than a continuum like in fluid systems. The models discussed represent the particle-particle interaction at meso and micro-scale.

2.1.1. Discrete Element Modeling (DEM)

Discrete Element Method is a simulation technique used to monitor the behaviour of each particle as a separate entity compared to other bulk continuum models. This method tracks the movement of each of the particles with in the space, records the collisions of each particle with the geometry as well as with each other and it is also subject to other force fields like gravity (Barrasso and Ramachandran (2015)) This model is based on the Newton's laws of the motion and is expressed as in Equations 1 and 2 :

$$m_i \frac{dv_i}{dt} = F_{i,net} \tag{1}$$

$$F_{i,net} = F_{i,coll} + F_{i,ext} \tag{2}$$

In the above equations m_i represents the mass of the particle, v_i represents the velocity of the particle, $F_{i,net}$ represents the net force on the particle, forces on the particle due to collisions and other external forces are represented in $F_{i,coll}$ $F_{i,ext}$ respectively.

The distance between each particle calculated at every time step and if the distance between two particles is less than the sum of the radii (for spherical particles) a collision between the two particles is recorded. The tolerance for overlap is low in the normal as well as the tangential direction (Cundall and Strack (1979)). Microscale DEM simulations are computationally demanding and simulations may take upto several days to replicate a few seconds of real time experiments. Many methods have been implemented to increase the speed of these simulations, such as scaling by increasing the size of the particles. These approximations are good in understanding the physics of the system but are not directly applicable to process-level simulations.

Thus, this method for granular powder is usually replaced by Population Balance Method which is a much quicker approximation as it is a bulk model.

2.1.2. PBM

Population balance models (PBM) predict how groups of discrete entities will behave on a bulk scale due to certain effects acting on the population with respect to time (Ramkrishna and Singh (2014)). In the context of process engineering and granulation, population balance models are used to describe how the number densities, of different types of particles, in the granulator change as rate processes such as aggregation and breakage reshape particles (Barrasso et al. (2013)). A general form of population balance model is shown here as equation 3.

$$\begin{aligned}\frac{\partial}{\partial t}F(\mathbf{v},\mathbf{x},t)+\frac{\partial}{\partial \mathbf{v}}[F(\mathbf{v},\mathbf{x},t)\frac{d\mathbf{v}}{dt}(\mathbf{v},\mathbf{x},t)]+\frac{\partial}{\partial \mathbf{x}}[F(\mathbf{v},\mathbf{x},t)\frac{d\mathbf{x}}{dt}(\mathbf{v},\mathbf{x},t)]\\ =\Re_{formation}(\mathbf{v},\mathbf{x},t)+\Re_{depletion}(\mathbf{v},\mathbf{x},t)+\dot{F}_{in}(\mathbf{v},\mathbf{x},t)-\dot{F}_{out}(\mathbf{v},\mathbf{x},t)\end{aligned} \tag{3}$$

In equation (3), **v** is a vector of internal coordinates. For modelling a granulation process **v** is commonly used to describe the solid, liquid, and gas content of each type of particle. The vector **x** represents external coordinates, usually spatial variance. For a granulation process this account for spatial variance in the particles as they flow along the granulator.

PBM extensively used for calculating granulation rate processes (e.g. aggregation, consolidation and breakage) but it cannot provide information related to mechanistic kernels of these rate processes i.e. collision frequency, collision efficiency, particle velocity etc. (Sen et al. (2014)). On the other hand, DEM able to determine these entities. A hybrid model for one-way coupling has been reported for continuous mixing (Sen et al. (2013) & Sen and Ramachandran (2013)) and is discussed in later section.

talk about kernels - empirical, semi mech, and DEM informed? talking about kernels then mentioning there is DEM informed could be GREAT SEGWAY to multi-physics models dotdotdot combined PBM DEM etc.

2.1.3. Multi-physics models

The use of multi-physics models has recently been adapted to understand the behaviour of particle systems. These models help understand the physics of the system at various scales $i\dot{e}^{\cdot}$ micro, meso and macro scale(Sen et al. (2014)). Particle process dynamics have been inferred from coupling of various physics models *viz.* Computational Fluid Dynamics (CFD), DEM and PBM. Earlier works from Sen et al. (2014) and Barrasso and Ramachandran (2015) have successfully predicted process dynamics of the granulation process.

In this work, a coupling of DEM and PBM was implemented. The PBM gives meso scale information and the DEM gives particle scale information. Combination of these two methods help comprehend the process dynamics with more accuracy. The calculations involved due to the number of particles involved in the DEM process as well as PBM become very computationally heavy. The recent development in the design of CPUs and increasing number of cores in the CPU, it makes sense to utilize them to make the simulations faster. Thus, implementation of various parallel computing techniques was employed in this work to help improve the simulation times.

2.2. Parallel Computing and Computer Architectures

The goal of parallel computing is to distribute large amounts of computation across many compute cores to solve problems faster (Wilkinson and Allen (2005)).

2.2.1. Computer Architecture

Parallel programs are commonly run on computer clusters. Computer clusters are a collection of nodes interconnected by a high speed communication network for message passing from one node to another. Analogously to a conventional PC each node as one or more CPUs and RAM. Commonly nodes are manufactured with two CPUs, each CPU is a multi-core meaning it has multiple compute cores that each can carry out calculations separately from one another. CPUs also have built in memory called cache that is much faster than RAM which is why for optimal performance cache utilization should be favoured over RAM when possible. On a node memory is divided by CPU sockets, so each CPU has direct access to memory local to its own socket, but accessing memory on another socket is much slower Jin et al. (2011). For this reason data that is needed for computation should be stored locally to the CPU that needs it.

Two common classes of computer architecture classified by memory locality features are distributed memory systems and shared memory systems. These two classes co-exist in a cluster, thus providing the benfits of each. All the nodes share memory using explicit message passing while each has its own independent memory. The cores on each node can access data from the shared memory without any explicit message passing statements from the user. While designing a parallel program all these aspects need to be considered for optimal perfomance of the code (Adhianto and Chapman (2007)).

2.2.2. Parallel Application Program Interfaces

Message Passing Interface (MPI) is a common parallel computing application interface standard. MPI is used for distributed memory parallel computing, this is because MPI will operate every MPI process as a discrete unit that does not share memory with the other processes unless explicit message passing is used. Even on shared a single node where the hardware is supports shared memory computing, MPI will still operate it in a distributed memory fashion Jin et al. (2011). Operating all cores as distinct units also means they each need their own copy of all variables used for computation which results in a large overall memory foot print compared to a same system if it was operated in shared memory.

Open Multi-Processing (OMP) is another application program interface stand for parallel computing. OMP is used for shared memory and can take advantage of shared memory systems which can result in much faster computation. It does not work well on distributed systems though. This prevents it from being used to efficiently carry out computations across multiple nodes of a cluster simultaneously Jin et al. (2011).

Since MPI is best for distributed computing and OMP is better for shared computing many individuals have studied the performance of MPI vs MPI+OMP methods and many studies have used MPI+OMP for scientific computation for improved performance. Often times a trade of is made between optimizing a program for performance and trying to make it flexible enough to run on many different computer architectures might need reference for this. A summary of some works addressing MPI+OMP methods for scientific computing and architecture features and concerns can be found in Bettencourt et al. (2017). In the conclusion to the work by Bettencourt et al. (2017) it was found that hybrid methods for PBMs allow the code flexibility for different architectures while still maintaining good performance. should I mention load balancing techniques of gunawan paper?. In the work of Bettencourt et al. (2017) only the external(spatial) coordinates of the PBM were parallelized. In this current work external and internal(compositions) calculations are parallelized. comment about how int and ext pll methods means better model for xyz reasons dotdotdot helps to explicitly say reason

3. Methods

3.1. DEM

3.1.1. LIGGGHTS

LAMMPS Improved for general granular and granular heat transfer simulations (LIGGGHTS® v3.60) (Kloss et al. (2012)) developed by DCS computing was used for all the simulation performed in this study. Edits were made to the compute_contact_atom source file to obtain particle − particle collisions. The aforementioned version of LIGGGHTS was compiled using the mvapich (mvapich2 v2.1) and intel (intel v15.0.2) compilers with the -o3 optimisation option as well as an option to use OpenMP threads was implemented. The speed improvements and the disadvantages are illustrated in Table (refer to the speed table). The studies were performed on STAMPEDE supercomputer located at Univeristy of Texas, Austin. The hardware configuration of each node consists of 2 8-core Intel Xeon E5-2680 processors based on the Sandy Bridge architecture, 32 gb of memory with QPI interconnects at 8.0 GT/s PCI-e lanes.

