Solution of Multidimensional Pure Aggregation Population Balance Equations using Pivot Techniques

Thesis to be submitted in partial fulfillment of the requirements for the degree

of

Master of Technology in Chemical Engineering

by

Kunal Katiyar
18CH30015

Under the guidance of

Prof. Jayanta Chakraborty

CHEMICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Department of Chemical Engineering
Indian Institute of Technology,
Kharagpur
India - 721302

CERTIFICATE

This is to certify that the work report entitled **Solution of Multidimensional Pure Aggregation Population Balance Equations using Pivot Techniques** is submitted by **Kunal Katiyar**, Roll Number: *18CH30015* to Department of Chemical Engineering, IIT Kharagpur, is an authentic and genuine record of the M.Tech Project-I carried out by him under my supervision. The report submitted by him has fulfilled all the requirements as per the norms and guidelines given by this institute for M.Tech Project-I.

Prof. Jayanta Chakraborty

Department of Chemical Engineering
Indian Institute of Technology, Kharagpur

Date: 13th November, 2022

ACKNOWLEDGEMENT

I take this opportunity to express my profound gratitude and indebtedness to my professor and faculty guide Prof. Jayanta Chakraborty for his sincere guidance and motivation towards my work. In these difficult times, I would like to express my heartfelt obligation for his coordination in extending every possible support for the completion of this project.

I would also like to sincerely thank and extend my earnest recognition to Prof. Jitendra Kumar for guiding and helping me throughout the whole project.

Kunal Katiyar
IIT Kharagpur
Date: 13th November, 2022

Contents

List of Figures

ABSTRACT

Particulate systems define a multitude of study areas like bubbles, particles,etc. and having a model to gather data about a population of such particles is very important in the case of aggregation. This study is directed to create a model to get the population balance in case of 1 dimensional aggregation of particles and extending it to 2 dimensions later on. For getting such numerical results efficiently and as close as possible to analytical solutions, discretization has been used through fixed pivot techniques. We divide internal attributes of particles to pre-defined closest neighbouring pivots and calculate the population at these pivots at a later time subject to pure aggregation. These populations are compared with the analytical solutions under different degrees of aggregation and a variety of grid sizes used in discretization. The results show that there is a minimal loss in accuracy in the results for finer grids and lower degrees of aggregations, but even in the cases of cruder grids the results are close enough that they can be used as a substitute.

Keywords: Aggregation, Population Balance Models, Fixed Pivot

Introduction

Population Balances are widely used in many studies including sub-micron chemical processes, crystallization, powders, colloids and can be even expanded to human populations. Since the solution of these provide the population of particles at a point, we can decipher a multitude of internal properties from it, which can then be averaged to gain a bigger understanding of the study at hand.

Particle aggregation needs to be studied in depth as it is a widely known phenomenon occurring in multiple domains including water treatment, paper-making, etc. Since aggregation is always in conjunction with growth (growing in size of particle) and breakage (division of particles) it makes the study of aggregation that much tougher. Population Balance Equations helps in resolving this problem, and we would look into solution of those in depth.

As Population Balance Equations are a set of integro-differential equations, it becomes necessary to generate good numerical results for this case, due to increased time taken to solve them. As the aggregation kernel becomes increasingly complex, the equation gets tougher to solve. To solve this problem, we must look into multiple types of numerical solutions that can be used to solve such equations efficiently in the case of aggregation. Majorly, we will be focusing on a type of method called "Fixed Pivot Method"[5] which helps in discretiziation of the continous equations and results in pretty accurate results when compared to analytical solutions.

Objective

The paper attempts to discuss the simulations of the numerical methods of Pure Aggregation. The main objectives of this study: are:

1. Discuss about the aggregation population balance model

2. Discuss about the numerical solutions that can be used to simulate the PBE

3. Simulation of the above PBE for the case of 1D Pure Aggregation

4. Improve upon the computation speeds of such simulations while providing an easy-to-access platform to re-run simulations on-the-go

Apart from the simulation, the paper also compares the results with existing literature and significantly decrease the computation speeds of the simulation

Aggregation

When two or more particles hit and come together, this is known as aggregation. In some situations, such as when liquid drops collide, the colliding particles entirely lose their individuality, making it impossible to distinguish the original particles from the aggregate. When solid particles come together because of their attractive potential, the grain boundary that corresponds to the surface of the original particles is often present in the aggregate, but it may subsequently disappear. We shall refer to both forms of aggregation as aggregation and treat the aggregated particle as a single entity.

Binary aggregation is the accumulation of two particles. Higher order aggregates, including ternary and quaternary, are possible. Binary aggregation is, however, much more likely in dilute systems than higher order aggregations, hence we will only take binary aggregation into consideration while assuming a dilute system.

Particle pair formation is necessary for binary aggregation. Therefore, we must provide the numerical density of particle pairs in order to represent the aggregation process. The number density of particles—or, more specifically, singlets—was previously described as a function of both their internal and external coordinates, and represents the number of particles confined in an area at a particular moment of time divided by the bin size, i.e., it is independent of the way of observation. A closer look at number balance and density would be necessary before we move forward.

3.1 Number Balance

As a size range's width widens, so does the number of particles within it, which has the appearance of an extensive property in bin width. This is comparable to mass having a wide range of properties in space. A point cannot contain any mass since it has zero volume by definition. Similar to this, a point on the size axis has no "volume" and is therefore unable to contain any particles.The number of particles in a particular area and size range divided by the area's volume and the number density of particles may be defined as the size range's width. The number density converges to a fixed value as the bin size decreases. If N_x is the quantity of particles in the size range of x to x + dx in the volume dV, then we will define the number density as:

$$f_1(x,t) = \frac{N_x}{dxdV}$$

The differential range of each internal coordinate should be utilised if the number density is defined using more than one internal coordinate. Consequently, the relationship between number and number density[1] becomes:

$$N(x,r) = f_1(x,r)dV_x dV_r$$

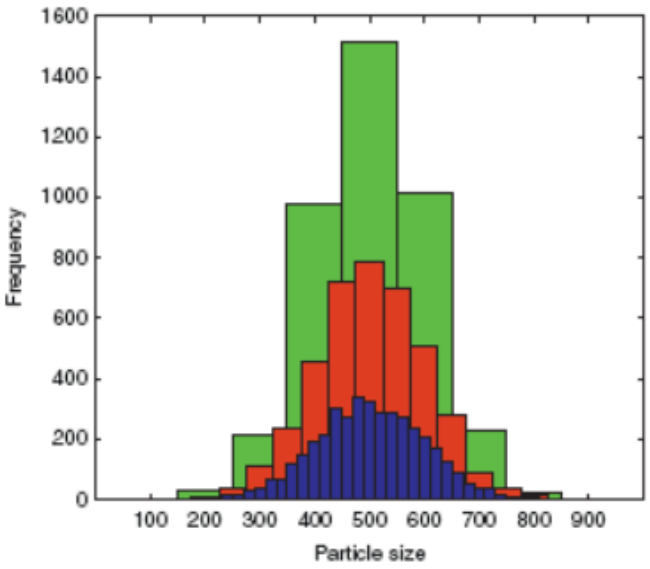

Figure 3.1: The same sample's particle size distribution is measured using three different bin sizes. The number of particles in each bin decreases as the bin size decreases.[1]

3.2 Aggregation Equation

The number density of particle pairs which are about to aggregate can now be written as $f_2(x, x', r, r', t)$. The number density function's subscript 2 indicates that this is the density of pairs rather than singlets. A scalar assumption has been made about the internal coordinate in an effort to simplify the equation. The internal coordinate can be interpreted as particle volume if nothing else is specified. Although we adopt these simplicities, the following treatment is universal. Also take note of the fact that we save the exterior coordinates since particle closeness is necessary for aggregation. The quantity:

$$f_2(x, x', r, r', t)dxdx'dV_r dV_r'$$

represents the number of pairs in which one particle is in x+dx and other will be in x'+dx', and the first particle in the pair is located in the volume dV_r around r and similarly with the second particle at location r'.If system snapshots can be taken, this data may be acquired.

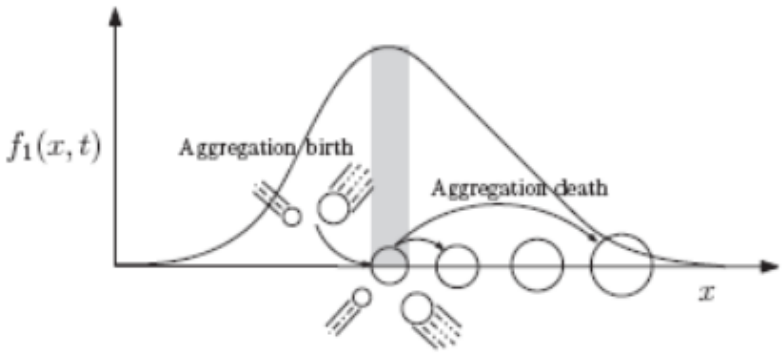

Figure 3.2: Pure aggregation resulting in birth and death of particles[1]

As the derivation is quite rigorous, it is skipped but it can be referred if needed[1]. We arrive at the final number density PDE which depends on the birth of the particles in the domain x+dx as well as death of particles which are already in the domain but due to size decrease leave the domain:

$$\frac{df_1(x,\mathbf{r},t)dxd\mathbf{V}_\mathbf{r}}{dt} =$$
$$\frac{1}{2}\int_{\Omega_{x'}}\int_{\Omega_{r'}} a(x',\tilde{x},\mathbf{r'},\tilde{\mathbf{r}},y(t))f_2(x',\tilde{x},\mathbf{r'},\tilde{\mathbf{r}},t)dx'd\tilde{x}d\mathbf{V}_{\mathbf{r'}}d\mathbf{V}_{\tilde{\mathbf{r}}}$$
$$-\int_{\Omega_{x'}}\int_{\Omega_{r'}} a(x,x',\mathbf{r},\mathbf{r'},y(t))f_2(x,x',\mathbf{r},\mathbf{r'},t)dxdx'd\mathbf{V}_\mathbf{r}d\mathbf{V}_{\mathbf{r'}}.$$

f_1 is the number density and this equation relates it to the birth and death of particles. Assuming that particles at $\tilde{x}$ and $\tilde{r}$ will aggregate with particles at x' and r' to produce particles at x and r, we have the function 'a' is the aggregation frequency as not all particles that collide will be aggregating. While the first term is for the birth term of particles forming inside the bin, the negative term indicates the death of particles leaving the bin hence decreasing the number density.

We require some simplications in the above formula with well defined integration boundaries which can be done so using Jacobian Matrices. After simplifications, we end up with the final equation:

$$\frac{\partial f_1(x,t)}{\partial t} = \frac{1}{2}\int_{x'=0}^{x'=x} a(x-x',x')f_1(x-x',t)f_1(x',t)dx'$$
$$-\int_{x'=0}^{x'=\infty} a(x,x')f_1(x,t)f_1(x',t)dx'.$$

A point to be noted, that for further cases, we will assume a constant aggregation kernel that is a(x,y) = 1, as this greatly simplifies the equation without hampering the results[6].

Numerical Methods

The preceding chapter's formulation of the population balance equation (PBE) resulted in a complex equation. However, it is possible to come up with an analytical solution for a few simplified examples, and these answers are crucial since they serve as a baseline for the numerical solution.
There are several numerical methods that may be used to solve PBEs. For the solution of PBEs, discretization methods are a particularly well-liked and flexible numerical methodology. Often used concepts include finite difference, finite volume, and finite components. The quadrature method of moments is another effective way for numerically solving the PBE. The most used is the fixed pivot method, which will be the focus of discussion moving on.

