

January 2015

# GRANULATION OF ULTRA-FINE POWDERS: EXAMINATION OF GRANULE MICROSTRUCTURE, CONSOLIDATION BEHAVIOR, AND POWDER FEEDING

Nathan Bellemore Davis  
*Purdue University*

Follow this and additional works at: [https://docs.lib.purdue.edu/open\\_access\\_dissertations](https://docs.lib.purdue.edu/open_access_dissertations)

---

## Recommended Citation

Davis, Nathan Bellemore, "GRANULATION OF ULTRA-FINE POWDERS: EXAMINATION OF GRANULE MICROSTRUCTURE, CONSOLIDATION BEHAVIOR, AND POWDER FEEDING" (2015). *Open Access Dissertations*. 1108.  
[https://docs.lib.purdue.edu/open\\_access\\_dissertations/1108](https://docs.lib.purdue.edu/open_access_dissertations/1108)

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact [epubs@purdue.edu](mailto:epubs@purdue.edu) for additional information.

**PURDUE UNIVERSITY  
GRADUATE SCHOOL  
Thesis/Dissertation Acceptance**

This is to certify that the thesis/dissertation prepared

By NATHAN BELLEMORE DAVIS

Entitled

GRANULATION OF ULTRA-FINE POWDERS: EXAMINATION OF GRANULE MICROSTRUCTURE,  
CONSOLIDATION BEHAVIOR, AND POWDER FEEDING

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

James Litster

Chair

Carl R. Wassgren

Michael T. Harris

Stephen P. Beaudoin

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): James Litster

Approved by: John A Morgan

Head of the Departmental Graduate Program

12/4/2015

Date

GRANULATION OF ULTRA-FINE POWDERS: EXAMINATION OF GRANULE  
MICROSTRUCTURE, CONSOLIDATION BEHAVIOR, AND POWDER FEEDING

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Nathan B. Davis

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2015

Purdue University

West Lafayette, Indiana

## ACKNOWLEDGEMENTS

Thank you to my supervisors, research group members, and family. This work would not have been possible without your support.

To Jim, thank you for your optimism throughout my time working with you. It helped me to remain positive and confident as we worked through the different issues that confronted my project, both research and funding related. You have helped me to become a better writer and researcher and have been a good role model for me as I start my scientific career.

To Steven Dale, thank you for all your help with the analysis technique and code writing. You were incredibly patient and helpful in answering all my questions and teaching me how to operate in the lab. Thank you for everything you've done for me.

To the rest of the Litster lab group, thank you for your thoughts and assistance over the last several years. You've been willing to help and discuss my questions and concerns and made me a better researcher in the process. I wish you all the best.

To Karis and Kelly, thank you for the work you did as undergraduate researchers. Your skills and work helped me in many ways. It let me focus on other aspects of the project, but also forced me to organize everything so that I could properly explain my work. I wish you both well in your future endeavors.

## TABLE OF CONTENTS

	Page
CHAPTER 1. INTRODUCTION.....	1
1.1 Background .....	1
1.2 Thesis Objectives.....	6
1.3 Thesis Outline.....	7
CHAPTER 2. LITERATURE REVIEW.....	9
2.1 Overview .....	9
2.2 Dimensionless Groups and Regime Maps.....	9
2.2.1 Granule Nucleation Theory.....	9
2.2.2 Coalescence and Consolidation .....	14
2.2.3 Consolidation and Coalescence models .....	16
2.3 Granulation of Ultra-Fine Powders .....	23
2.4 Suspension Based Agglomeration.....	28
2.5 Fluidization of Ultra-Fine Powders.....	31
2.6 Review of X-Ray Computed Tomography .....	32
2.6.1 Introduction .....	32
2.6.2 X-Ray Absorption .....	33
2.6.3 Image Resolution .....	35

	Page
2.6.4 Potential Image Artifacts .....	37
2.6.5 Beam Hardening .....	38
2.6.6 Photon Starvation .....	39
2.6.7 Metallic Materials .....	40
2.6.8 Object Motion Artifacts .....	42
2.6.9 Ring Artifacts.....	43
2.7 Quantitative XRCT Analysis of Granule Microstructure.....	43
2.8 Loss-in-Weight Feeding.....	49
2.9 Critical Summary .....	51
CHAPTER 3. MATERIAL AND METHODS.....	54
3.1 Material Properties .....	54
3.2 Static Bed Granulation .....	55
3.3 Drum Granulation .....	58
3.4 Image Analysis.....	60
3.4.1 XRCT Analysis .....	60
3.4.2 Image Preparation .....	60
3.4.3 Granule Analysis .....	62
CHAPTER 4. STATIC BED GRANULATION .....	66
4.1 Introduction .....	66
4.2 Experimental Plan .....	66
4.3 Results .....	66
4.3.1 Powder Characteristics .....	66

	Page
4.3.2 Analysis of Granule Structure .....	69
4.3.3 Effects of Primary Particle Size .....	74
4.3.4 Effects of Bed Preparation .....	76
4.4 Discussion.....	78
4.5 Conclusions .....	81
CHAPTER 5. DRUM GRANULATION .....	83
5.1 Introduction .....	83
5.2 Theory .....	83
5.3 Experimental Plan .....	84
5.4 Results & Discussion .....	85
5.4.1 Analysis of Granule Structure .....	85
5.4.2 Granule Consolidation with Time .....	89
5.4.3 Effect of Binder Viscosity .....	94
5.4.4 Discussion .....	97
5.5 Conclusions .....	101
CHAPTER 6. POWDER FEEDING .....	104
6.1 Introduction .....	104
6.2 Materials & Methods .....	104
6.3 Results & Discussion .....	108
6.3.1 FT4 Results .....	108
6.3.2 Feeder Experiments .....	112
6.3.3 Variability Results .....	117

	Page
6.3.4 Spline Fitting .....	121
6.3.5 Fine Screw Experiments.....	122
6.3.6 Fed Material Experiments.....	124
6.4 Conclusions .....	125
CHAPTER 7. CONCLUSIONS AND FUTURE DIRECTIONS.....	127
7.1 Conclusions .....	127
7.2 Recommendations for further work .....	129
REFERENCES .....	132
APPENDICES	
APPENDIX A: IMAGE ANALYSIS CODE.....	141
APPENDIX B: ADDITIONAL GRANULE IMAGES.....	277
APPENDIX C: RESULTS TABLES .....	281
APPENDIX D: ELECTRONIC FILES .....	283
VITA .....	284



## LIST OF FIGURES

Figure	Page
Figure 1.1: Granule Formation Mechanism (A) Tunneling (B) Spreading (C) Crater <sup>5</sup> .....	4
Figure 2.1: Nucleation Regime Map <sup>12</sup> .....	11
Figure 2.2: Granule Nucleation Regime Map developed as a function of the Bond Number and Bed Porosity <sup>5</sup> .....	12
Figure 2.3: Proposed growth regime map where steady-growth-to-induction-growth boundaries are functions of $St_v$ <sup>18</sup> .....	15
Figure 2.4: Schematic describing the nucleation immersion mechanism <sup>11</sup> .....	19
Figure 2.5: X-ray tomographic sections of ten melt-pelletized granules after (a) 1 min (b) 2 min for immersion mechanism <sup>11</sup> .....	22
Figure 2.6: Proposed agglomeration method involving primary and secondary agglomeration in a suspension <sup>43</sup> .....	29
Figure 2.7: Size distribution of primary agglomerates formed in suspension by Pagnoux et al. <sup>43</sup> . Note narrower size distribution derived from Method 2.....	29
Figure 2.8: SEM micrograph of the inner structure of a sphere elaborated by colloidal granulation according to (a) method 1, (b) method 2 of powder deagglomeration <sup>43</sup> .....	30
Figure 2.9: Images of frozen spheres (a) before drying (b) after drying (c) after sintering <sup>43</sup> .....	30

Figure	Page
Figure 2.10: Illustration of four experimental approaches to X-ray microtomography data collection: a pencil, b fan, c parallel, and d cone beam methods. P is X-ray source, C is collimator, O is object being imaged, $X_2$ is axis about which sample is rotated to produce different views required for reconstruction, S is slit, and D is detector <sup>8</sup> .....	35
Figure 2.11: CT number profiles obtained across the center of a uniform water phantom without calibration correction (a) and with calibration correction (b) <sup>54</sup> .....	39
Figure 2.12: An image slice of a granule formed using sub-micron zinc oxide particles. The material is too dense to be properly imaged by the equipment.....	42
Figure 2.13: Quotient H and the different diameters for quotient H calculation. Dgran for homogeneous, dgran and dvoid for inhomogeneous. D1 is horizontal, D2 is diagonal, D3 is vertical <sup>42</sup> .....	45
Figure 3.1: Diagram of Static Bed Granule Production, provided by Heather Emady .....	56
Figure 3.2: Drum Granulation Experimental Setup .....	58
Figure 3.3: (A) Raw XRCT Slice, 0.5 $\mu\text{m}$ Alumina (B) Segmented Particles (black) (C) Convex Hull Output (white=particle) (D) Color Coded Connectivity Map (E) External Void Removed .....	62
Figure 3.4: Diagram of Feret's Diameter, including minimum and maximum measurements <sup>77</sup> .....	64
Figure 3.5: Demonstration of Feret's Diameter process for separating macro-voids from pore space. (A) Raw Void Space Image (B) Processed with objects below 30 $\mu\text{m}$ excluded .....	64
Figure 4.1: SEM Micrographs (A) Powder A (B) Powder B (C) Powder C (D) Powder D ...	67
Figure 4.2: Particle size distribution of well dispersed alumina particles (Malvern Mastersizer 2000) .....	68
Figure 4.3: Particle size distribution of 108 micron aggregates (sieve size analysis) .....	68
Figure 4.4: Ring Shear Tester results for alumina oxide powders .....	69
Figure 4.5: Microscope Images of alumina granules. (A) Powder A1 (B) Powder A2 (C) Powder A3.....	70