3.1.2. Geometry

In this study, the Lödige CoriMix CM5 continuous high shear granulator has been studied. Its geometry was developed using the SolidWorks^{TM} (Dassault Systèmes). This granulator consisted of a high speed rotating element enclosed within a horizontal cylindrical casing. The casing (shown in Figure 1) consists of a cylinder with diameter of 120 mm at the inlet and 130 mm at the outlet and having a total length of 440 mm. A vertical inlet port is provided at one end of the casing and an angled outlet port is provided at the larger end of the case.

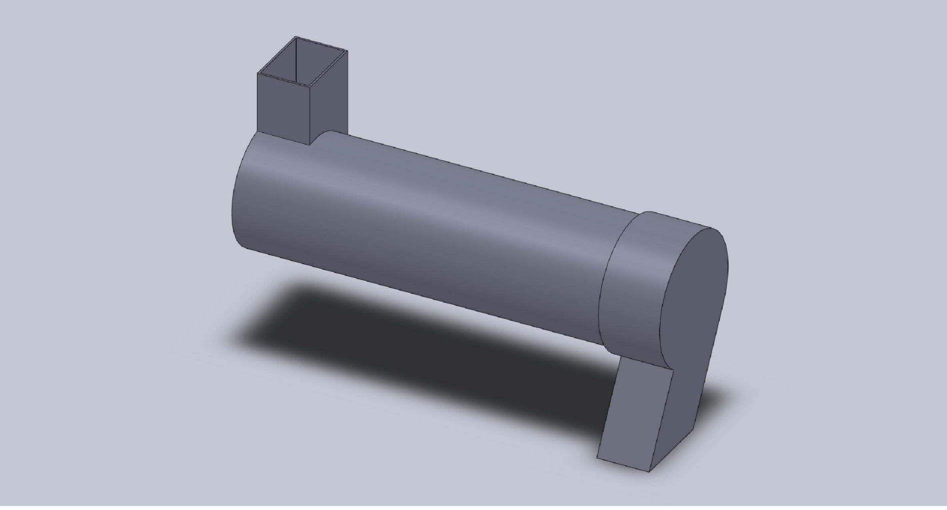


Figure 1: Shows an isometric view of the granulator casing.

The impeller consists of a cylindrical shaft of length 370 mm and diameter 68 mm with four flattened sides 15 mm wide running along the axis. The blades are placed on these flattened sides as shown in figure 2. There are three different blade elements on the shaft (figure 2). At the granulator inlet, there are 4 paddle shaped feed elements following which there are 20 tear drop shaped shearing elements and finally 4 trapezoidal blades near the exit. All these elements are placed in a spiral configuration. The final configuration of the granulator is shown in figure 3.

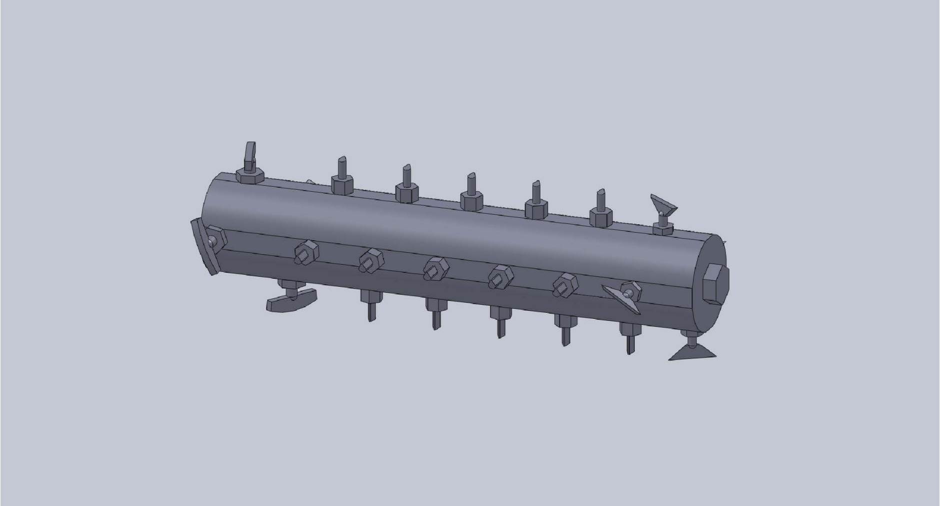
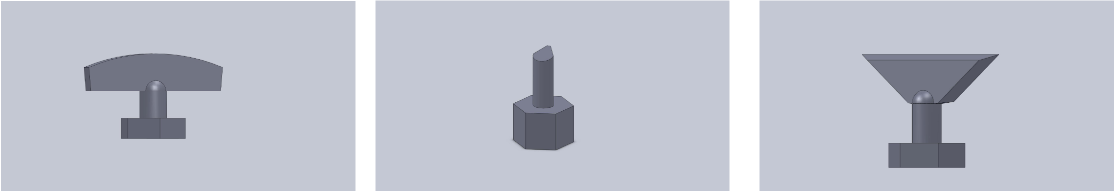


Figure 2: a) Shows the side view of the impeller. b) and c) show the front and back views of the impeller which correspond to the inlet and outlet ends respectively.



(a) Feed element

(b) Shear element

(c) Exit element

Figure 3: Various components of the impeller

3.1.3. Meshing

After the geometry was built in SolidWorks^{TM} (Dassault Systèmes) the shell and impeller were exported as STL files. The coarsest output option was used to keep the STL files small and simple for faster computations times. They were also exported not keeping there original coordinates This resulted in the impeller having 2802 faces and 1281 points with approximately a file size of 775 KBs. The shell had 1948 faces and 720 points and size was about 544 KBs.

Meshlab was used to align the STL files for importing into LIGGGHTS®. No mesh treatments were used on the STLs.

The meshes were then imported into LIGGGHTS® using the write command. This resulted in 50 elements of the impeller file having ”highly skewed elements”, which have more than 5 neighbours per surface or have an angle less than 0.0181185 degrees, that according to LIGGGHTS® would degrade parallel performance. The write exclusion list command in LIGGGHTS® was used and this exclusion list file as then used in the simulation to skip the highly skewed elements during the simulation

3.1.4. DEM input file settings

The DEM simulation in LIGGGHTS® are setup using an input script which defines the physical parameters of the particles, importing of the geometry, particle insertion commands, various physics models to be used during the simulation as well as various compute and dump commands to help print the data required for post-processing of the data. A script was prepared each of the particle was considered granular in nature. The Hertzian model was used for non-adhesive elastic contact between the granular particles. The particles were inserted inside the granluator from the inlet at a constant mass flow rate of 15 kilgrams per hour. The rotation speed of the impeller was kept throughout the study at 2000 rotations per minute. Such a high rotation speed was chosens since this would lead to high shear between the particles and the walls of the shell resulting in better size control of the granules. There were 2 sets of simulations that were performed, one with mono-sized partcles and second consisting of a distribution of sizes. The particle radii chosen for mono-sized simulation varied 0.59mm - 2mm, consecutive particles radii had volume twice of one before them. The radii range of the distributed size simulation was 1mm - 3mm. The difference in the mechanics of these two simulations is dicussed later in the results section. The physical constants used for the simulations are given in Table 1.