4.1 Fixed Pivot Method using Finite Volume

The internal coordinate space must be divided into smaller chunks as the initial stage in discretizing the PBE. The particle size space covers several orders of magnitude in the majority of scenarios of practical significance. An alternative to a linear grid is a geometric grid. The smallest node in a geometric grid is found at a very tiny size, and to get the next node, each succeeding node point is multiplied by a ratio (r). As a result, the size axis' nth node will be at $v_1 r^n$. In finite volume discretization, the PBE is reduced to a collection of ODEs by integrating the equation across the limited volume of the bin. The quantity of particles on nodes consequently serves as a representation of the number density distribution. These nodes, which are often referred to as "pivots," are situated in the middle of each bin.[7]

Let's take the aggregation of two particles with sizes x_j and x_k as an example. Aggregation is often described as the transformation of the property f(x) from f(x_j) + f(x_k) to f(x_i+ x_k), where f(x) is an extensive property that can be obtained from x, the particle's size. The size of a new aggregate perfectly fits one of the xi's in the context of strictly discrete populations and uniform grid (X_i = ix_i). As a result, changes in any f(x) attribute associated with the aggregation of two particles are precisely maintained for this discretization.

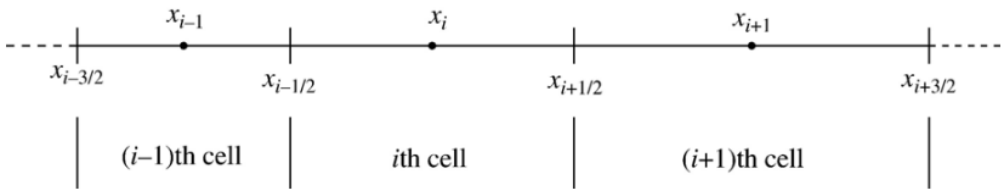

Figure 4.1: A discretized size domain[2]

Assigning fractions a(v,x_i) and b(v,x_{i+1}) to particle populations at x_i and x_{i+1}, respectively, represents the breakage or aggregation that results in the production of a particle of size v in the size range x_i, x_{i+1}. These fractions meet the following equations in order for two generic qualities, $f_1(v)$ and $f_2(v)$, to be conserved:

$$a(v, x_i)f_1(x_i) + b(v, x_{i+1})f_1(x_{i+1}) = f_1(v)$$

Hounslow[8] created a novel discretization method (vi=2v_{i-1}) that conserved mass, did not need the assessment of any double integrals, and guaranteed the proper development of the total number of particles. Hounslow[8] found pertinent events that alter the size distribution under the assumption of a stepwise uniform number density distribution. They appropriately accounted for numbers when constructing the discretized equations, but little effort was made to conserve volume (mass). By including a correction factor, mass conservation was made possible, and its value was calculated by forcing mass conservation. This element proved to be kernel-independent.[**?**]

The main idea of fixed pivot technique was first given by Kumar and Ramkrishna[7]. The discrete sizes that have been determined in advance as the fixed pivots indicate the population of particles that are continually dispersed. Thus, the equation is

formed:

$$n(v,t) = \sum_i N_i(t)\delta(v - v_i)$$

The new idea of internal consistency of discretization is the technique's main focus. To get the required moments (or general features) of the size distribution, one can modify the discretized set of equations for a PBE algebraically. The equations produced above must be identical to those derived by discretizing the equations for the required qualities directly, not only in the case of a very small grid but also for an arbitrarily coarse grid. This is necessary for internal consistency of discretization. In order to retain the characteristics of the particles that correspond to the required moments of the size distribution, the events leading to the production of non-pivot particles were represented in a way that ensured internal consistency.

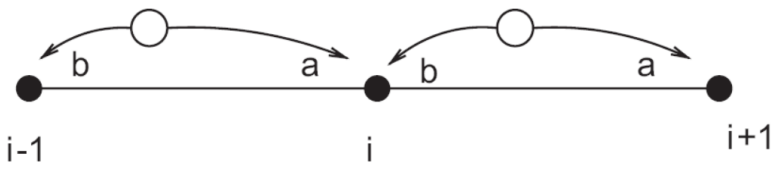

Figure 4.2: Assignment of non-pivot particles of volume v to the neighboring pivots x_{i1} and x_i in the fixed pivot technique of Kumar and Ramkrishna[3]

According to Fig. 4.1, if fractions a and b are given to the pivots to the right and left of the non-pivot particle, respectively, then

$$a + b = 1, ax_i + bx_{i-1} = v$$

To increase the accuracy of the numerical solution, pivots might be scattered sparsely in certain areas and densely in others. For the pure aggregation process, we have a discretized equation that is:

$$\frac{dN_i}{dt} = \sum_{j,k}^{j>=k} (1 - \frac{1}{2}\delta_{j,k})\eta Q_{j,k} N_j N_k - N_i \sum_{k=1}^{M} Q_{i,k} N_k$$

where,

$$\eta = \begin{cases} \frac{x_{i+1}-v}{x_{i+1}-x_i} & \text{if } x_i <= v <= x_{i+1} \\ \frac{v-x_{i-1}}{x_i-x_{i-1}} & \text{if } x_{i-1} <= v <= x_i \end{cases}$$

The fixed pivot approach has been shown to be highly reliable when used as the foundation for the creation of numerical algorithms for the modelling of systems including

additional phenomena. In the size range where number density quickly declines with particle volume, the fixed pivot strategy tends to over-predict size distribution when paired with coarse dispersion of pivots. The percentage of the population that falls within this size range is frequently extremely modest and is not of concern. However, unlike the prior strategies in this class, this problem may be quickly fixed if necessary by using densely dispersed pivots in the tail area.

4.2 Fixed Pivot in 2D Grid

The computational domain is assumed to be a rectangle [x_min, x_max] X [y_min, y_max] covered by cells C_{ij} [$x_{i-1/2}$, $x_{i+1/2}$] X [$y_{i-1/2}$, $y_{i+1/2}$] assuming a Cartesian grid. with The values 1 ¡= i ¡= M_x and 1 ¡= j ¡= M_y. The population's pivots (representative coordinates) in cell C_{ij} are marked by (x_i, y_j). Figure 4.2 depicts a typical grid for the purpose of clarity. It is important to emphasise that the grid might be irregular in both coordinate directions.[4]

The two close pivots are given a new particle, and the two unknown fractions are calculated by enforcing the conservation of an equal number of attributes, according to the fixed pivot approach for one-dimensional distributions. Which pivots will be utilised and, consequently, what attributes will be maintained, are undoubtedly the first issues that must be resolved in order to expand this technique to bidimensional distributions.