Figure	Page
Figure 4.6: Representative XRCT Slices. (A) Powder A1 (B) Powder A2 (C) Powder A3 (D) Power B1 (E) Powder C1 (F) Powder D1 [Note: Granule contrast has been increased to improve object visibility in print] .....	71
Figure 4.7: 3D macro-void reconstructions (A) Powder A1 (B) Powder A2 (C) Powder A3 .....	73
Figure 4.8: Fraction of total granule volume contained in macro-voids .....	75
Figure 4.9: Average size of largest individual macro-voids .....	76
Figure 4.10: Axial distribution for 0.5 $\mu\text{m}$ alumina granules. Describes distribution of macro-void volume as a fraction of total macro-void volume. ....	78
Figure 4.11: 0.5 $\mu\text{m}$ primary particles that have been passed through different sieve sizes. (A) A1 (B) A2 (C) A3.....	79
Figure 5.1: Microscope Images of Alumina Granules. (A): Powder A5 3 sec (B): Powder A5 5 min (C): Powder A5 15 min (D): Powder C5 3 sec (E): Powder C5 1 min (F): Powder C5 5 min .....	86
Figure 5.2: Representative diagrams of granule microstructures. (A): Powder A Nuclei (B): Powder A with powder shell (C): Powder C with large enclosed macro-void (D): Powder C with squished macro-void (E): Powder C with internal void connecting to granule exterior .....	87
Figure 5.3: XRCT Slices of Powder A & C Granules. (A): Powder A5 3 seconds (B): Powder A5 5 minutes (C): Powder C6 1 minute (D) Powder C6 1 minute (E) Powder C4 1 minute .....	89
Figure 5.4 Macro-void Volume Fraction of Powder A granules .....	90
Figure 5.5: Macro-Void Volume Fraction of Powder C Granules .....	91
Figure 5.6: Diameter of Largest Macro-Void in Powder A granules expressed as Equivalent Sphere Volume Diameter .....	92
Figure 5.7: Diameter of Largest Macro-Void in Powder C granules expressed as Equivalent Sphere Volume Diameter .....	92
Figure 5.8 Powder A 5.5 cP Radial Analysis of macro-void volume fraction .....	93

Figure	Page
Figure 5.9: Powder C 5.5 cP Radial Analysis of macro-void volume fraction .....	94
Figure 5.10: Powder A 35 cP Radial Analysis of macro-void volume fraction .....	95
Figure 5.11: Powder A 70 cP Radial Analysis of macro-void volume fraction .....	96
Figure 5.12: Powder C 35 cP Radial Analysis of macro-void volume fraction .....	96
Figure 5.13: Powder C 70 cP Radial Analysis of macro-void volume fraction .....	97
Figure 5.14: Powder A5, 15 minutes, broken shell showing hollow agglomerate structure .....	98
Figure 5.15: Labeled Connected Macro-Void Volumes for Powder A6 granules (A) 3 seconds (B) 5 minutes .....	98
Figure 6.1: Mass flow rates reported by control box using 1 second sampling times for 10% DC. Large amounts of data are lost for SSF and MicroAPAP w/ SiO <sub>2</sub> .....	108
Figure 6.2: Shear Cell results for raw materials, unconfined yield strength and maximum principal stress. Scaled so that x and y axes are proportionately sized .....	109
Figure 6.3: Shear cell results for previously fed APAP materials. Scaled so that x and y axes are proportionately sized .....	110
Figure 6.4: Basic flowability energy results for coarse screw materials, covering Raw and Fed materials.....	111
Figure 6.5: % Compressibility for raw and fed powders.....	112
Figure 6.6: Calculated Mass Flow Rate against Setpoint Mass Flow Rate with 1:1 line and +/- 10% lines.....	114
Figure 6.7: 25% DC Coarse Screw Mass Flow Rates calculated from net weight data...	115
Figure 6.8: 10% DC coarse screw mass flow rates calculated from net weight data .....	116
Figure 6.9: 5% DC coarse screw mass flow rates calculated from net weight data .....	116
Figure 6.10: 2% DC coarse screw mass flow rates calculated from net weight data .....	117
Figure 6.11: Mass Flow Rate RSDs sorted low-high for 25% DC, non-spline fitting .....	118
Figure 6.12: %RSD against different materials and screw speeds for coarse screw experiments. ....	119
Figure 6.13: %RSD as a function of the BFE at 25% DC for coarse screw experiments..	120

Figure	Page
Figure 6.14: %RSD as a function of unconfined yield stress (kPA) at 25% DC for coarse screw experiments.....	120
Figure 6.15: %RSD as a function of compressibility for 25% DC coarse screw experiments .....	121
Figure 6.16: %RSD against % DC for spline fit calculated RSDs .....	122
Figure 6.17: Spline Fitting RSD values for 25% DC.....	122
Figure 6.18: RSD results for fine screw experiments.....	123
Figure 6.19: RSD against BFE for fine screw experiments at 25% DC.....	124
Figure 6.20: % RSD for raw and fed material coarse screw experiments at 5% and 10% DC.....	125

## ABSTRACT

Davis, Nathan B. Ph.D., Purdue University, December 2015. Granulation Behavior of Ultra-Fine Powders: Examination of Granule Microstructure, Consolidation Behavior, and Powder Feeding. Major Professor: James Litster

Ultra-fine powders, sized between 0.1-10  $\mu\text{m}$ , are commonly used materials in a variety of industries including detergents, catalysts, paint pigments, and agricultural products. Ultra-fine powders are known to have complex behaviors due to cohesive forces and are considered difficult to handle, feed, and form consistent final products. One specific issue is that wet granules formed from ultra-fine powders are difficult to densify and take significant amounts of time to reach a desired granule density. However, a thorough review of the wet granulation literature shows that ultra-fine powders have received little direct study, with studies focused upon materials that are greater than 20  $\mu\text{m}$  in average particle size where cohesive forces have a lesser impact. This thesis hypothesizes that formation of complex structures within ultra-fine powder granules is responsible for densification issues and that the granule microstructure can be controlled through careful powder handling and preparation prior to the granulation. Furthermore, this thesis hypothesizes that ultra-fine powders have unique behaviors separate from larger primary particles which requires specific studies in the applicable size range to understand ultra-fine powder granulation.

In order to investigate this hypothesis, methods are developed to describe the granule microstructure and specifically the granule void phase distribution in 3D space. X-ray computed tomography (XRCT) and image analysis techniques are used to isolate, identify and describe the spatial distribution of the various granule phases (particle, binder, internal air). Additional methods for distinguishing between large macro-voids and the pore space between primary particles are created as well as tools for quantifying macro-void size, shape, volume fraction in the granule ( $\epsilon_{\text{void}}$ ) and distribution of the voids within the granule structure. Descriptions of macro-void size, shape, and volume fraction show that the macro-void properties depend upon primary particle size, powder history, liquid binder and method of granule formation.

The granule microstructure measurement methods are developed to describe the internal structure of single-droplet nuclei granules formed in a static powder bed. Alumina powders with mean size varying from 0.5 to 100  $\mu\text{m}$  are used as model powders with water and polymer solutions as the liquid binders. The size, shape, and macro-void volume fraction ( $\epsilon_{\text{void}}$ ) of the macro-voids is used to describe the effects of primary particle size and powder bed preparation on granule microstructure. Granules formed from ultra-fine powders show the presence of large spheroidal macro-voids distributed throughout a particle matrix (primary particles and pores). Granules formed from coarser powders (larger than 10  $\mu\text{m}$ ) show either no macro-voids or non-spheroidal macro-voids which are described as “cracks” within the granule microstructure. Smaller primary particles within the ultra-fine powder range are found to increase the size and  $\epsilon_{\text{void}}$  of

measured macro-voids and the complexity of the structure. The maximum void size and  $\epsilon_{\text{void}}$  are dependent upon the powder bed preparation technique. Sifting the material through a 1.4 mm sieve produces a larger maximum macro-voids size and larger  $\epsilon_{\text{void}}$  than producing powder beds by sifting through either 710  $\mu\text{m}$  or 500  $\mu\text{m}$  sieves. Sifting of 0.5  $\mu\text{m}$  primary particles results in formation of stable, large, spheroidal agglomerates while other tested materials do not form stable structures from sifting.

The developed methods are also applied to single-droplet granules formed in a tumbling drum to investigate granule microstructures from a moving bed. The effects of consolidation time and liquid binder viscosity are also evaluated and the results are compared to predictions from the surface-tension flow model of the nucleation immersion mechanism developed by Hounslow et al.. Two of the model powders (mean size 0.5  $\mu\text{m}$  and 25  $\mu\text{m}$ ) and three polymer solution binders with viscosities ranging from 5.5 to 70  $\text{mPa}\cdot\text{s}$  are used and their structure measured as a function of tumbling time up to 15 minutes. All granules are found to be hollow with a large central macro-void. The central void persists regardless of run time or liquid binder viscosity. The 0.5  $\mu\text{m}$  powder granules form a powder shell around the hollow structure supported by a network of powder agglomerates. As consolidation time increases, the thickness of the powder shell slowly increases, but the internal void structure is unchanged. Increasing the liquid binder viscosity increases the void size and  $\epsilon_{\text{void}}$ . Granules formed from the 25  $\mu\text{m}$  powder have a simpler structure. They have a uniform packing structure of the primary particles surrounding a central void. This structure forms within the first few seconds and is then



unaffected by either time on liquid viscosity. Apart from the persistence of the central void, the kinetics of nucleation for the coarse powder agrees with Hounslow's model. However, the ultrafine powder granules have a complex multiscale structure that is not predicted from a simple nucleation model.