The simulation data was collected after a constant number of iterations for the visulation of the particles inside the shell, further post processing and to be used in the PBM. The collisions between each of the particles and the collisions between of the particle and the geometry was also collected.

Table 1: Physical Properties of the particle for the LIGGGHTS® input script

Parameter	**Value**	**Units**
Young's Modulus of particles	8×10^6	$N.m^{-2}$
Young's Modulus of Geometry	1×10^9	$N.m^{-2}$
Poisson's Ratio	0.2	$-$
Coefficient of restitution (constant for all collisions)	0.4	$-$
Coefficient of static friction	0.5	$-$
Coefficient of rolling friction	0.2	$-$
Density of the granules	500	$kg.m^{-3}$

3.1.5. DEM data post processing

The post processing of the data obtained from the DEM simulations was done using MATLAB®. The first test run on the output data was to determine if the simulation had reached steady-state. The mass inside the granulator was found out by averaging it over 5 time steps and then compared to mass inside the granulator after every 10000 time steps (about 5x10^{-4} seconds) with a tolerance of about 10%. If the mass was found to be constant for most of the iterations, it was considered to be at steady state. Another test to determine steady state was to monitor the number of collision inside the granulator. It can be seen that the number of collision start to oscillate around a mean value. The number of collisions were then plotted and steady state time was determined.

A precautionary script was also run so as to determine that no particles were lost due to overlap of the geometry with the particles as well as from particle particle overlap.

3.2. PBM

3.2.1. Model Development

The main PBM equation developed for this work can be expressed as shown below:

$$\frac{d}{dt}F(s_1, s_2, x) = \Re_{agg}(s_1, s_2, x) + R_{break}(s_1, s_2, x) + \dot{F}_{in}(s_1, s_2, x) - \dot{F}_{out}(s_1, s_2, x) \tag{4}$$

citation?

where, $F(s_1, s_2, x)$ is the number of particles with an 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, $F_{in}(s_1, s_2, x)$ and going out, $F_{out}(s_1, s_2, x)$ of the spatial compartment due to particle transfer affect the number of particles in different size classes. The rate of change of liquid volume is calculated using the equation:

$$\begin{aligned}\frac{d}{dt}F(s_1, s_2, x)l(s_1, s_2, x) =& \Re_{liq,agg}(s_1, s_2, x) + \Re_{liq,break}(s_1, s_2, x) + \dot{F}_{in}(s_1, s_2, x)l_{in}(s_1, s_2, x) \\ &- \dot{F}_{out}(s_1, s_2, x)l_{out}(s_1, s_2, x) + F(s_1, s_2, x)\dot{l}_{add}(s_1, s_2, x) \qquad (5)\end{aligned}$$

where, $l(s_1, s_2, x)$ is the amount of liquid volume in each particle with API volume of s_1 and excipient volume of s_2 in the spatial compartment x. $\Re_{liq,agg}(s_1, s_2, x)$ and $\Re_{liq,break}(s_1, s_2, x)$ are respectively the rates of liquid transferred between size classed due to aggregation and breakage. $l_{in}(s_1, s_2, x)$ and $l_{out}(s_1, s_2, x)$ are respectively the liquid volumes of the particles coming in and going out of the spatial compartment. $l_{add}(s_1, s_2, x)$ is the volume of liquid acquired by each particle in the compartment at every time step due to external liquid addition.

Similarly, the rate of change of gas volume is calculated using the following equation:

$$\begin{aligned}\frac{d}{dt}F(s_1, s_2, x)g(s_1, s_2, x) =& \Re_{gas,agg}(s_1, s_2, x) + \Re_{gas,break}(s_1, s_2, x) + \dot{F}_{in}(s_1, s_2, x)g_{in}(s_1, s_2, x) \\ &- \dot{F}_{out}(s_1, s_2, x)g_{out}(s_1, s_2, x) + F(s_1, s_2, x)\dot{g}_{cons}(s_1, s_2, x) \qquad (6)\end{aligned}$$

citation?

where, $g(s_1, s_2, x)$ is the gas volume of each particle with API volume of s_1 and excipient volume of s_2 in the spatial compartment x. $\Re_{gas,agg}(s_1, s_2, x)$ and $\Re_{gas,break}(s_1, s_2, x)$ are respectively the rates of gas transferred between size classed due to aggregation and breakage. $g_{in}(s_1, s_2, x)$ and $g_{out}(s_1, s_2, x)$ are respectively the gas volume of the particles entering and leaving the spatial compartment. $g_{cons}(s_1, s_2, x)$ is the volume of gas coming out of each particle in the compartment at every time-step due to consolidation of the particles.

The rate of aggregation, $\Re_{agg}(s_1, s_2, x)$ in Equation 4 is calculated as

$$\begin{aligned}\Re_{agg}(s_1, s_2, x) =& \frac{1}{2}\int_0^{s_1}\int_0^{s_2} \beta(s_1', s_2', s_1 - s_1', s_2 - s_2', x)F(s_1', s_2', x)F(s_1 - s_1', s_2 - s_2', x)ds_1'ds_2' \\ &- F(s_1, s_2, x)\int_0^{s_{max_1}-s_1}\int_0^{s_{max_2}-s_2} \beta(s_1, s_2, s_1', s_2', x)F(s_1', s_2', x)ds_1'ds_2' \qquad (7)\end{aligned}$$

citation?

where, the aggregation kernel, $\beta(s_1, s_2, s_1', s_2', x)$ is expressed as a function of collision frequency (C) and collision efficiency (ψ)

$$\beta(s_1, s_2, s_1', s_2', x) = \beta_o C(s_1, s_2, s_1', s_2')\psi(s_1, s_2, s_1', s_2', x) \qquad (8)$$

where, β_o is aggregation rate constant.

Collision frequency is a function of particle size and is calculated from the number of collisions between group of particles obtained from LIGGGHTS®. A recent study shows that collision frequency depends on PSD as well (Sen et al. (2014)). Collision frerquency can be expressed as:

$$C(s_1, s_2, s_1', s_2') = \frac{N_{coll}(s_1, s_2, s_1', s_2')}{F(s_1, s_2)F(s_1', s_2')\Delta t} \qquad (9)$$

In equation(9), N_{coll} is the number of collision between two solid particles in time interval Δt. The ψ in equation (8) can be expressed asvi

$$\psi((s_1, s_2, s_1', s_2') = \begin{cases} \psi_0, & LC((s_1, s_2) \geq LC_{min} \ \ or \ \ LC((s_1', s_2') \geq LC_{min} \\ 0, & LC((s_1, s_2) < LC_{min} \ \ or \ \ LC((s_1', s_2') < LC_{min} \end{cases} \tag{10}$$

In above equation, LC is the liquid content of particles and LC_{min} stands for minimum liquid content required for coalescence of particle.

Similarly, the breakage rate is expressed as-

$$\Re_{break}(s_1, s_2, x) = \int_0^{s_{max_1}} \int_0^{s_{max_2}} K_{break}(s_1', s_2', x) F(s_1', s_2', x) ds_1' ds_2' - K_{break}(s_1, s_2, x) F(s_1, s_2, x) \tag{11}$$

citation?

where, the breakage kernel $K_{break}(s_1, s_2, x)$ is formulated as

$$K_{break}(s_1, s_2, x) = C_{impact} \int_{U_{break}}^{\infty} p(U) dU \tag{12}$$

citation?