After choosing the four pivots that surround a new particle—namely, the points (x_i, y_j), (x_i, y_{j+1}), (x_{i+1}, y_j), and (x_{i+1}, y_{j+1})—so that $x_i < \hat{x} <$ and x_{i+1} are formed, and $y_j < \hat{y} <$ and y_{j+1} are formed, respectively, where (x, y) are the internal coordinates of the new particle. As seen in Figure 4.2, the preservation of four qualities will allow us to calculate the four fractions a, b, c, and d. Choosing which pivots to divide the particle into efficiently is debatable, but there are multiple examples of different approaches to this. This results in the conservation of four properties:[9][10]

$$f_n(\hat{x}, \hat{y}) = af_n(x_i, y_j) + bf_n(x_{i+1}, y_j) + cf_n(x_i, y_{j+1}) + df_n(x_{i+1}, y_{j+1})$$

Internal consistency in terms of the number of particles and the mass of each component is typically the key concern, and it is entirely consistent with the technique

presented here. For such situations, the following qualities are perhaps the most intriguing:

$$f_1(x,y) = 1, f_2(x,y) = x, f_3(x,y) = y, f_4(x,y) = xy$$

It is clear from the assignment mechanism outlined above that any particle created in any of the four surrounding rectangles will contribute to the pivot (x_i, y_j).

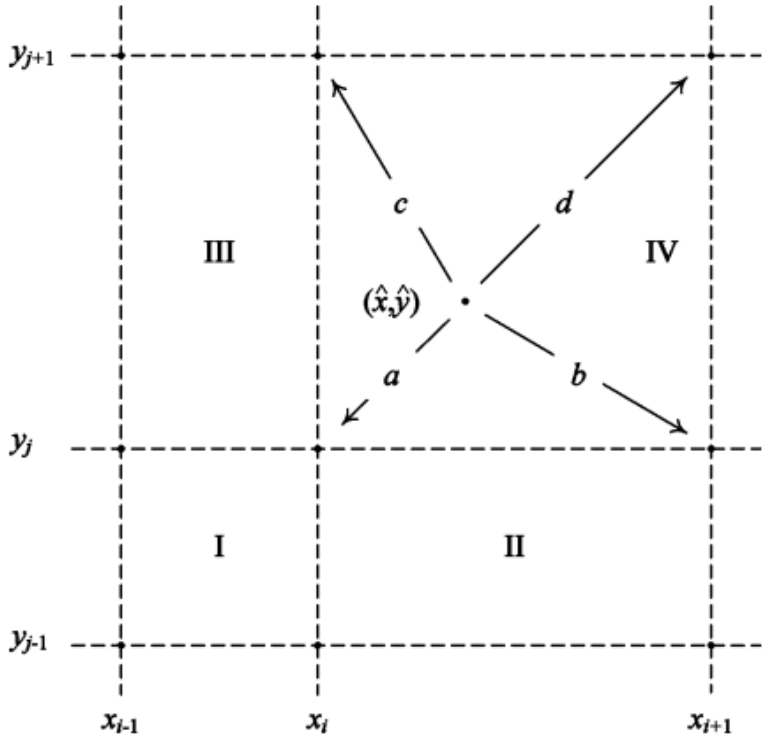

Figure 4.3: Assigning a newly generated particle as a result of aggregation. Every fractional particle born in the rectangles I, II, III, or IV is given to the pivot (x_i, y_j).[4]

This results in the following set of equations after simplification for the case of 2D Pure Aggregation using Fixed Pivot Method:

$$\frac{dN_{ij}}{dt} = \sum_{k,l,q,r \in \Omega_{ij}} (1 - \frac{1}{2}\delta_{kq}\delta_{lr})\eta_{ij}(\hat{x},\hat{y})\beta_{kl,qr}N_{kl}N_{qr} - N_{ij}\sum_{k,l}\beta_{ij,kl}N_{kl}$$

where δ_{ij} is the Kronecker Delta and,

$$\Omega_{ij} = \{\{k,l,q,r\} : 1 \leq k \leq i, 1 \leq l \leq j, l \leq r \leq j, 1 + \delta_{rl}(k-1) \leq q \leq i, (\hat{x},\hat{y}) \in R_{ij}\}$$

$$\beta_{ij,kl} = \beta(x_i, y_j; x_j, y_l; t)$$

The one-dimensional expression provided by Kumar and Ramkrishna is naturally extended by this discretized PBE. Only once, during the grid creation process, are the various combinations of cells C_{kl} and C_{qr} that might result in the development of a new particle in the area R_{ij} calculated. Similarly, the time-independent values of $\eta_{ij}(x, y)$ are computed at the start of the calculations and then saved in an interaction matrix.

4.3 Subdivision of space: Triangulation Method

This method is listed for the sake of completion. No progress has been done in implementing the below method, but will be worked on in the future.

The above method used the preservation of 2^n properties in the case of n-d PBE's, which makes it highly computational in the case of higher n. The triangular method proposes to use only n+1 properties for any n-d PBE while resulting in minimal loss in accuracy.[5]

Delaunay triangulation is created by joining natural pivots; the resulting forms are referred to as natural elements. Tetrahedrons become natural elements for 3-D spaces as a result of this process, while triangles become natural elements for 2-D spaces. The smallest number of points in n-d space that can create a shape that encloses an n-d region are provided by naturally occurring elements with n + 1 vertices. Therefore, natural elements distribute the populations of non-pivot particles to the smallest number of pivot populations—n+1 pivots for natural elements vs. 2^n pivots for hypercuboid elements when it comes to representing a non-pivot particle through pivots.
The benefits of using triangulation are binary: (i) In order to represent non-pivot particles, fewer pivots are needed, which leads to decreased dispersion, and (ii) variable distribution of pivots that are dense in some local areas or along arbitrary curves and sparse elsewhere in n-d space. For the solution of multidimensional PBEs, the discretization approach employed in the fixed pivot technique for 1-d PBE holds true. The creation of pivots on which the particle population is represented is the initial stage in the discretization process. Using the following approximation for number

density:

$$n(v,t) = \sum_{k=1}^{M} N_k(t)\delta(v - x_k)$$

where **v** and x_k are n-d vectors, x_k represent location of pivots. We must "triangulate" the pivots since their distribution is so broad and flexible. This process entails combining (n + 1) pivots together (in n-d space) to create elements, then indexing those elements.[5]