Finally, the feeding behavior of several ultra-fine and coarser powders used in the pharmaceutical industry is evaluated using the relative standard deviation (RSD) of the mass flow rate for a twin-screw feeder run in volumetric mode to predict quality of feeding. The RSD results are compared to bulk powder flow properties (unconfined yield stress, compressibility, basic flowability energy) measured with the FT4 Powder Rheometer. The RSD of the mass flow rate does not correlate with any of the measured properties. The quality of material feeding and likelihood of failure remains a complicated endeavor with multiple types of failure that are not well described by a single material property.

## CHAPTER 1. INTRODUCTION

### 1.1 Background

Wet granulation is a size enlargement technique used in particle engineering to create semi-permanent agglomerates by bringing mixtures of powder and liquid together. These agglomerates, hereafter referred to as granules, will then typically have excess liquid removed during a drying step prior to further processing. The drying process creates solid bonds between primary particles from materials dissolved in the liquid phase. These solid bonds provide strength and stability to the granule which allows it to act as a single, larger particle during further processing. They also impart a specific microstructure to the granule.

Wet granulation is used to improve the bulk properties and handling of fine powders, especially micron-sized materials. Micron-sized materials offer many advantages due to the increased surface area per unit volume that is available for dissolution, reaction, etc. These attributes are highly desirable, for example, in pharmaceutical and catalysis applications. However, micron-sized particles are typically difficult to use as primary particles during processing and handling as they have low bulk density, low flowability, and a high propensity to create dust. These issues are the result of cohesive forces, such as Van der Waals forces, capillary forces and electrostatic forces, which increase in

strength and become increasingly significant as the primary particle size decreases. This is especially true for materials below 10  $\mu\text{m}$  where the cohesive forces begin to dominate gravity forces and allow for the creation of increasingly complex structures<sup>1</sup>. The goal of wet granulation is to improve the bulk properties without losing the advantages that fine powders have for product performance. The granules produced by wet granulation can be handled, transported, and processed with minimal cohesive issues because the agglomerated particles behave as a single large object. The desirable properties of the small particles can be retrieved by break up and dissolution of the granule in liquids. These capabilities have made wet granulation a widely used process in industries which use fine powders such as the fertilizer, pharmaceutical, chemical, food, and catalysts<sup>2</sup>.

Typical fundamental wet granulation studies in literature are performed with a combination of model materials, such as glass ballotini, and large particles, typically greater than 20  $\mu\text{m}$ . These materials are generally well-behaved and comparatively easy to work with when compared to ultra-fine powder, defined here as powders with average particle sizes between 0.1-10  $\mu\text{m}$ . There are few published papers on wet granulation of ultra-fine powders even though they are used as pharmaceutical API's, paint pigments, catalyst supports, and food additives. Ultra-fine powders are expected to have more complex microstructures in granulation and discussion with industry indicate that handling is more complex, processing times are longer and granule properties are subject to large measurement variations<sup>3</sup>. The issues of long processing

times and variation in granule property measurements are of concern to industry. The subject of complex behaviors, such as the effects of powder handling or powder history is largely anecdotal and unexplored. A study of the effects of humidity on powder properties at 5 relative humidity (RH) levels between 17% and 94% RH has shown that different values are reported at ranges typical of industrial material storage, which are 20% to 40% RH<sup>4</sup>.

There are published studies which indicate that ultra-fine powders form granules differently than granules formed from large powders<sup>5,6</sup>. Granules formed from high Bond number processes were shown to form via the Tunneling mechanism in static powder beds while low Bond number processes formed granules through the Spreading/Crater mechanism. This resulted in significantly different granules shapes as shown in Figure 1.1, where Tunneling granules are rounded objects and Spreading/Crater granules have a flatter surface at the top of granule.

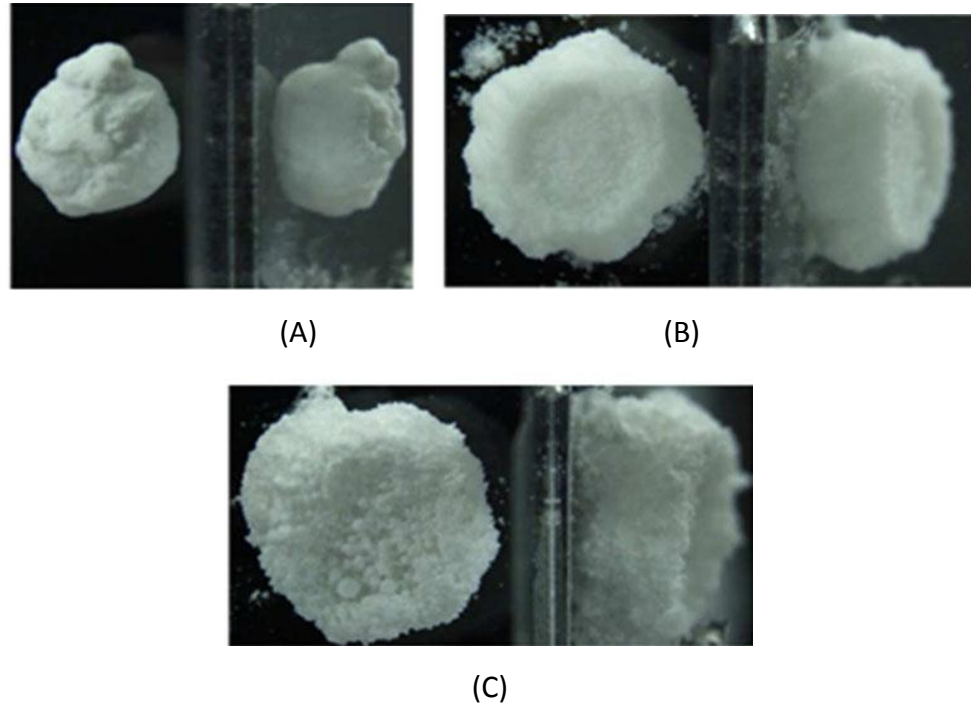


Figure 1.1: Granule Formation Mechanism (A) Tunneling (B) Spreading (C) Crater<sup>5</sup>

The majority of research into wet granulation and specifically densification has been done on powder/liquid combinations that likely fall into the Spreading/Crater regime on the basis of primary particle size. The difference in shape and formation mechanism for the Tunneling regime, where ultra-fine powders are expected to exist, shows that granules formed from ultra-fine powders may also have different densification behaviors than granules formed from larger materials.

The most common way to describe granule densification is to make it a function of granule porosity. The granule porosity is simply defined as the proportion of granule volume not taken up by the solid phase. More complicated analyses, such as those used in simulations, may choose to consider a 3 phase system of solids, liquids, and gases. The porosity is always considered as a global average granule property where the void

volume is assumed to be evenly distributed throughout the granule. Porosity is a commonly used measurement because it is easy to obtain and is usually a suitable descriptor for granulation processes. However, this global approach to describing granule densification does not account for any potential differences that are a result of differences in granule microstructure. The global approach treats a hollow core-shell granule arrangement the same as a distributed pore network in terms of granule densification and cannot tell the difference between these types of microstructures. Microstructure differences can cause differences in granule strength and yield stress measurements, but these measurements do not actually tell what the microstructure is. An examination of the microstructures of granules formed from ultra-fine powders should be capable of identifying why such granules take a long time to densify.

The best way to evaluate and visualize granule microstructure is using X-Ray Computed Tomography (XRCT) because it is non-destructive provides a 3-D representation of the interior of the granule. This 3-D view is superior to 2-D stereography methods such as cutting a granule in half and using SEM to get a snapshot view of part of a granule. XRCT uses differences in material density and x-ray adsorption coupled with high-powered computers to produce images that distinguish between different phases within the granule. The image resolution is determined by a combination of available x-ray energy, computing power, and scan volume. Recent advances in the areas of available x-ray energy and computing power have led to the creation of  $\mu$ CT and nanoCT units capable of image resolutions at 1  $\mu$ m for whole granules or even tens of nanometers for smaller

scan volumes<sup>7,8</sup>. However, most existing analyses of granule microstructure using XRCT are created to describe a very specific observed structure and cannot be applied to other types of granules or other types of structures. A recent, significant advance in performing image analysis on XRCT data was created by Dale et. al. who studied granule microstructures to evaluate the effects upon breakage<sup>9,10</sup>. He created an analysis technique which evaluated the size, connections and positional distribution of the various granule phases, the particles, binder, and air phases. The ability to quantify, locate, and label the different phases within the granule provides the opportunity to properly evaluate and compare a variety of granule structures, especially those of granules formed from ultra-fine powders, as a function of production method and time.

## 1.2 Thesis Objectives

The primary goal of this thesis is to quantitatively evaluate the granulation behavior of ultra-fine powders. To accomplish this, it will be necessary to develop novel, widely applicable methods for describing different granule structures and enable comparisons between granules. The thesis will use the developed methods to quantify the differences in granule structures as a function of time, particle size, and processing conditions. This thesis will also evaluate the effects of processing/handling on material properties for cohesive powders. Specifically the objectives of this thesis are to:

1. Develop a widely applicable methods for quantifying differences in microstructure and void distribution for granules formed from ultra-fine powders;

2. Use the developed methods to evaluate the effects of primary particle size and preparation method on granule microstructures;

3. Use the developed methods to evaluate the accuracy of the surface tension driven nucleation immersion model as applied to granules formed in a tumbling drum; and

4. Use relative standard deviation (RSD) to evaluate the relationship between and the effects of twin-screw feeding and the flow properties of cohesive materials.