The rate of increase of liquid volume of one particle, $\dot{l}_{add}(s_1, s_2, x)$ is expressed as $\frac{(s_1+s_2)(\dot{m}_{spray}(1-c_{binder})-\dot{m}_e vap)}{m_{solid}(x)}$ where, $(s_1 + s_2)$ is the total solid volume of the particle; $\dot{m}_s pray$ is the rate of external liquid addition, c_{binder} is the concentration of binder in the external liquid (which is assumed to be zero in this case as pure liquid is added); $\dot{m}_{evap}$ is the rate of evaporation of liquid from the system (which is also assumed to be zero in this case) and m_{solid} is the total amount of solid present in the compartment.

The rate of decrease in gas volume per particle due to consolidation is calculated using the following expression:

$$\dot{g}_{cons}(s_1, s_2, x) = c * (\nu_{impeller})^{\omega} * V(s_1, s_2, x) \frac{(1 - \epsilon_{min})}{s} [g(s_1, s_2, x) + l(s_1, s_2, x) - (s_1 + s_2) \frac{\epsilon_{min}}{1 - \epsilon_{min}}] \tag{13}$$

where, c and ω are the consolidation constants; $v_{impeller}$ is the impeller rotational speed; $V(s_1, s_2, x)$ is the volume of particle, ϵ_{min} is the minimum porosity; $g(s_1, s_2, x)$ and $l(s_1, s_2, x)$ are respectively the gas and liquid volume of the particle.

Particle transfer rate, $F_{out}(s_1, s_2, x)$ in Equation 4 is calculated as $F(s_1, s_2, x) * \frac{\nu_{compartment(x)} * dt}{d_{compartment}}$ where, $\nu_{compartment(x)}$ and $d_{compartment}$ are respectively the average velocity of particles in compartment x and the distance between the mid-points of two adjacent compartment, which is the distance particles have to travel to move to the next spatial compartment. dt is the time-step. The values of various parameters used in the model are provided in Table 2.

3.2.2. Parameters

3.3. PBM Parallel C++

3.3.1. Discretization & Parallelizing PBM

The PBM was discretized by converting each of its coordinates in to discrete bins. For the spatial coordinates a linear bin spacing was used. For the internal coordinates, solid,liquid, and gas

Table 2: Parameters for PBM from Anik's hetero. agg. paper. currently place holder

Parameter	**Symbol**	**Value**	**Units**
Time step	δt	0.5	s
Total granulation time	T	$20, 120$	s
Velocity in axial direction	v_{axial}	1	ms^{-1}
Velocity in radial direction	v_{radial}	1	ms^{-1}
Aggregation constant	β_0	1×10^{-12}	$-$
Initial particle diameter	R	15	μm
Breakage kernel constant	B	5×10^4	$-$
Diameter of impeller	D	0.1	m
Impeller rotation speed	RPM	300	rmp
Minimum granule porosity	ϵ_{min}	0.1	$-$
Consolidation rate	C	1×10^{-3}	$-$
Total starting particles in batch	$F_{initial}$	1×10^6	$-$
Liquid to solid ratio	L/S	0.7	$-$
Number of Compartments	c	3	$-$
Number of first solid bins	s	16	$-$
Number of second solid bins	ss	16	$-$

a nonlinear binning was used. get more details from Anik on this will probably need more detail on binning for reproducability

Once the PBM had been discretized (compartmentalized/binned) a finite differences method was used which created a system of ordinary differential equations (ODEs). The numerical integration technique used to evaluate the system of ODEs was first order Euler integration check with Anik that is what we used as it is commonly used to solve these types of systems and author found it improved speeds while having minimal impact on accuracy. To obtain the most optimal parallel performance, when solving the PBM, work loads were distributed in a manner which took into account the shared memory and distributed memory aspects of the clusters the PBM was being run on. To parallelize the model in a why which could take advantage of shared memory but still effectively run across a distributed system both OMP and MPI were implemented.

One MPI process was used per CPU socket and one OMP process was used per CPU core, as authors (Bettencourt et al. (2017)) found it resulted in the best performance. MPI was used for message passing from one node to another while OMP was used for calculations on each node that could be efficiently solved using a shared memory system i.e. calculations were inter-dependent but could be computed simultaneously.

Pseudo code is presented below illustrating how the calculations are distributed and carried out during the simulation. For each time step the MPI processes are made responsible for a specific chunk of the spatial compartments. Then each OMP thread, inside of each MPI process, is allocated to one of the cores of the the multi-core CPU the MPI process is bound too. The OMP processes divide up and compute $\Re_{agg}$ and $\Re_{brk}$. (include more detail about how they do it? last paper reviewer complained that could not understand figure by JUST reading what I wrote about it in meat of paper)

After $\Re_{agg}$ and $\Re_{brk}$ are calculated the MPI processes calculate the new PSD value for their chunk at that specific time step, $F_{t,c}$. The slave processes send their $F_{t,c}$ to the master processes which collects them into the complete $F_{t,all}$. The master process then broadcasts the $F_{t,all}$ value to all slave processes.

A crucial feature of the PBM is that the current PSD ($F_{t,all}$) value is used to compute a new time step size for the next iteration. This means all of the MPI processes need to have the same dynamic time step size at each iteration for the calculations to be properly carried out in parallel. Since the completely updated $F_{t,all}$ value is shared before calculating a new time step each process will have the same $F_{t,all}$ value. As a result each process calculates the same size for the new time step. Did not include the liquid and gas PBMs in this but hoping they will be some what assumed? Also the Ragg omp distributed work is an a what about private OMP vars specified that has impact on how model is solved etc. Should look into this. might change based on locking/blocking tests that need to be implemented still.

3.4. RP & PBM+DEM Communication

4. Results

4.1. PBM

4.1.1. PBM Validation

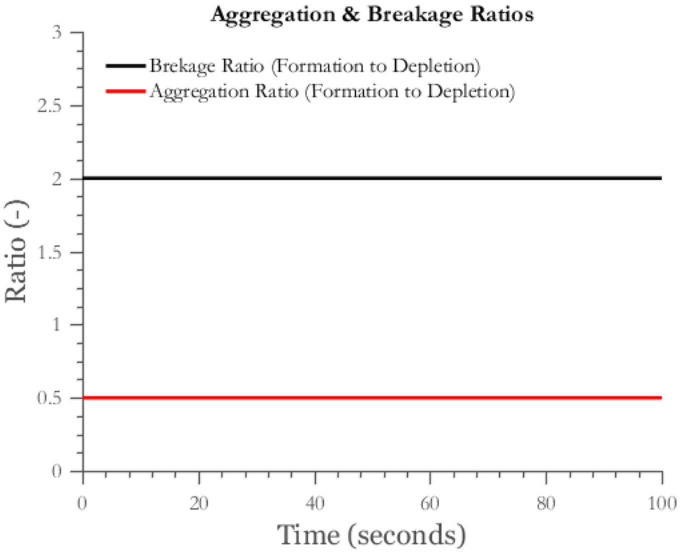


Figure 4: Ratio of formation-to-depletion through aggregation and breakage over time. Breakage ratio of 2 and aggregation ratio of 0.5 indicate mass conservation in the model. NOTE DID NOT HAVE .fig file for this figure so it is in as JPG will need to replace