Thus, triangulated elements stand in for domains that have n + 1 pivots as vertices. The particle population of the n + 1 pivots that surround it serves as a representation for every non-pivot particle that enters this domain as a result of particulate activities. An essential stage is figuring out which element a freshly generated particle belongs to. This conclusion is relatively easily reached for the 1-d case by resolving inequalities like $x_{i-1} < (x_j + x_k) < x_{i+1}$. Such a search in higher dimensional space is not easy since one has to know which triangle or tetrahedron, out of many others present, has the specified point $(x_j + x_k)$, inside of it. However, only one search has to be done at the moment pivots are generated. This results in the following final equation:

$$\frac{dN_i}{dt} = \sum_{j,k}^{j>=k} (1 - \frac{1}{2}\delta_{j,k})\eta_{j,k}^{i} Q_{j,k} N_j N_k - N_i(t) \sum_{j} Q_{i,j} N_j(t)$$

Here, the coefficient matrix is denoted as $\eta_{j,k}^{i} = f_i(x_j + x_k, x_i)$. When particles represented by the jth and kth pivots aggregate and maintain the P_j attributes of the aggregated particle, it indicates the portion of the particle that is assigned to the population at the ith pivot, where j = 1, 2,..., (n+1). The matrix $\eta_{j,k}^{i}$ is undoubtedly the technique's fundamental component. The problem's dimensionality has no effect on the structure of the solution. In theory, this matrix may be formed together with the grid and maintained indefinitely for use because its components are independent of the shape of the aggregation kernel or the changing population of the particles.[1]

System Description and Results (1D)

We start with some initial total number of particles (N_0) which go on to aggregate into different bins. For the sake of simplification we have taken (N_0)=1. We assume a constant aggregation kernel. We have contrasted our numerical findings with the analytical answers for the constant aggregation kernels and the following gamma initial condition:

$$n(v,0) = \frac{N_0}{v_0}\frac{v}{v_0}exp(\frac{-v}{v_0})$$

Using geometric grids for three different values of parameter s, the numerical findings for the preservation of $\zeta = 0$ (numbers) and v = 1 (mass) have been produced. The smallest particle analysed in this study is 2000 times smaller in size than the original average particle size. The quantity of particles and the percent mass contained in particles with a diameter less than this have little bearing on the numerical outcomes.

5.1 Variables

Table 5.1: Variables and Parameters

Parameter/Variable	Symbol	Numerical Value
Initial Total Number	N_0	1
Initial Average Size	v_0	0.01
Maximum size/volume	v_{max}	10000.0
Minimum size/volume	v_{min}	0.000005
Time span for integration	t_{span}	[0,2]
Aggregation Kernel	a_0	1
Geometric Ratio	r	2

5.2 Simulation

This simulation was done in C++ as it provides a much faster computation time in comparison to Matlab code, which has been compared too according to Figure 5.1 which shows the numerical density values at different pivots. The results are also extremely close but there has been a 10x decrease in computation times (from 0.256542 sec to 0.024586 sec). This was extremely necessary as we shall see, for a finer grid (smaller geometric ratio) we get much better results, but the computation time required increases by a lot. We can afford to get much better results now in the same time by using this simulation code.

While MATLAB excels in terms of portability and "easy of programming," "performance" has been proven to be an area where it falls short. Modern scientific applications have very high memory needs, and MATLAB itself uses a significant fraction of the system memory, which contributes to this issue.

```
1   0.000015 2.498750e+01
2   0.000030 2.496876e+01
3   0.000060 2.493753e+01
4   0.000120 2.487510e+01
5   0.000240 2.475040e+01
6   0.000480 2.450162e+01
7   0.000960 2.400675e+01
8   0.001920 2.302923e+01
9   0.003840 2.113308e+01
10  0.007680 1.763325e+01
11  0.015360 1.198623e+01
12  0.030720 5.397294e+00
```

Figure 5.1: Matlab (0.256542 sec)

```
1   0.000015 2.508787e+01
2   0.000030 2.506850e+01
3   0.000060 2.503587e+01
4   0.000120 2.497070e+01
5   0.000240 2.484069e+01
6   0.000480 2.458206e+01
7   0.000960 2.407019e+01
8   0.001920 2.306739e+01
9   0.003840 2.114369e+01
10  0.007680 1.762932e+01
11  0.015360 1.198440e+01
12  0.030720 5.397487e+00
```

Figure 5.2: C++ (0.024586 sec)