### 1.3 Thesis Outline

To complete the objectives listed above, Chapter 2 give a comprehensive literature review on the current understanding of the wet granulation behavior of ultra-fine powders. The study of ultra-fine powders in wet granulation is sparse so this section also includes overviews of the densification literature, granule nucleation, XRCT studies, and powder feeding.

In Chapter 3 the materials and methods used for the static bed and drum granulation studies of alumina are described, including references tables labeling the material/process combinations that are used in Chapters 4 and 5. This section also includes the description of new methods of analyzing XRCT data in addition to those created by Steven Dale<sup>9,10</sup>. This includes a description of the surface area measurement technique and the selection process for macro-voids and micro-voids performed using a Feret's diameter measurement.



Chapter 4 uses the methods discussed in Chapter 3 for a static bed, single-droplet nucleation study of the microstructure of alumina granules. The granules are formed by passing one of four different size fractions of chemically similar  $\alpha$ -alumina ( $d_{50}$ = 0.5  $\mu\text{m}$  to 108  $\mu\text{m}$ ) through one of three sieves (1.4 mm to 400  $\mu\text{m}$ ) to form a static bed. The microstructures are compared on the basis of maximum macro-void size and total volume fraction of macro-voids.

Chapter 5 uses the methods discussed in Chapter 3 for a time based, single droplet nucleation study in a tumbling drum. The granules are formed from one of two  $\alpha$ -alumina powders ( $d_{50}$  = 0.5  $\mu\text{m}$  or 25  $\mu\text{m}$ ) that have been passed through 1.4 mm sieve and granules are collected at 4 different time points. The microstructure results as a function of time are compared to predictions of granule behavior from the surface tension driven model from Hounslow's nucleation immersion mechanism<sup>11</sup>.

Chapter 6 is a powder feeding study using relative standard deviation (RSD) to evaluate the relationship between feeding behavior of cohesive powders and powder properties measured with the FT4 Powder Rheometer. A description of the process and analysis methods used is included.

Chapter 7 describes the main contributions of the current work and suggests extensions of the work that could be pursued. Appendices are included to aid in continuation the work and to provide full access to the data that was described by averages in the chapters. Additional images of experimental granules are included as are 3-D reconstructions and other images of interest.

## CHAPTER 2. LITERATURE REVIEW

### 2.1 Overview

The existing wet granulation literature has seen significant advances in models for predicting granule nucleation and consolidation behavior. Section 2.2 will discuss these models and issues pertinent to ultra-fine powders. Section 2.3 will evaluate the existing wet granulation covering ultra-fine powders. Section 2.4 & 2.5 will discuss relevant literature from suspension based agglomeration and fluidization of ultra-fine powders which can be applied to wet granulation. Section 2.6 & 2.7 will cover the XRCT process and application in literature. Section 2.8 covers literature on loss-in-weight feeding. Section 2.9 contains a critical literature summary.

### 2.2 Dimensionless Groups and Regime Maps

#### 2.2.1 Granule Nucleation Theory

A review of the available granulation literature has identified five key dimensionless groups that are related to controlling the formation and growth of granules. These are the drop penetration time ( $\tau_p$ ), dimensionless spray flux ( $\psi_a$ ), Bond number ( $Bo_g^*$ ), Stokes deformation number ( $St_{def}$ ), and the maximum granule pore saturation ( $s_{max}$ ).

The drop penetration time is the time required for a single droplet of binder liquid to fully penetrate the powder surface and is used to describe nucleation behavior. This

equation is derived under conditions where the wetting is promoted by capillary pressure and resisted by viscous dissipation<sup>3,12,13</sup>:

$$t_p = 1.35 \frac{V_d^{2/3}}{\varepsilon_{eff}^2 R_{eff}} \frac{\mu}{\gamma_{lv} \cos \theta} \quad (2.1)$$

where  $V_d$  is the drop volume,  $\mu$  is the liquid viscosity,  $\varepsilon_{eff}$  is the powder bed porosity,  $R_{eff}$  is the pore radius,  $\gamma_{lv}$  is the liquid surface tension and  $\theta$  is the solid-liquid contact angle. The drop penetration time (2.1) can be made dimensionless by dividing by the circulation time ( $t_c$ ), defined as the time required for a packet of powder to return to the spray zone.

$$\tau_p = \frac{t_p}{t_c} \quad (2.2)$$

The effective pore size between particles is defined as:

$$R_{eff} = \frac{\varphi d_{32}}{3} \frac{\varepsilon_{eff}}{(1-\varepsilon_{eff})} \quad (2.3)$$

where  $\varphi$  is the particle shape factor and  $d_{32}$  is the surface mean particle diameter. They do note that their model works well for most materials except for ultra-fine ZnO and TiO<sub>2</sub> which are expected to have a more complex microstructure do to the highly cohesive nature of the material. Ultra-fine powders are materials that can self-aggregate and may have more than one effective pore size, the one between primary particles in an aggregate and the pore size between aggregates. They do not attempt to address the case of primary particles agglomerated either strongly or loosely in the bed to determine which set of pore space will govern the behavior.

The second group used to describe nucleation is  $\Psi_a$ , which is a measure of the density of drops falling on the powder surface<sup>14</sup>:

$$\Psi_a = \frac{3V}{2Ad_d} \quad (2.4)$$

where  $V$  is the volumetric spray rate,  $A$  is the area flux of powder through the spray zone, and  $d_d$  is the droplet diameter.

$\tau$  and  $\Psi$  have been used to define a regime map, proposed by Hapgood et. al., which can be used to predict general behaviors and summarizes potential nucleation regimes<sup>15</sup>. In order to operate in the drop controlled nucleation region, where one drop forms one granule, both dimensionless groups must be less than 0.1 (Figure 2.1).

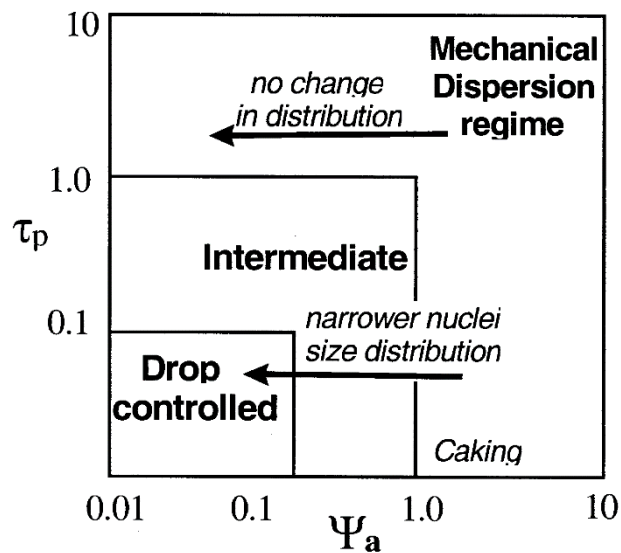


Figure 2.1: Nucleation Regime Map<sup>12</sup>

A third dimensionless group which is used to predict the nucleation mechanism is the Bond number, which is a measure of the ratio of capillary to gravity forces acting upon a particle<sup>6,16</sup>:

$$Bo_g^* = \frac{\gamma \cos\theta}{d_{32}^2 \rho_p g} \tag{2.5}$$

Where  $\gamma$  is the liquid surface tension,  $\theta$  is the solid-liquid contact angle,  $d_{32}$  is the surface mean particle diameter,  $\rho_p$  is the particle density and  $g$  represents the force of gravity.

The Bond number predicts whether a granule will be formed through the Tunneling mechanism or the Spreading/Crater mechanism in drop controlled nucleation. Emady et al. have created a regime map in Figure 2.2 which is defined by the Bond number and the bed porosity and also matched their experimental results to their model criterion<sup>5</sup>.

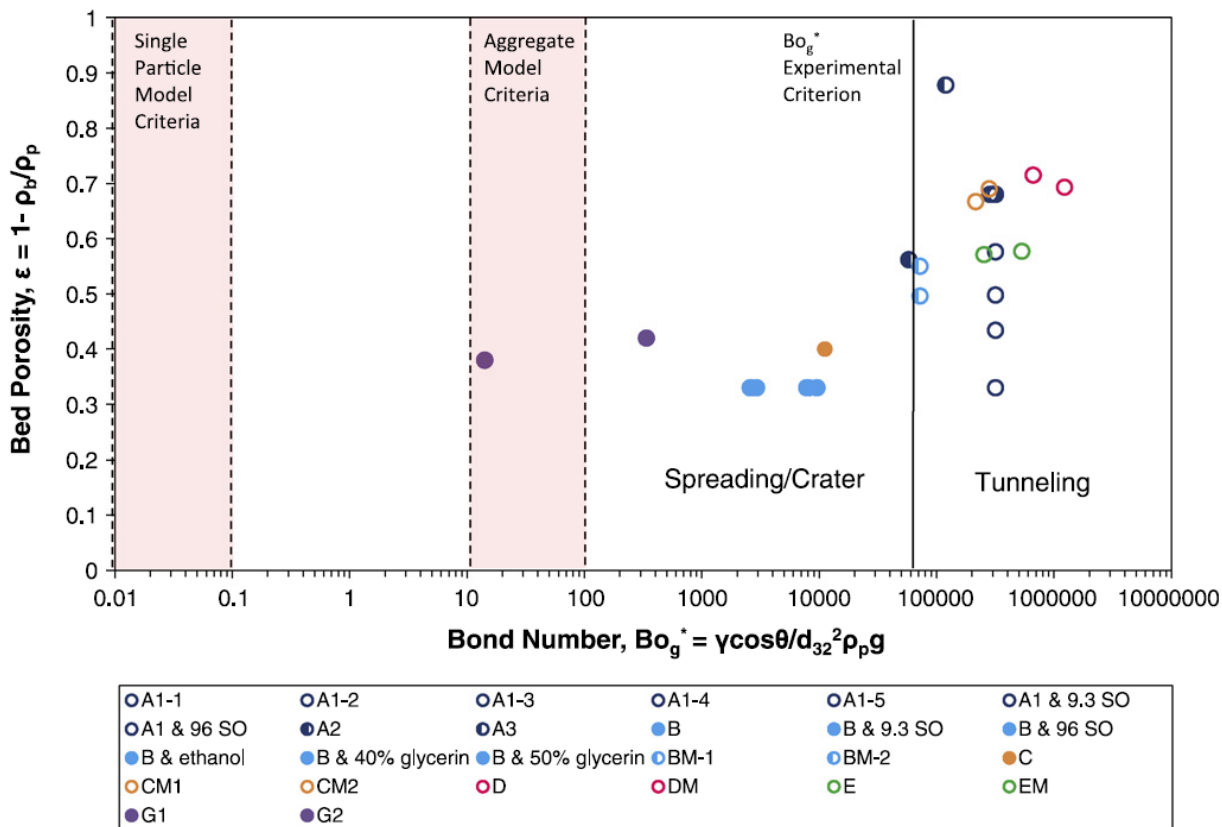


Figure 2.2: Granule Nucleation Regime Map developed as a function of the Bond Number and Bed Porosity<sup>5</sup>