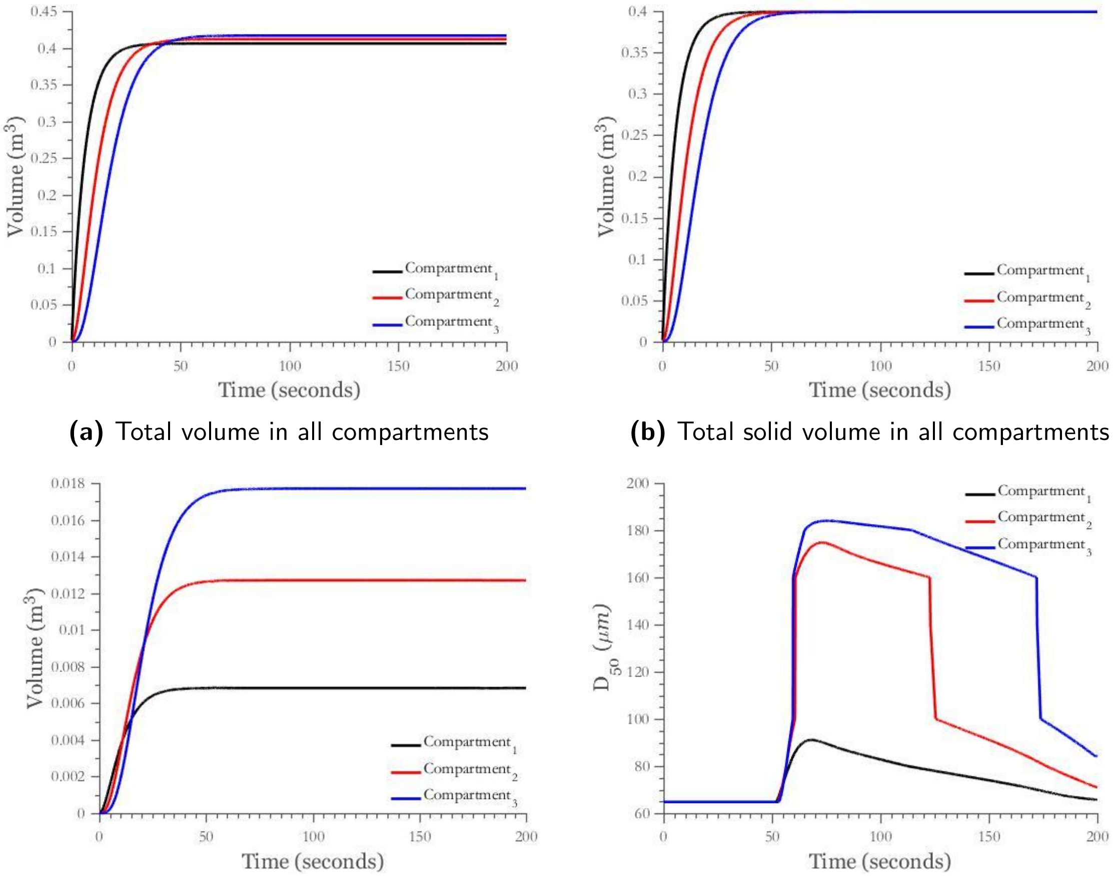
The ratio between the number of particles formed due to aggregation and the number of particles depleted due to aggregation and the ratio of the number of particles formed due to breakage to the number of particles depleted due to breakage are plotted. In aggregation two particles agglomerate to form one particle and in breakage one particle breaks to form two particles. So, these ratios are expected to be 0.5 and 2 respectively. As can be seen from Figure 4, these ratios are accurate confirming that mass is conserved accurately in the model.

The granulator was divided into 3 compartments spatially and the total volume, solid volume and pore volume and the median diameter d_50 in each compartment were plotted to study the granulation behaviour and are shown in Figure 5.

Algorithm 1 Pseudo code

```
while $t < t_{final}$ do
      // the spatial domain is divided into equal chunks (with in 1 bin size)
      // each MPI process is assigned on chunk of spatial domain shown as $c_{low}$ to $c_{up}$
      // sum all $c_{low_i}$ to $c_{up_i}$ is = to [0,numCompartments]
    for each MPI processes do $c = c_{low_i}$ to $c_{up_i}$
      // each MPI process is further divided with OMP to take advantage of multi-core CPU
      // each OMP process is allocated to a single compute core
      // $\Re$ integrals $(i1)$ $\int_0^{s_2}$, $(i2)$ $\int_0^{s_{max_2}-s_2}$, and $(i3)$ $\int_0^{s_{max_2}-s_2}$ are divided into smaller integrals
      // $\int_{i_1low_n}^{i_1up_n}$, $\int_{i_2low_n}^{i_2up_n}$, and $\int_{i_3low_n}^{i_3up_n}$ which are solved by the "n" OMP processes
      // allocated to that MPI process (CPU)
        for each OMP process do
            $$\begin{aligned}\Re_{agg}(s_1,s_2,c) =& \frac{1}{2}\int_0^{s_1}\int_{i_1low_n}^{i_1up_n}\beta(s_1',s_2',s_1-s_1',s_2-s_2',c)F(s_1',s_2',c)F(s_1-s_1',s_2-s_2',c)ds_1'ds_2' \\ &- F(s_1,s_2,c)\int_0^{s_{max_1}-s_1}\int_{i_2low_n}^{i_2up_n}\beta(s_1,s_2,s_1',s_2',c)F(s_1',s_2',c)ds_1'ds_2'\end{aligned}$$
            $$\Re_{break}(s_1,s_2,c) = \int_0^{s_{max_1}}\int_{i_3low_n}^{i_3up_n}K_{break}(s_1',s_2',c)F(s_1',s_2',c)ds_1'ds_2' - K_{break}(s_1,s_2,c)F(s_1,s_2,c)$$
        end for
        $$\begin{aligned}F_{t,c} &= \frac{\Delta F(s_1,s_2,c)}{\Delta t}\Delta t + F(s_1,s_2,c)_{t-1} \\ &= (\Re_{agg}(s_1,s_2,c) + \Re_{break}(s_1,s_2,c) + \dot{F}_{in}(s_1,s_2,c) - \dot{F}_{out}(s_1,s_2,c))\Delta t + F(s_1,s_2,c)_{t-1}\end{aligned}$$
    end for
    MPI Send $F_{t,c}$ to Master MPI process
    MPI Recv $F_{t,c}$ from MPI all slave processes
    Master consolidate all $F_{t,c}$ chunks into a complete $F_{t,all}$
    Master does inter-bin particle transfers (updates $F_{t,all}$)
    MPI Bcast $F_{t,all}$ to all slave processes
    $t_{new} = t + timestep$
end while
```

It can be seen from Figure 5a that the total volume starts to increase first in compartment 1 followed by compartment 2 and then compartment 3. This happens as gradually particles entering compartment 1 moves to the other compartment due to particle transfer from compartment 1 to compartment 2 and then compartment 3. In Figure 5b it is observed that the solid volume similar to the total volume increases first in compartment 1 and last in compartment 3. The solid volume becomes constant and equal in all the compartments at around 30-50 seconds and steady state is reached when the rate of particle volume being transported through the compartments and leaving the system is equal to the rate of particles entering the system. Although, as seen in Figure 5c the pore volume which is the sum of the gas and the liquid volume is highest in compartment 3 and lowest in compartment 1. This happens due to the external liquid addition to the system. As the particles move from compartment 1 to compartment 3, they gradually acquire a higher amount liquid, thereby increasing the pore volume. In Figure 5d, the D_50 is seen to be increasing from compartment 1 to 3. This happens because of the size enlargement of large particles coming in from the previous compartment because of the external liquid added to each compartment and a longer residence time in the granulator.



(a) Total volume in all compartments

(b) Total solid volume in all compartments

(c) Total pore volume in all compartments

(d) D_{50} in all compartments

Figure 5: Volume and D50 in all compartments over time. Volumes become constant as steady state is reached. Median diameter increases and then decreases as bigger particles leave the system and smaller particles occupy that volume.