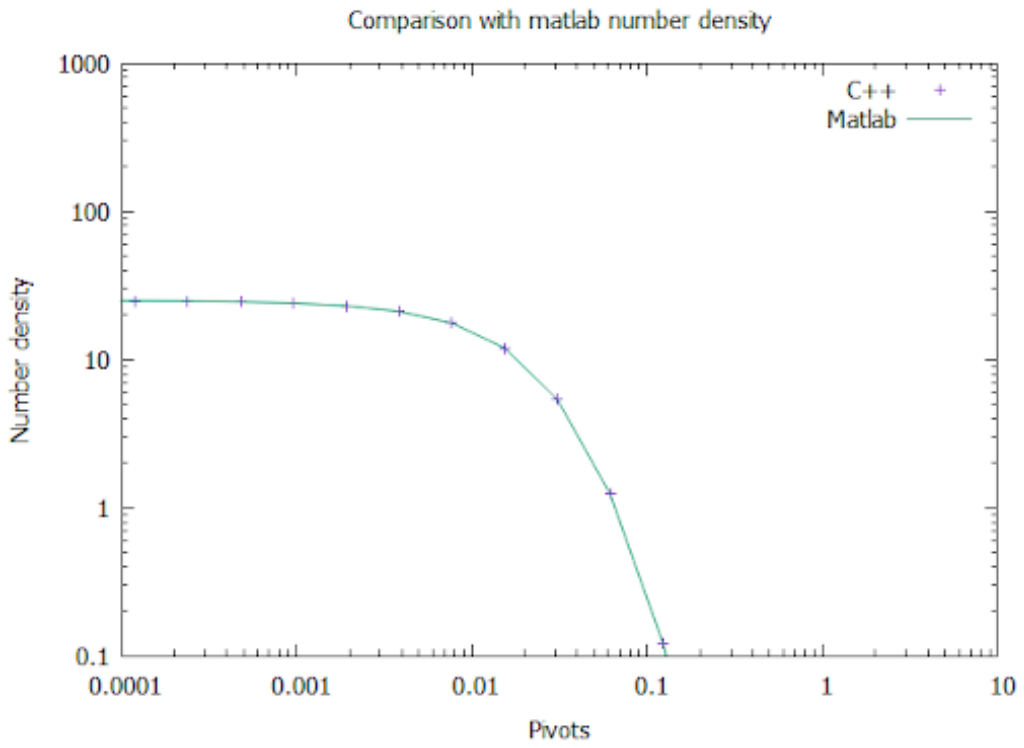

Figure 5.3: Comparison of Numerical Density in C++ and Matlab

The code also has a caller function where a user can easily change the above listed variables and can run the code to get results on the go.

```
double r=2; //Geometric Ratio
int N0=1; //Initial Total Number
double v0 = 0.01, v_max = 10000.0, v_min = 0.000005; // Initial Average Size, Maximum size/volume
double t_start = 0; //Time span start (for integration)
double t_end = 10000; // Time span end (for integration)
double a0 =1.0; // Constant Kernel
```

Figure 5.4: Caller function with user variables

I solved the above-discussed 1D Pure Aggregation Fixed Pivot ODE equation, and gathered the initial, numerical, and analytical numerical densities of the particles. We compare the results with Kumar and Ramakrishna and get similar results.

The analytical equation that is being used to compare the results is written as:

$$
\begin{aligned}
& n(x_1, x_2, \ldots, x_m, t) \\
& \quad = \frac{4N_0}{(\tau+2)^2} \prod_{i=1}^{m} \frac{(p_i+1)^{(p_i+1)}}{x_{i0}} \\
& \quad \times \exp\left[\frac{-(p_i+1)x_i}{x_{i0}}\right] \times \sum_{k=0}^{\infty} \left(\frac{\tau}{\tau+2}\right)^k \\
& \quad \times \prod_{j=1}^{m} \frac{[(p_j+1)^{(p_j+1)}]^k (x_j/x_{j0})^{(k+1)(p_j+1)-1}}{\Gamma[(p_j+1)(k+1)]}.
\end{aligned}
$$

where we can keep the value of p_i=1 for the case of gamma initial condition. The equation has not been plotted due to a lack of space in the diagram, but it matches the results perfectly.

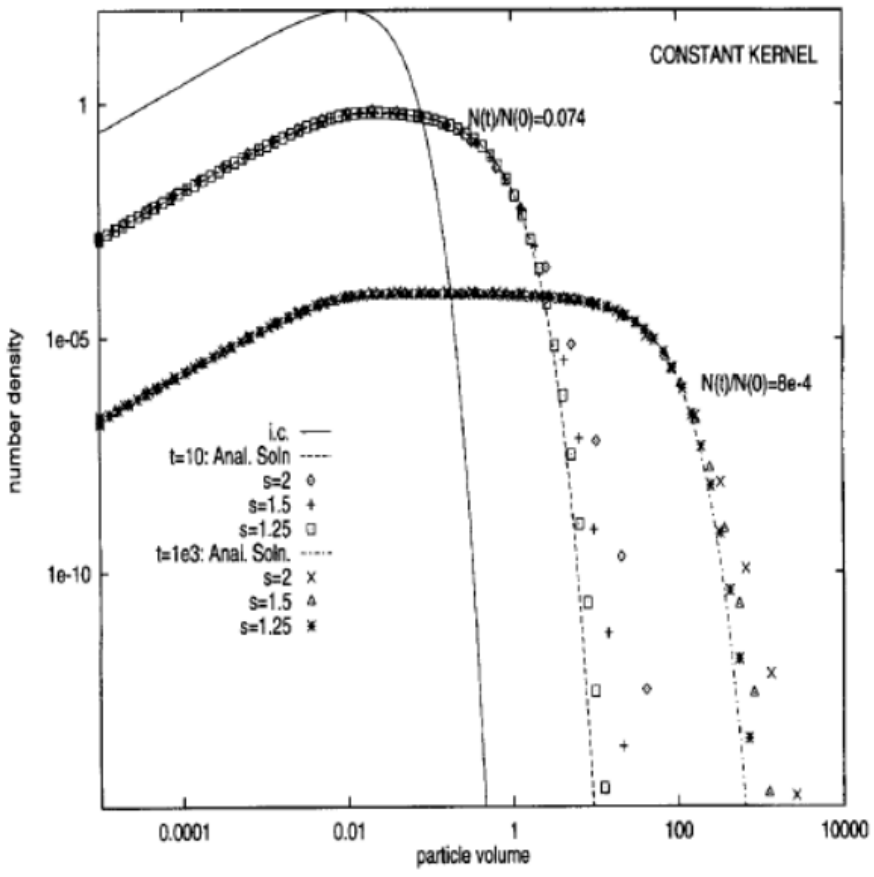

Figure 5.5: Results from Kumar and Ramakrishn[4]

The full size distributions for the constant kernel at two degrees of aggregation [defined as N(t)/N(O)] and for three choices of grid parameter s are shown in Figure 5.5. (2, 1.5 and 1.25). To emphasise the differences from the analytical conclusions in the size range where the number density declines sharply, the findings have been shown on a log-log scale. These variations are not visible when using a linear scale. The picture demonstrates that even for the coarsest size grid, the numerical findings are in great agreement with the analytical results for tiny to moderate particle size ranges. The numerical findings for a coarse grid are overpredicted for big particle sizes where number densities are low.