The Tunneling mechanism occurs when the surface tension forces and capillary pressure forces exceed the weight of the aggregate and are expected to occur at high values of  $Bo_g^*$ . This typically corresponds with particles in the range of sizes which includes the ultra-fine region. The Tunneling mechanism has also been found to be insensitive to the liquid properties and drop release height<sup>6</sup>. Emady developed two pore balance models to predict the regime transition, one based on primary particles (2.4) and one based on powder aggregates (2.5):

$$Tunneling \text{ if } \frac{F_{s,agg} + F_{c,agg}}{F_{g,agg}} > 1 \quad (2.4)$$

$$\left( \frac{F_s + F_c}{F_g} \right) \left( \frac{r_{32}}{R_{agg}} \right) > 1 \quad (2.5)$$

Where  $F_s$  represents surface tension forces for a single particle,  $F_c$  is the capillary pressure forces acting on a single particle,  $F_g$  is the force of gravity acting on a single particle,  $r_{32}$  is the Sauter mean diameter, and  $R_{agg}$  is the aggregate size. Emady et. al. note that their models under predict the ratio of forces by at least an order of magnitude relative to the experimentally determined values. Emady assumed an average ratio of particle size to aggregate size, assumed a characteristic pore size for the system and neglected particle-particle adhesion forces. Improvements in the understanding of the effects of agglomerate size on the granulation behavior could be used to improve this model. Their work does not attempt to characterize the microstructure of the granules they formed.

Particle size, liquid viscosity, surface tension and contact angle are some of the key properties that have been identified to govern nucleation. There are additional papers which discuss either granule nucleation behavior, but none are as directly important to this study as those previously discussed. Charles-Williams et. al. looked at the effects of changes in liquid properties upon the competing spreading/infiltration processes through single-droplet nucleation experiments. Changes in viscosity were found to have a greater impact upon the infiltration rate than upon the spreading rate, such that higher viscosity liquids will spread out more than lower viscosity liquids on the same material<sup>17</sup>.

### 2.2.2 Coalescence and Consolidation

$St_{def}$  and  $s_{max}$  are the key dimensionless groups controlling granule coalescence and consolidation. Iveson et. al. used  $St_{def}$  and  $s_{max}$  to construct a regime map to predict types of granule growth based on powder and liquid binder properties<sup>18</sup>. The various regions (Figure 2.3) have been defined as induction growth, nucleation only, crumb, slurry, and steady growth (see Table 2.1).

Table 2.1: Qualitative description of granule growth regimes as described by Iveson et al.<sup>18</sup>

Regime	Qualitative Description of the Regime Behavior
Induction	Consolidation until sufficient liquid reaches granule surface, then growth occurs
Nucleation Only	No growth occurs after binder addition, insufficient binder in system
Crumb	Granules are too weak to form permanently. Continual formation and breakage
Slurry	Excess binder causes the formulation to be describable as slush
Steady Growth	Granules grow steadily as a function of time

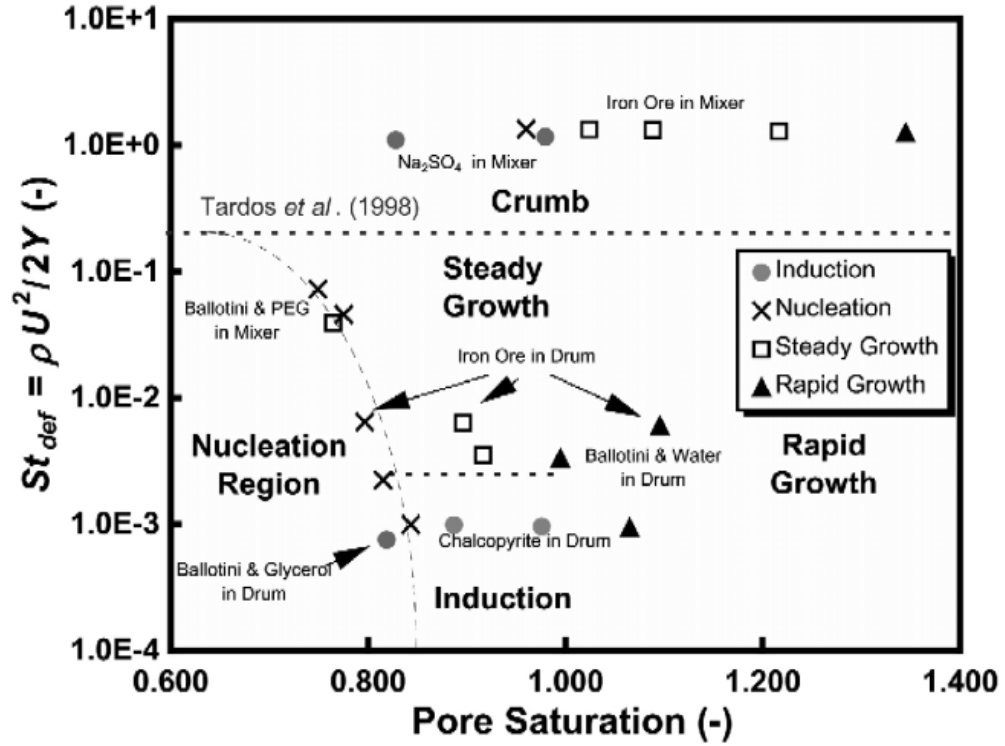


Figure 2.3: Proposed growth regime map where steady-growth-to-induction-growth

boundaries are functions of  $St_v$ <sup>18</sup>

The  $St_{def}$  is used to quantify the granule deformation during impact<sup>16</sup>:

$$St_{def} = \frac{\rho_g U_c^2}{2Y_g} \quad (2.6)$$

where  $\rho_g$  is granule density,  $U_c$  is the collision velocity in the granulator, and  $Y_g$  is the dynamic yield stress of the granules.  $s_{max}$  is the pore saturation at minimum granule porosity<sup>18</sup>:

$$s_{max} = \frac{w\rho_s(1-\varepsilon_{min})}{\rho_l\varepsilon_{min}} \quad (2.7)$$



where  $w$  is the liquid to solid mass ratio,  $\rho_s$  is the true particle density,  $\rho_l$  is the liquid binder density, and  $\varepsilon_{min}$  is the minimum granule porosity. These are used to define the different growth regimes (Figure 2.3) where the transition regions are approximated because granule growth is also dependent upon other variables that are independent of  $St_{def}$  and  $s_{max}$ <sup>18</sup>. In order to determine specific information about the rate of granule growth, which this regime map does not provide, it is necessary to turn to various empirical models which attempt to explain the consolidation and coalescence phenomenon.

### 2.2.3 Consolidation and Coalescence models

Consolidation models examine either granule porosity or the inter-particle gap distance. An experimental study performed by Iveson and co-workers in a tumbling drum examined the changing porosity of various size fractions of glass ballotini ( $D_{3,2}$  8-39  $\mu\text{m}$ )<sup>19,20</sup>. The change in porosity with time was well described by a first order exponential decay process that, when integrated, is given by:

$$\frac{\varepsilon - \varepsilon_{min}}{\varepsilon_0 - \varepsilon_{min}} = \exp(-kt) \quad (2.8)$$

where  $\varepsilon_0$  is the average initial porosity of the batch,  $\varepsilon_{min}$  is the minimum porosity achieved, and  $k$  is the consolidation rate constant<sup>19</sup>. It is postulated that  $k$  is proportional to frequency and energy of the granule collisions and inversely proportional to the dynamic yield strength of the formulation.

There are also two theoretical models in the literature. Ouchiyama and Tanaka describe the granule consolidation rate by<sup>21</sup>:

$$\frac{d\varepsilon}{d\tau} \cong - \left\{ 1 - \frac{(1-\varepsilon)^3}{\varepsilon K_\varepsilon} \right\}^n \quad (2.9)$$

where  $\varepsilon$  is granule porosity at a time  $t$ ,  $K_\varepsilon$  is dimensionless granule compaction rate,  $\tau$  is dimensionless compaction time, and  $n$  is a parameter describing the distribution of granule impact energies. The model proposed by Ennis et al. includes the effects of binder viscosity, which is not considered by Ouchiyama and Tanaka, and proposes that the rate of consolidation will increase with increasing viscous Stokes number ( $St_v$ ) by<sup>22</sup>:

$$\frac{\Delta x}{h} = 1 - \exp(-St_v) \quad (2.10)$$

where  $\Delta X$  is the reduction in inter-particle gap distance  $h$  per collision. Both of these models predict that factors which increase granule strength should decrease the consolidation rate, increases in kinetic energy will increase consolidation rate, and that the consolidation will slow as granules densify. These predictions are consistent with the experimental conclusions of Iveson and co-workers<sup>19,20</sup>.