4.1.2. Parallel C++ PBM Validation

show PSD or D_50 is the same as Matlab or serial PBM

1. fig D_50 Matlab vs Parallel

4.1.3. Parallel PBM Performance

show that RP has minimal impact on performance

show that performance is mostly unaffected by RP

1. fig scaling

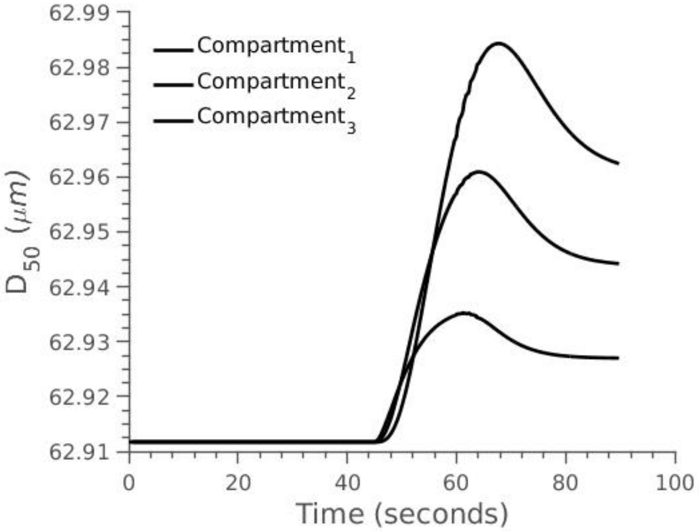


Figure 6: D_{50} of Matlab PBM vs Parallel PBM

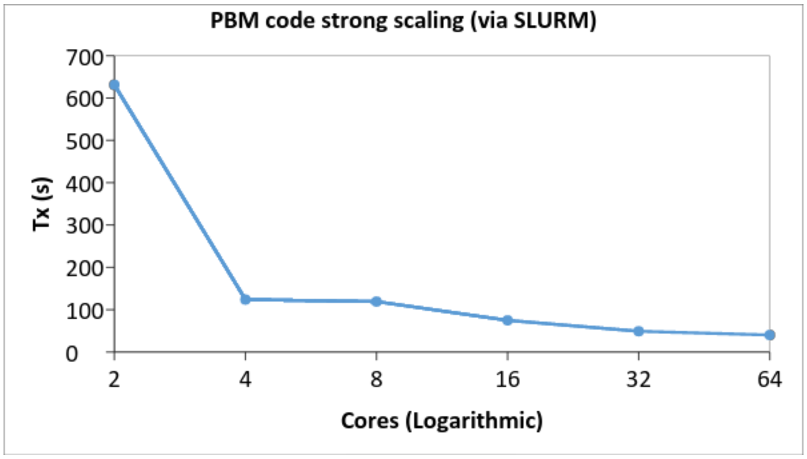


Figure 7: PBM strong scale slurm

2. fig scaling w/ RP

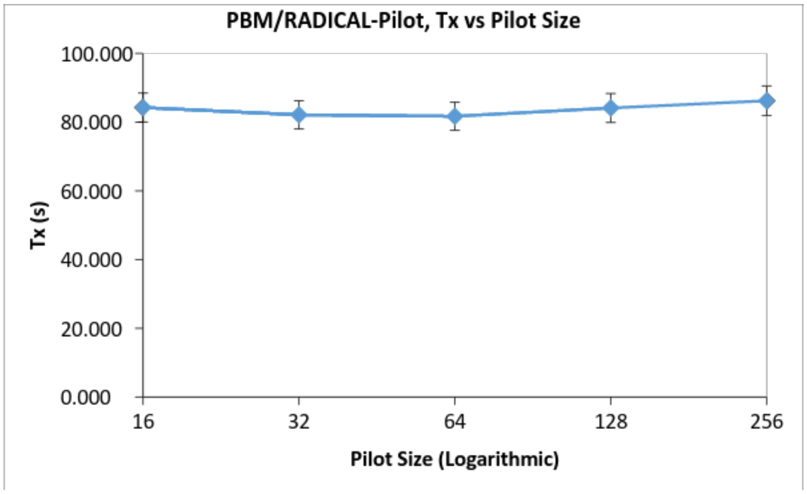


Figure 8: PBM strong scale RP

4.1.4. Parallel PBM Parameter space and Parameter Estimation

I. show how effective parallel pbm is for parameter estimation

II. Find/ explore ranges of DEM data that PBM can use to find Critical parameters and sensitivities will be useful to us in linking and in picking best DEM parameters to vary and best parameters for PBM+DEM code

1. fig range of some parameters?

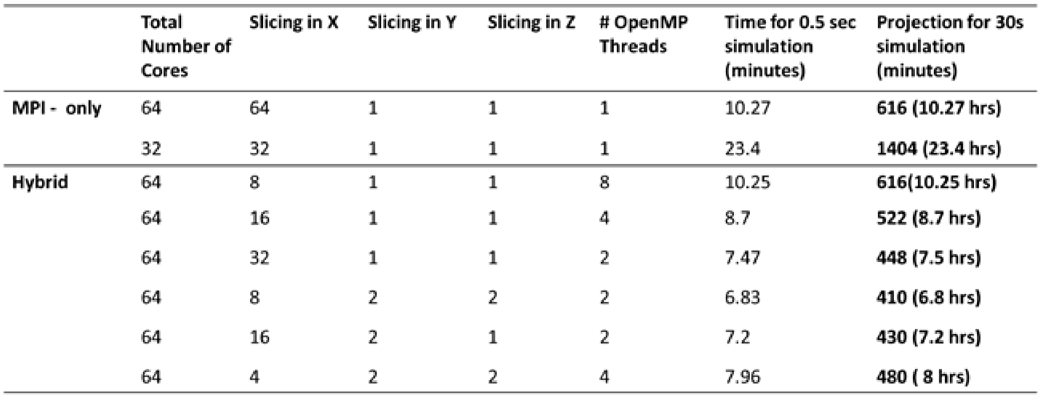
2. fig range of other pbm parameter ?

4.2. DEM

4.2.1. DEM Spacial Decomposition Studies

The effect of spacial decomposition on the simulation time

Speed improvements and issues using only MPI and hybrid (MPI + OpenMP)



	Total Number of Cores	Slicing in X	Slicing in Y	Slicing in Z	# OpenMP Threads	Time for 0.5 sec simulation (minutes)	Projection for 30s simulation (minutes)
MPI - only	64	64	1	1	1	10.27	**616 (10.27 hrs)**
	32	32	1	1	1	23.4	**1404 (23.4 hrs)**
Hybrid	64	8	1	1	8	10.25	**616(10.25 hrs)**
	64	16	1	1	4	8.7	**522 (8.7 hrs)**
	64	32	1	1	2	7.47	**448 (7.5 hrs)**
	64	8	2	2	2	6.83	**410 (6.8 hrs)**
	64	16	2	1	2	7.2	**430 (7.2 hrs)**
	64	4	2	2	4	7.96	**480 (8 hrs)**

Figure 9: The effect of spacial decomposition on the performance of the simulation

fig vary PSD (range and/or particle sizes) to see how C_{coll} etc will be affected - important for PBM Kernel