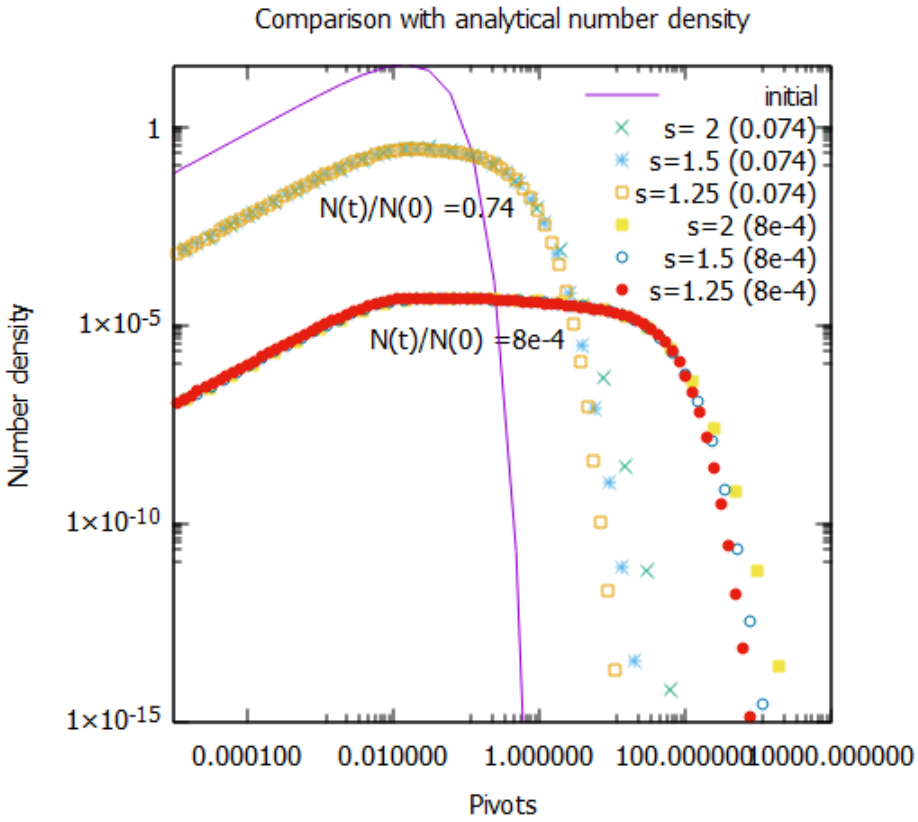

Figure 5.6: Our Results

From figure 5.5 and 5.6 it can be seen that there is a minute difference in the values, and hence we are able to safely recreate the results in C++. However, the image demonstrates that even in this small range, the fluctuation of number density may be quite accurately anticipated by using a finer grid. This is well demonstrated by

the very strong agreement for s = 1.25, both for low and high degrees of aggregation, spanning several orders of magnitude. It should be noted that the level of overprediction for s = 1.5 (a moderate grid) is far lower than that for s = 2, and that this grid may offer sufficient accuracy for applications that only require precise information in the highly populated size ranges. Of course, a finer grid may always be utilised for more accuracy by simply decreasing the value of the s parameter.

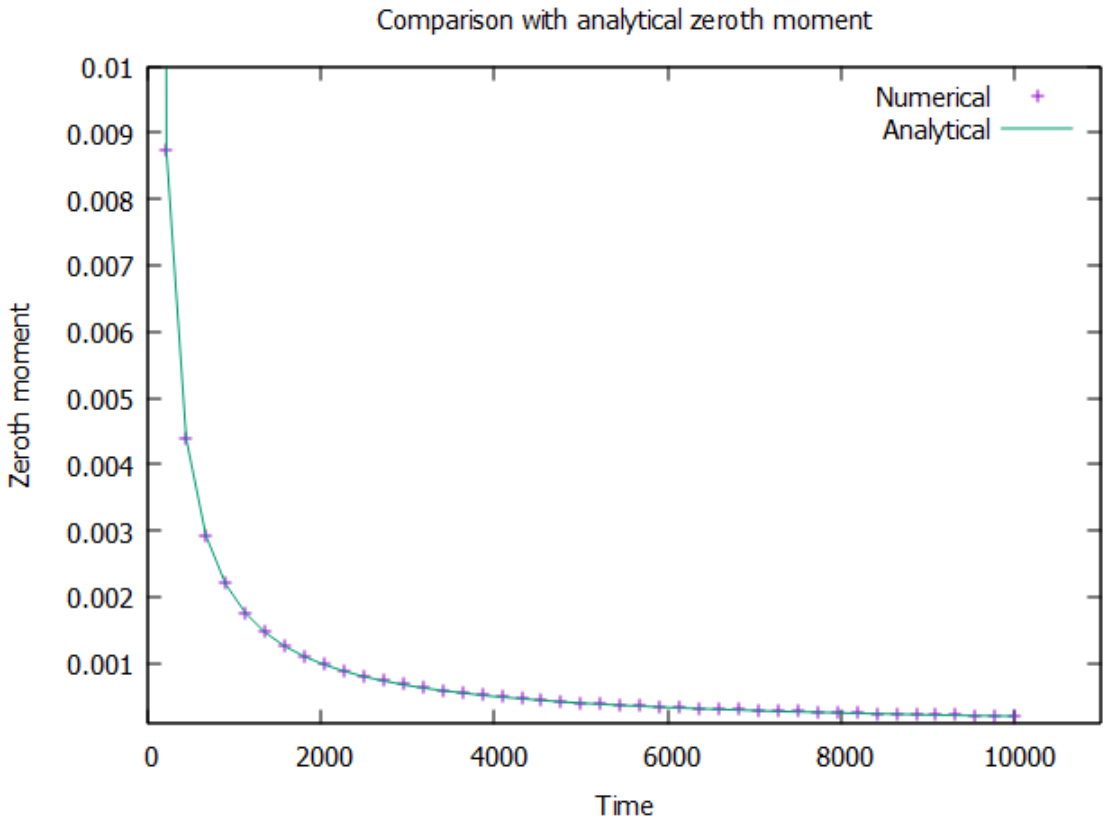

Figure 5.7: Analytical vs Numerical results for zeroth moment

As we can see from figure 5.7, the zeroth moment is almost similar when comparison between analytical and numerical methods is done.