Coalescence models fall into two different classes as defined by Liu et al., where class I models consider only if the particles stick or rebound during collision and class II models consider whether the particles bond sufficiently during contact to avoid breakage<sup>23</sup>. The two types of granules that can be described are the non-deformable granules (associated with induction growth) and the deformable granules (associated with steady

growth). The work of Sochon et al. (2005) on the granulation of zinc oxide indicates showed induction style growth <sup>24</sup>.

The Ennis model assumes the existence of elastic spheres with asperities that are coated with a uniform layer of a viscous liquid binder. Granules stick and coalesce if the initial kinetic energy is fully dissipated, otherwise granules will rebound. The model predicts that coalescence will always occur when  $St_v$  is less than some critical viscous Stokes number ( $St_v^*$ )<sup>13,22</sup>:

$$St_v = \frac{4\rho u_0 D}{9\mu} \quad (2.11)$$

$$St_v^* = \left(1 + \frac{1}{e}\right) \ln\left(\frac{h}{h_a}\right) \quad (2.12)$$

where  $\rho$  is granule density,  $u_0$  is half the initial relative velocity of impact,  $D$  is the granule diameter,  $\mu$  is liquid viscosity,  $e$  is the coefficient of restitution,  $h$  the thickness of the liquid surface layer and  $h_a$  the characteristic height of surface asperities. They define three regions of growth which are summarized in Table 2.2:

Table 2.2: Conditions for existence in different regimes from Ennis Coalescence Model<sup>22</sup>

Non-Inertial Regime	$St_v \ll St_v^*$	All collisions are successful regardless of size
Inertial Regime	$St_v \approx St_v^*$	Likelihood of coalescence becomes dependent upon size
Coating Regime	$St_v \gg St_v^*$	All collisions unsuccessful. Growth occurs by coating of powder onto the surface of already existing granules.

In their review of granulation rate processes, Hapgood et al. note that this model assumes elastic granules, a distinct surface layer of liquid and that viscous forces are dominant, so that the model is only applicable to initial nucleation or an induction-growth system<sup>13</sup>.

A different nucleation/growth mechanism has been described and modeled for high-shear mixers, termed immersion and dispersion<sup>11,25</sup>. Schaefer and Mathiesen first described the immersion and dispersion methods for granule nucleation in a high-shear mixer undergoing melt pelletization<sup>25</sup>. Dispersion occurs if the liquid droplets are comparable in size to the primary particles, in which case they will spread across the surface based upon each collisions. The immersion mechanism, depicted in Figure 2.4, occurs when liquid is present as droplets much larger than the primary particles and the primary particles will be spread across the surface of the liquid and will be driven into the liquid droplet through either surface tension forces or deformation driven diffusive flow<sup>11</sup>.

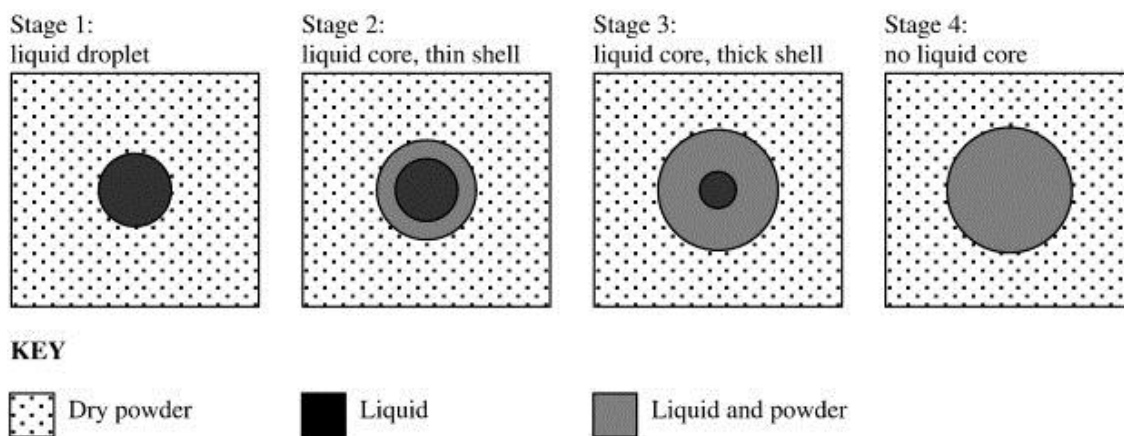


Figure 2.4: Schematic describing the nucleation immersion mechanism<sup>11</sup>

Hounslow et al have created kinetic models to predict the immersion mechanism behavior and show XRCT images in Figure 2.5 of granules collected after 1 minute and 2 minute that were formed through the immersion mechanism<sup>11</sup>. These materials show an inner core that consists of binder and an outer core of powder at 1 minute while the 2 minute granules show little or no evidence of a binder core. The immersion mechanism is one that has strong potential to occur given the primary particle size used in this study. The primary focus of the Hounslow work was developing models for predicting granule growth behaviors to use in multi-scale models. The integrated form (approximate explicit) of the surface tension driven flow model is <sup>11</sup>:

$$v = v_L \left( 1 + \frac{1 - \phi_{cp}}{\phi_{cp}} \sqrt{\frac{\vartheta}{t_{max}}} \right) \quad (2.13)$$

Where  $v$  is nucleus volume,  $v_L$  is liquid volume,  $\phi_{cp}$  is the critical-packing liquid volume fraction and  $\vartheta$  is time. They note that this model is based on the hypothesis that particles are drawn into the liquid drop and surface tension forces replace liquid at the center with particles. The displaced liquid is then capable of drawing in a new layer of particles. The rate of this layering process is highly dependent on the critical-packing factor which is typical of flow through porous media and the length of time for which layering occurs is inversely related to the primary particle size.

Model 1 (2.13) can be rewritten to predict the maximum granule growth time after which the granule should have a solid core. This model, written for a spherical geometry, could be applied to granulation of ultra-fine powders in a tumbling drum:

$$t_{max} = \frac{18.75\mu h_0^2}{\gamma d} \frac{1-\phi_{cp}^{1/3}}{\phi_{cp}^3} \quad (2.14)$$

Where  $t_{max}$  is the maximum growth time,  $\mu$  is liquid viscosity,  $\gamma$  is liquid surface tension,  $h_0$  is the initial droplet size,  $d$  is the primary particle diameter, and  $\phi_{cp}$  is the critical liquid packing fraction<sup>11</sup>. This provides an opportunity to test their model predictions against real world experiments and use XRCT to measure any changes in microstructure.

Pitt et. al. recently tested the nucleation immersion model in a static bed situation to evaluate the accuracy of the underlying assumptions<sup>26</sup>. The authors concluded that while the model accurately predicts the trends as a function of material properties, the experimental values are one to two orders of magnitude higher than the model predictions. This is attributed to secondary migration of the liquid after the drop had fully penetrated the static powder bed and is not accounted for in the original model. This experimental result for relevant time scales to model predictions must be accounted for in experimental testing in experimental granulation equipment.

In their second model, for diffusive driven flow, Hounslow et al. hypothesize that liquid is drawn to the surface through repeated small deformations. They have created an approximate expression for the spherical geometry as follows<sup>11</sup>:

$$v \cong \frac{v_L}{\phi_{cp}} \left( 1 - (1 - \phi_{cp}) e^{-\frac{12D_{eff}\phi_{cp}^{2/3}}{h_0^2}t} \right) \quad (2.15)$$

where  $D_{eff}$  is the effective diffusivity and  $h_0$  is the initial droplet size. For this model, the rate of grow follows an exponential decay model and the time constant is scaled by the

final size of the nucleus rather than initial droplet size. Hounslow et al. note that Model 1 (2.13) is based on well understood physics while Model 2 (2.15) is phenomenological and there is a need for experimental validation of their work. The immersion mechanism itself is of interest due to the difference in sizes between liquid drops and primary particles. The models may not be an accurate representation for ultra-fine powder behavior because the models are created to describe melt pelletization.

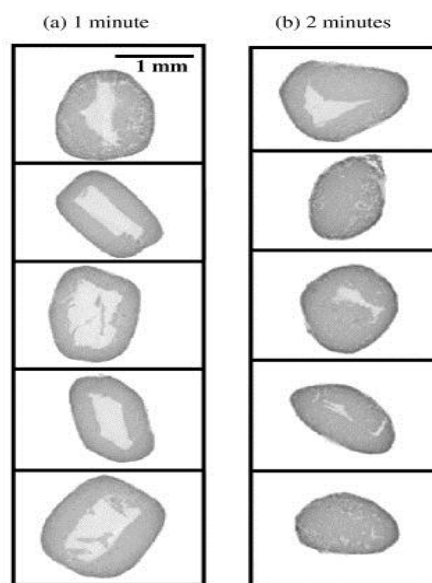


Figure 2.5: X-ray tomographic sections of ten melt-pelletized granules after (a) 1 min (b) 2 min for immersion mechanism<sup>11</sup>.

All of the previously discussed models make use of global granule properties. This can work well for predictions of homogeneous systems where the properties are well understood. In such cases, the internal structure of the granule should be well accounted for by the global property. This approach is less applicable to more complicated, non-homogeneous systems that can arise with the use of ultra-fine

powders. These materials can create granules with a complicated internal structure that is not well described by a global property, such as porosity.