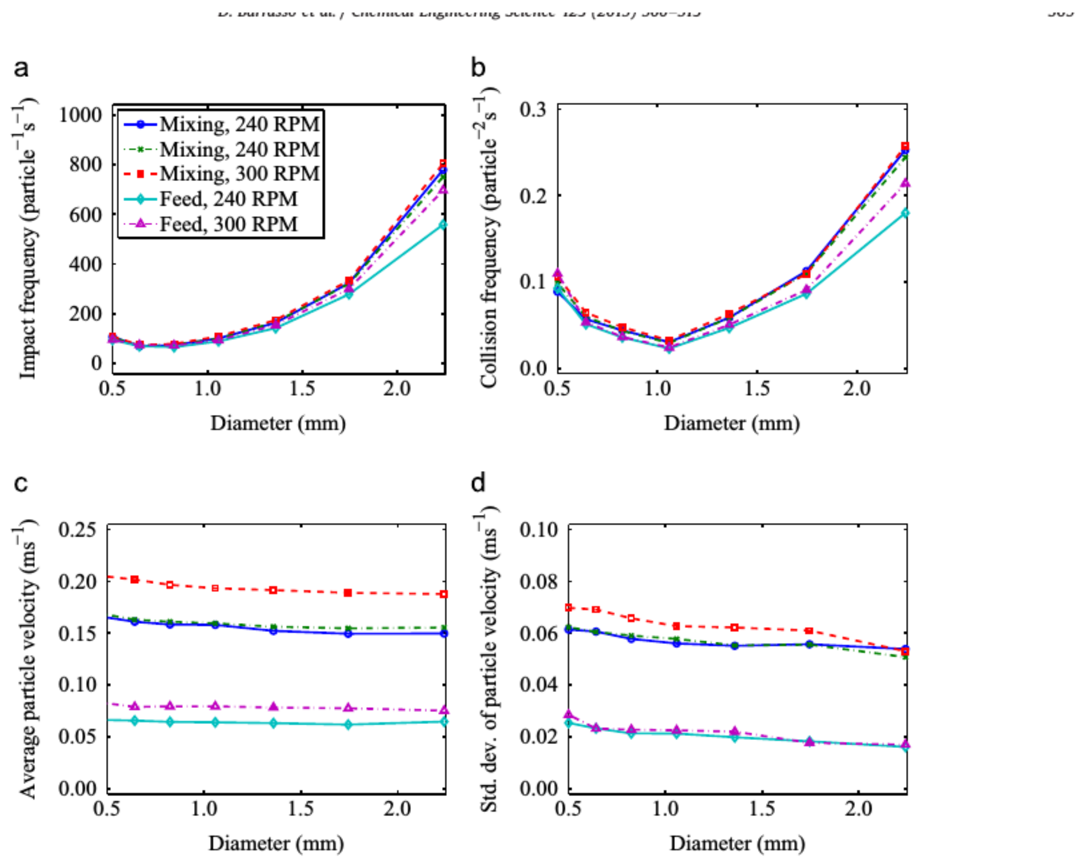


Fig. 6. DEM results showing the effects of screw element type and rotational speed on (a) impact frequency, (b) collision frequency with 1-mm particles, (c) average particle velocity, and (d) standard deviation of particle velocity for each size class.

Figure 10: fig showing sensitivity of C_{coll} and etc to RPM

Talk about over-slicing or excess decomposition(confirm with profs) collection, respectively.

feed screw geometries at two different screw speeds. The mixing

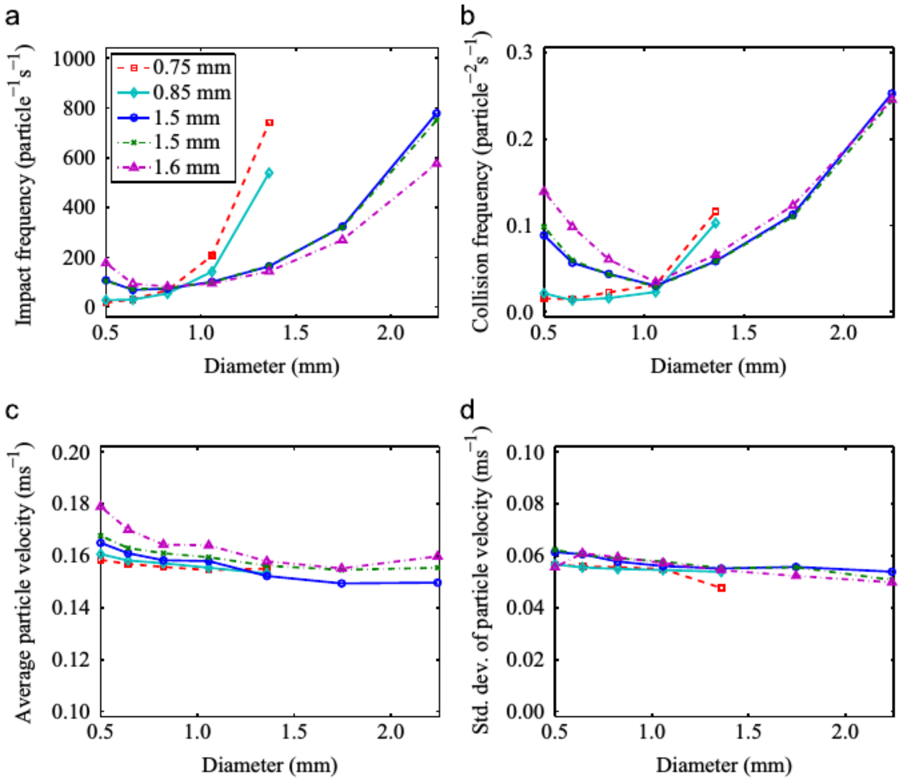


Fig. 4. DEM results showing the effects of size distribution on (a) impact frequency, (b) collision frequency with 1-mm particles, (c) average particle velocity, and (d) standard deviation of particle velocity for each size class. Median diameters are listed in the legend. The standard deviation of the diameter is fixed at 0.2 mm for the smallest two simulations and 0.5 mm for the larger simulations.

Figure 11: fig from dana 2015 mechanistic bi-directional

4.2.2. DEM Performance

Talk about how the time for simulation varies with # of cores

1. fig scaling

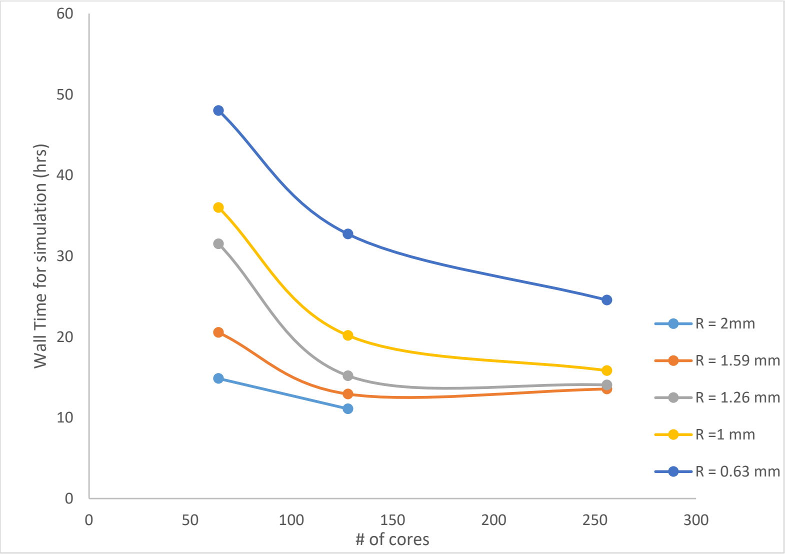


Figure 12: The variation in the amount of time taken for the simulation as a function of # of cores

2. fig scaling w/ RP

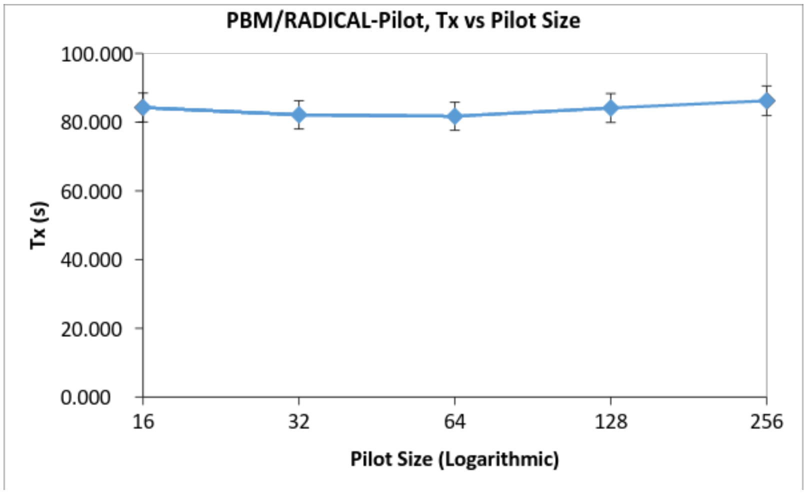


Figure 13: DEM scale with RP

4.3. PBM+DEM - RP

4.3.1. PBM+DEM Validation/Accuracy?

Talk about how the physics remain the same whether mono-sized particles are used or a size distribution is used.