Conclusions

In our work, we studied the pure aggregation of particles and explored different numerical methods to solve them.

The important points from this study are:

1. Aggregation is highly dependent on the initial number of particles and on the distribution of the numerical density of the existing particles.

2. Numerical density is a key intensive property to represent the number of particles of a particular size in an area.

3. As the solution of pure aggregation is a double integral equation, we require simplifications in it before simulation. There are multiple numerical methods, primarily the fixed pivot method.

4. The fixed pivot method divides the internal attributes of particles among fixed pivots while maintaining consistency in the total value. This helps in discretization of the equation, and makes the equation much easier to solve.

5. The new term introduces a variable, $\eta^i_{j,k}$ which represents the values of internal coordinates when particles at jth and kth pivot aggregate to form particle at ith pivot. This was the crux of the numerical method, as this can be solved in advance before solving the ODE, and hence saves a huge chunk of computation time.

6. The results when compared show minimal discrepancy with respect to existing

results while gaining a significant loss in computation time, allowing for much complex cases to be analyzed.

7. The results also showed that for higher size, the numerical results were not the most accurate, but since the number of particles at that numerical density is lower, it does not affect the solution significantly.

8.It was also shown that by choosing a finer grid (by having more pivots to represent the same data) we can increase the accurac [7][3][11] with respect to analytical solutions, but it comes with a huge computation time increase.

Future Work

There is much to be done in this research of Pure Aggregation Simulation:

7.1 Other Numerical Methods

The first work that needs to be completed is to complete the 2D fixed pivot technique and then move onto the n-D technique which requires only n+1 properties. The complexity of the method of n+1 properties method makes it a challenging task to simulate and compare with the existing results.[5]

This method will also use mesh generating software like Fluent so that n-D equations can be visualized correctly before the simulation. [5][9]

7.2 Complex Aggregation Kernel

For the sake of simplicity, we have solved the equation for aggregation kernel a_0=1, but in multiple real life population balances we have complex aggregation kernel. Attempts will be made to solve them as well, starting from degree 1 (sum) and then for degree 2 as well (product). [4]

7.3 Use of CUDA

As the code is written in C++ we can make use of CUDA libraries which works on using the GPU of computers in intensive computation and decreasing the computation time severalfold. Taking the help from Rohit, Anwesha and Anuj, the parallelization

can be implemented while solving the ODE as well as creating the grid and eta functions.

The scientific community, which deals with larger and more complicated issues, as well as the industrial community, which uses process models to examine its processes in silico, have always needed speed in addition to increasing numerical accuracy. This is demonstrated by the fact that lengthier run durations are seen even in a high-level language environment like MATLAB as the computational burden grows practically polynomially with system complexity.

In an effort to increase calculation efficiency, researchers regularly upgrade their hardware to include the newest CPUs and larger amounts of RAM, but the majority of them do not create codes that fully take advantage of the parallel processing abilities of the most recent multicore/multiprocessor CPUs. The limitation of running a code sequentially on one core can be overcome, and the programmer can also take advantage of other massively parallel processors like the GPU (Graphics Processing Unit), which reduces computation time significantly, by parallelizing an existing code.[12]

Nomenclature

δ	Kronecker Delta
η	Coefficient Matrix
a_0	Aggregation Kernel
N	Number of particles
n	Number density
N_0	Initial Total Number
r	Geometric Ratio
t_{span}	Time Span for integration
v_0	Initial Average Size
v_{max}	Maximum size/volume
v_{min}	Minimum size/volume

Code

The code that is written for this report can be accessed through the following link:

MTP-1 Code

The 1D code is complete but the 2D code is still under work. Instructions for running the code are provided in the link.

Bibliography

[1] J. Chakraborty. *Engineering of submicron particles*. Wiley, 2019.

[2] Laurenzi I. J., Bartels J. D., and Diamond S. L. A general algorithm for exact simulation of multicomponent aggregation processes. *J. Comput. Phys*, 177, 418-449, 2002.

[3] Kumar S. and Ramkrishna D. On the solutions of population balance equation by discretization-ii. a moving pivot technique. *Chemical Engineering Science*, 51, 1333–1342, 1996.

[4] Vale H. and McKenna T. Solution of the population balance equation for two-component aggregation by an extended fxed pivot technique. *Ind Eng Chem Res*, 44(20):7885–91, 2005.

[5] Chakraborty J. and Kumar S. A new framework for solution of multidimensional population balance equations. *Chemical Engineering Science*, 62, 4112–4125, 2007.

[6] Pilinis C. Derivation and numerical solution of the species mass distribution equations for multicomponent particulate system. *Atmospheric Environment*, 24A (7), 1923–1928, 1990.

[7] Kumar S. and Ramkrishna D. On the solutions of population balance equation by discretization-i. a fixed pivot technique. *Chemical Engineering Science*, 51, 1311–1332, 1996.

[8] Hounslow M. J., Ryall R. L., and Marshall V. R. A discretized population balance for nucleation, growth and aggregation. *A.I.Ch.E.J.*, A.I.Ch.E.J., 1988.

[9] J. Kumar, G. Warnecke, M. Peglow, and S. Heinrich. An efficient numerical technique for solving population balance equation involving aggregation, breakage, growth and nucleation. *Powder Technol.*, 179, 205–228, 2008.

[10] Chi Nguyen-Thanh, Vinh Phu Nguyen, A. Vaucorbeil, T. Mandal, and Jian-Ying Wu. Jive: An open source, research-oriented c++ library for solving partial differential equations. *Adv. Eng. Softw.*, 122,103-112, 2020.

[11] Kumar S. and Ramkrishna D. On the solution of population balance equation by discretization-iii. nucleation, growth and aggregation of particles. *Chemical Engineering Science*, 57, 4659–4679, 1997.

[12] Prakash A.V. and Ramachandran R. Chaudhury A. Parallel simulation of population balance model-based particulate processes using multicore cpus and gpus. *Model. Simul. Eng.*, 2, 2013.