### 2.3 Granulation of Ultra-Fine Powders

The overall study of granulation has primarily focused upon the use of model materials, such as glass ballotini and lactose. These are powders which are relatively easy to work with, that flow well, are often easy to obtain in a variety of size ranges and have a relatively large mean size ( $>20\ \mu\text{m}$ ). The study of ultra-fine particles ( $0.1\text{-}10\ \mu\text{m}$ ) behavior in wet granulation is uncommon, despite the fact that many powders of industrial interest for detergents, pigments, agricultural chemicals and ceramics are in this size range. These powders can be difficult to work with due to powder flow issues, availability of material, and a tendency to self-agglomerate into weak networks with a larger particle size<sup>27-29</sup>. As the primary particle size is reduced below  $10\ \mu\text{m}$ , van der Waals forces increase to a point at which they become non-negligible and capable of countering the weight of the individual particles<sup>1</sup>. The exact magnitude of this force also depends on particle shape and roughness, surface properties and the spread of the distribution. Van der Waals forces can lead to self-agglomeration of the dry primary particles, resulting in complex and potentially unwanted behaviors and making the powder behavior very sensitive to its prior history.

In their study of drop penetration time, Hapgood et. al found that their model was effective in all cases except when used for ultrafine powders (zinc oxide and titanium dioxide)<sup>3</sup>. Their penetration model assumes an effective porosity which is not as



accurate for ultrafine powder beds which are expected to have a more complex microstructure than model materials. It does note that the pore size is important to the drop penetration time, but does not address whether the important pore size is the one between primary agglomerates or between the primary particles. The study of hydrophobic nucleation mechanisms by Eshtiaghi et al. showed the formation of hollow and collapsed granules while using X-ray computed tomography (XRCT) to confirm their structural observations<sup>27-29</sup>. However, their work used a variety of materials, ranging in size from 0.01-320  $\mu\text{m}$  and these structures are not unique to the ultrafine powders, but are instead assigned to the hydrophobic nature of the materials. The work by Emady et al., which identified the existence of Tunneling and Spreading/Crater and the Bond number as a predictive measure is of great interest and has been previously discussed (see Chapter 2.2)<sup>5,6,16</sup>.

The work of Van den Dries et al. qualified three different methods of granule nucleation that were observed in a high-shear mixer<sup>30</sup>. Rough et al. studied an unusual scenario using a semi-solid paste as binder in a high-shear mixer for detergent granulation using 5  $\mu\text{m}$  zeolite powder, which means their observations are likely unique to their system<sup>31-33</sup>. Their work looks at the agglomeration mechanisms, bulk density characterization, and effects of solid formulations. Afarani et al. showed, for alumina in a high-shear environment, that increasing binder content led to a wide size distribution, enhanced attrition and bulk compression strength of sintered granules, but provide only some SEM images for structural examination<sup>34</sup>. A study of alumina granule growth in a

high-shear mixer by Saleh et al. notes three different growth regimes based upon granule diameters and changes in liquid to solid ratio<sup>35</sup>. Both studies note that they “homogenized” their bulk material at process conditions prior to experimentation, but do not note why such a step is necessary nor do they attempt to test different handling methods. There are a number of other studies in the wet granulation literature which make use of ultrafine powders<sup>24,35-41</sup>. Often, only one of several materials used was in the ultrafine range<sup>3,5,6,16,27-30,36,39,41</sup>.

There are a few examples in the literature of authors studying how nucleation of ultra-fine powders occurs. The theories/models proposed do well with the properties that they measure but do not attempt to look at the granule microstructures in a quantitative manner to test their proposals. There is no explanation or examination of whether a particular mechanism can result in different microstructures. There have been no systematic studies the effects of pre-granulation upon standard granule characteristics or their microstructure. Some of the authors who have used ultra-fine powders are primarily interested in post granulation uses and are not especially concerned with initial granule structures that they have formed.

Table 2.3: Literature studies of wetting and nucleation behavior using ultra-fine

Authors	Size	Materials	Comments
Hapgood et al. <sup>3</sup>	0.73-113	Glass ballotini, lactose, zinc oxide, titanium dioxide	Examined kinetics of drop penetration for single droplet granules. Proposes a new two-phase drop penetration model
Emady et. al. <sup>5,16</sup>	2.97-67.5	Not Stated	Creates regime map based upon Bond number to predict nucleation mechanism. Also proposes models to predict behavior of particles and agglomerates.
Emady et al. <sup>6</sup>	2.97-15	Not Stated	Identifies regimes and mechanisms for single droplet nucleation
Hapgood and Khanmohammadi <sup>36</sup>	3.76-185	Glass ballotini, Salicylic acid, 2-Ethoxybenzamide	Nucleation of hydrophobic powders using single droplet experiments
Eshtiaghi and Hapgood <sup>27-29</sup>	0.01-320	Fumed silica, polytetra fluoroethylene, Glass Beads	Examination of formation of liquid marbles from hydrophobic powders. Updates a proposed quantitative framework for this mechanism.
Van den Dries et al. <sup>30</sup>	6,10,60	Lactose	Examination of nucleation mechanisms in a high shear mixture. Creates a model using liquid penetration, binder dispersion and granule breakage. Results presented based upon changes in viscosity
Rough et al. <sup>31-33</sup>	4.87	Zeolite (P&G)	High-shear mixer granulator using highly viscous paste material as binder. Examines behaviors of interest to P&G for this specific combination of materials. Binder choice means behaviors are likely unique.
Afarani et al. <sup>34</sup>	1, 6.8-8.1	Alumina, Silicon dioxide	Examining possibility of producing alumina granules in high-shear mixer for production of ceramics. Main objective is to study effects of sintering

			temperature, but binder % and L/S ratio are also tested.
Saleh et al. <sup>35</sup>	2.7,8.6	Alumina	Granule growth behaviors for alumina in a high-shear mixer. Identifies 3 regimes of growth behavior based upon L/S ratio.
Charles-Williams et al. <sup>37</sup>	5.8 and 2	Granulac 230 and hydrophobic limestone	Granule growth behaviors of hydrophobic and hydrophilic mixtures. Proposes preferential nucleation and layering mechanism
Zizek et al. <sup>38</sup>	9.67	Dolomite	Comparison of PBE and EKE coalescence kernels to experimental granule size distribution of dolomite and three-component binder formulation
Sochon et al. <sup>24</sup>	0.2	Zinc Oxide	Focus is on strength testing of agglomerates, but also tests granule growth at various levels of binder content for zinc oxide powder
Herd <sup>39</sup>	4-66	Fe-Al phosphates	Short communication on induction times as a function of water content
Johansen and Schæfer <sup>40</sup>	4.4-4.8	Calcium carbonate (Durcal)	Examined binder liquid requirements and growth mechanisms for melt agglomeration. Varied size distribution, surface area and shape
Fu et al. <sup>41</sup>	9-49	Calcium carbonate (Durcal)	Comparison of granule collisions to theoretical models (Type I & II) through measurement of restitution coefficient and max contact area

## 2.4 Suspension Based Agglomeration

There are examples of other processes using ultrafine powders to create agglomerates with a variety of internal structures. The internal structure of these agglomerates has an impact on its behavior during further processing. For example, Eckhard and Nebelung [2011] showed a change from ductile to brittle behavior in the compaction of spray dried agglomerates by changing the structure from homogeneous to inhomogeneous<sup>42</sup>. They explain this by stating that inhomogeneous or “hollow” spray-dried granules can be created by using a well dispersed suspension which allows for particle mobility during drying. They cite the following work from Pagnoux et al. as proof<sup>43</sup>.

A different suspension based technique used by Pagnoux et. al. makes large, spherical agglomerates directly in suspension through continuous stirring of primary agglomerates created from alumina (0.4  $\mu\text{m}$  average) and silica (25 nm)<sup>43</sup>. The granule structure was changed from solid to hollow by adding an additional step to the primary agglomeration stage, producing a narrower primary agglomerate distribution, described in Figures 2.6 & 2.7. The application of shearing after the ultrasonic treatment was found to reduce the overall size of agglomerates.

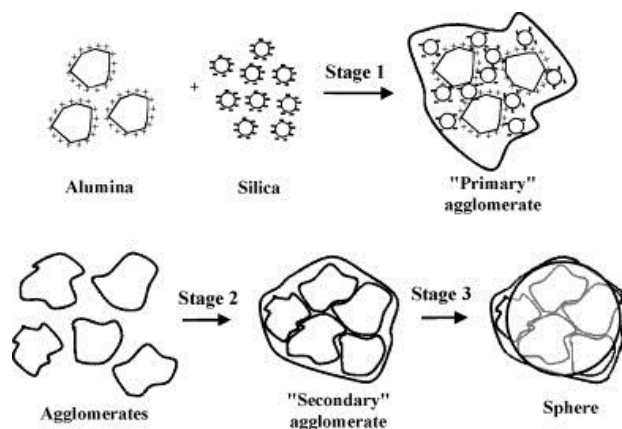


Figure 2.6: Proposed agglomeration method involving primary and secondary agglomeration in a suspension<sup>43</sup>.