Figs comparing the one-way coupling from both these sources.

4.3.2. PBM+DEM Performance

strong scaling

fig PBM + DEM RP strong scaling

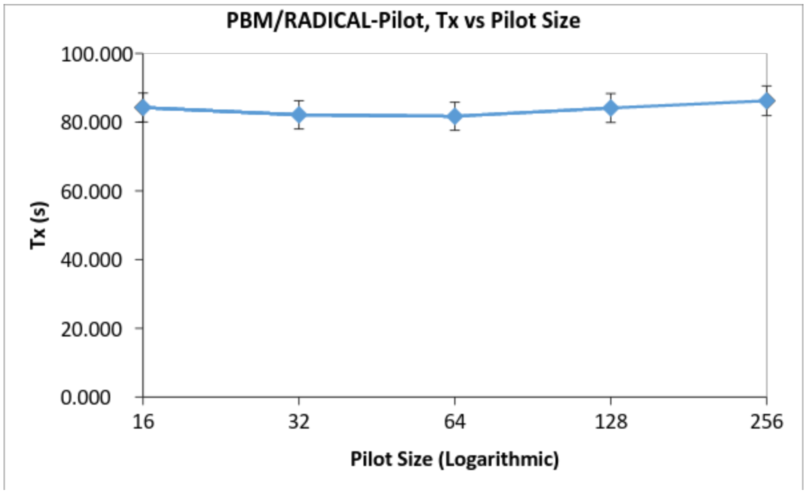


Figure 14: PBM+DEM scale with RP

4.3.3. PBM+DEM Parameter studies

show how PBM+DEM captures multi-physics as parameters changed. helps validate and support model development. show we have made a useful tool for future work.

fig

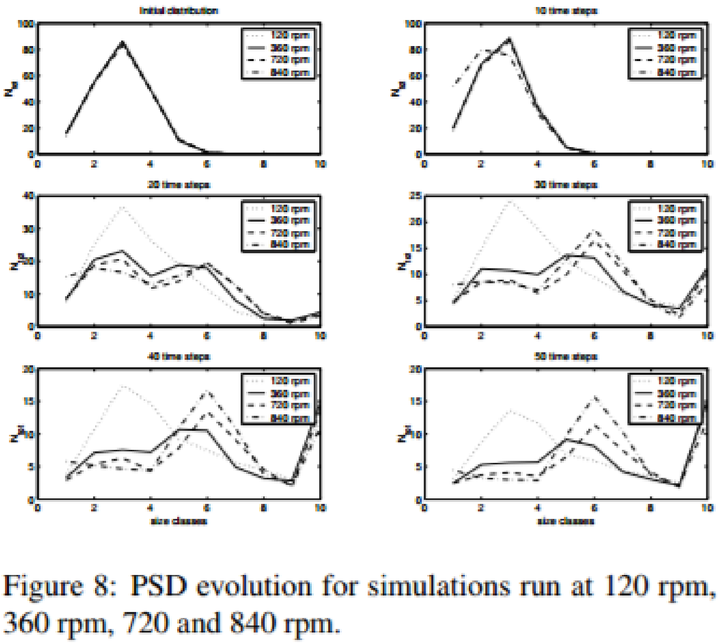


Figure 8: PSD evolution for simulations run at 120 rpm, 360 rpm, 720 and 840 rpm.

Figure 15: PBM+DEM scale with RP

5. Conclusions

References

Adhianto, L., Chapman, B., 2007. Performance modeling of communication and computation in hybrid {MPI} and openmp applications. Simulation Modelling Practice and Theory 15 (4), 481 – 491, performance Modelling and Analysis of Communication Systems.
URL http://www.sciencedirect.com/science/article/pii/S1569190X06001109

Barrasso, D., Ramachandran, R., 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.
URL http://www.sciencedirect.com/science/article/pii/S0263876214001920

Barrasso, D., Walia, S., Ramachandran, R., 2013. Multi-component population balance modeling of continuous granulation processes: A parametric study and comparison with experimental trends. Powder Technology 241, 85 – 97.
URL http://www.sciencedirect.com/science/article/pii/S0032591013001666

Bettencourt, F. E., Chaturbedi, A., Ramachandran, R., 2017. Parallelization methods for efficient simulation of high dimensional population balance models of granulation. Computers & Chemical Engineering (Article in press).
URL http://www.sciencedirect.com/science/article/pii/S0098135417301072

Cundall, P. A., Strack, O. D. L., 1979. A discrete numerical model for granular assemblies. Gotechnique 29 (1), 47–65.
URL http://dx.doi.org/10.1680/geot.1979.29.1.47

Jin, H., Jespersen, D., Mehrotra, P., Biswas, R., Huang, L., Chapman, B., 2011. High performance computing using {MPI} and openmp on multi-core parallel systems. Parallel Computing 37 (9), 562 – 575, emerging Programming Paradigms for Large-Scale Scientific Computing.
URL http://www.sciencedirect.com/science/article/pii/S0167819111000159

Ketterhagen, W. R., am Ende, M. T., Hancock, B. C., 2009. Process modeling in the pharmaceutical industry using the discrete element method. Journal of pharmaceutical sciences 98 (2), 442–470.

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

Ramachandran, R., Immanuel, C. D., Stepanek, F., Litster, J. D., Doyle III, F. J., 2009. A mechanistic model for breakage in population balances of granulation: Theoretical kernel development and experimental validation. Chemical Engineering Research and Design 87 (4), 598 – 614, 13th European Conference on Mixing: New developments towards more efficient and sustainable operations.
URL http://www.sciencedirect.com/science/article/pii/S0263876208003225

Ramkrishna, D., Singh, M. R., 2014. Population Balance Modeling: Current Status and Future Prospects. Annual Review of Chemical and Biomolecular Engineering 5 (1), 123–146.
URL http://dx.doi.org/10.1146/annurev-chembioeng-060713-040241

Sen, M., Barrasso, D., Singh, R., Ramachandran, R., 2014. A multi-scale hybrid cfd-dem-pbm description of a fluid-bed granulation process. Processes 2 (1), 89–111.
URL http://www.mdpi.com/2227-9717/2/1/89

Sen, M., Dubey, A., Singh, R., Ramachandran, R., 2013. Mathematical Development and Comparison of a Hybrid PBM-DEM Description of a Continuous Powder Mixing Process. Journal of Powder Technology 2013, 1–11.
URL http://www.hindawi.com/archive/2013/843784/

Sen, M., Ramachandran, R., 2013. A multi-dimensional population balance model approach to continuous powder mixing processes. Advanced Powder Technology 24 (1), 51–59.

Seville, J., Tzn, U., Clift, R., 1997. Processing of Particulate Solids. Springer Netherlands, Dordrecht, dOI: 10.1007/978-94-009-1459-9.
URL http://link.springer.com/10.1007/978-94-009-1459-9

Wilkinson, B., Allen, C., 2005. Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers. An Alan R. Apt book. Pearson/Prentice Hall.
URL https://books.google.com/books?id=F8C2QgAACAAJ