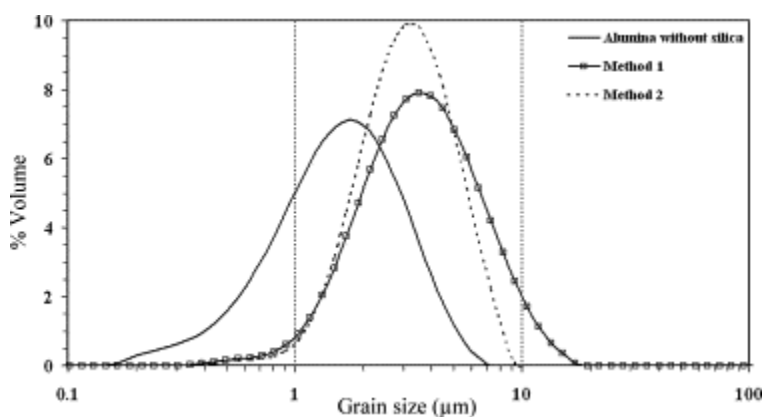


Figure 2.7: Size distribution of primary agglomerates formed in suspension by Pagnoux et al.<sup>43</sup>. Note narrower side distribution derived from Method 2.

The homogenous granule structure, when dried, presents with cracks in the granule which are a common feature of drying of dense agglomerates where the particles cannot move. The right hand side of Figure 2.8 shows the hollow granule after granule sintering, as an explanation for the cracks observed. Pagnoux et al. prove that the wet granule is solid by freezing and slicing open a wet, dry, and sintered granule. In their

images in Figure 2.9, it can be seen that the wet granule is solid and a hole appears in the dry and sintered granules.

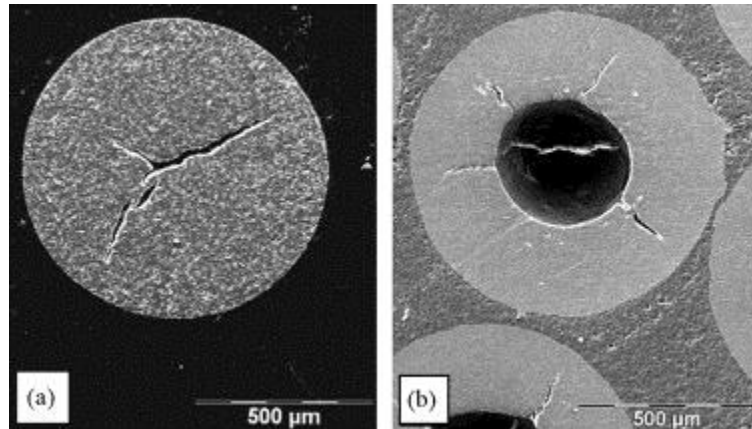


Figure 2.8: SEM micrograph of the inner structure of a sphere elaborated by colloidal granulation according to (a) method 1, (b) method 2 of powder deagglomeration<sup>43</sup>.

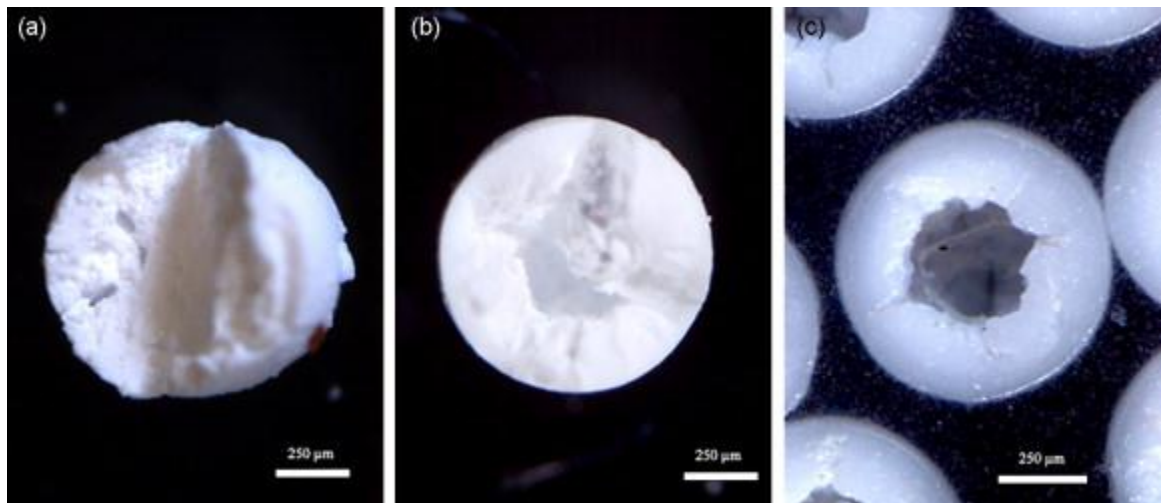


Figure 2.9: Images of frozen spheres (a) before drying (b) after drying (c) after sintering<sup>43</sup>.

It is possible that the internal granule structures can also be controlled through changes in handling in traditional wet granulation applications. An imaging technique, such as X-ray tomography, capable of quantifying internal structures must be used to verify this

concept. The focus of suspension based agglomeration papers is typically upon the further processing of the materials into ceramics and the resulting structures rather than upon the individual granules.

## 2.5 Fluidization of Ultra-Fine Powders

One method for handling of cohesive materials (Geldart Group C), including ultra-fine powders, is the use of fluidized beds<sup>44-51</sup>. The standard behaviors that arise in fluidization of cohesive materials are plugging, channeling, bed disruption and agglomeration with that last being of potential interest for this study. A series of experiments performed by Wang et. al. on a set of powders ranging in size from 0.01  $\mu\text{m}$ -18.1  $\mu\text{m}$  showed a tiered arrangement to the resulting fluidized bed<sup>44</sup>. The bottom of the bed consisted of a fixed layer of large agglomerates (2.8 mm) made from fine particles (approximately 7  $\mu\text{m}$  in diameter) with the next layer above consisting of fluidized 0.3 mm agglomerates formed from 17.8  $\mu\text{m}$  average particles and the upper layer consisted of fully fluidized discrete primary particles<sup>44</sup>. Other works have found that the fluidization conditions can be improved through the use of vibration, which can reduce the difference in size between agglomerates at the top and bottom of the bed such that the agglomerates of 5.5  $\mu\text{m}$  CaCO<sub>3</sub> were 400-500  $\mu\text{m}$  in all sections of the bed<sup>45-48</sup>.

The techniques that have been described here are a method that could be used for controlling the structure of a bed formed from ultra-fine powders. These papers



indicate that ultra-fine powders can form complex structures during handling and bed preparation.

## 2.6 Review of X-Ray Computed Tomography

### 2.6.1 Introduction

X-ray tomography (XRT) images internal structures through variations in X-ray absorptivity<sup>8</sup>. It has the advantage of being a non-invasive and non-destructive way of obtaining information on the internal structure of an object of interest. One advantage of the non-destructive nature of the test, as applied to the material sciences, is that it allows the sample to be imaged, retrieved, experimented upon, and imaged a second time to see specific changes as a result of the experiment. This allows flaws or defects in a material to be observed and their propagation measured when stressed. The term X-ray computed microtomography (microCT) is actually an arbitrary distinction used to define the scale of interest. A useful definition for microCT is the imaging of volume elements of  $\sim 50 \mu\text{m}$  voxels or smaller in comparison with conventional tomography or nanoCT (which purports to have a spatial resolution significantly below  $1 \mu\text{m}$ )<sup>7</sup>.

MicroCT depends, in its simplest description, upon using the measurement of the “transmitted intensity of a finely collimate beam of radiation”<sup>52</sup>. A single two-dimensional slice of the object can be created (mathematically reconstructed) through measuring this intensity over many different ray paths through the sample. The object can then be moved up (or down) relative to the radiation source to obtain a new slice and is repeated until the entire object has been imaged. Further work can then be done

to take this series of slices and create a three-dimensional representation of the object and/or the features of interest. These differences in X-ray attenuation are closely related to the material density and changes in the density are often associated with material and phase boundaries <sup>53</sup>.

### 2.6.2 X-Ray Absorption

The variation in the measured intensity of X-rays (also known as the attenuation) of wavelength  $\lambda$  is described by the Lambert-Beer law <sup>52</sup>:

$$\frac{I}{I_0} = e^{-\mu x} \quad (2.16)$$

$I_0$  is the intensity of the original X-ray beam,  $I$  is the intensity of the beam after it has traversed a layer of material of thickness  $x$ , and  $\mu$  is the linear attenuation coefficient.

This linear attenuation coefficient is dependent upon both the mass attenuation coefficient and the mass density which, when substituted back into (2.16) gives:

$$\frac{I}{I_0} = e^{\left(\frac{-\mu}{\rho}\right)\rho x} \quad (2.17)$$

Where  $\frac{\mu}{\rho}$  is the mass attenuation coefficient (typically in  $\frac{cm^2}{g}$ ) and  $\rho$  is the mass density ( $\frac{g}{cm^3}$ )<sup>52</sup>. This is an end-point relationship which is not as useful in determining specific structural features. It is of greater use to describe the attenuation using the differential form which will examine what occurs within each small thickness element  $dx$  along a given ray path<sup>8</sup>:

$$\frac{dI}{I} = -\frac{\mu}{\rho} dx \quad (2.18)$$

Simplifying this equation using the assumption of the smallest possible value of  $dx$  (realistically speaking), the value of  $\frac{\mu}{\rho}$  can be assumed constant over a particular thickness element and just written as  $\mu$ . The objective then shifts to knowing the absorption coefficient at each point along the ray path in order to perform the reconstruction. The primary issue in computed tomography is then to assign the correct values of  $\mu$  to each position while knowing only the values of the line integral for various ray paths <sup>8</sup>:

$$\int \mu(s) ds = \ln\left(\frac{I_0}{I}\right) \quad (2.19)$$

where  $\mu(s)$  is the linear absorption coefficient at position  $s$  along ray  $\mathbf{s}$ . This reconstruction of the objects total volume can then be performed by collecting high resolution radiographs from enough different directions. The exact method used for the reconstruction will depend upon the method of data collection, with a variety of choices being available both for collection and reconstruction of data <sup>54</sup>:

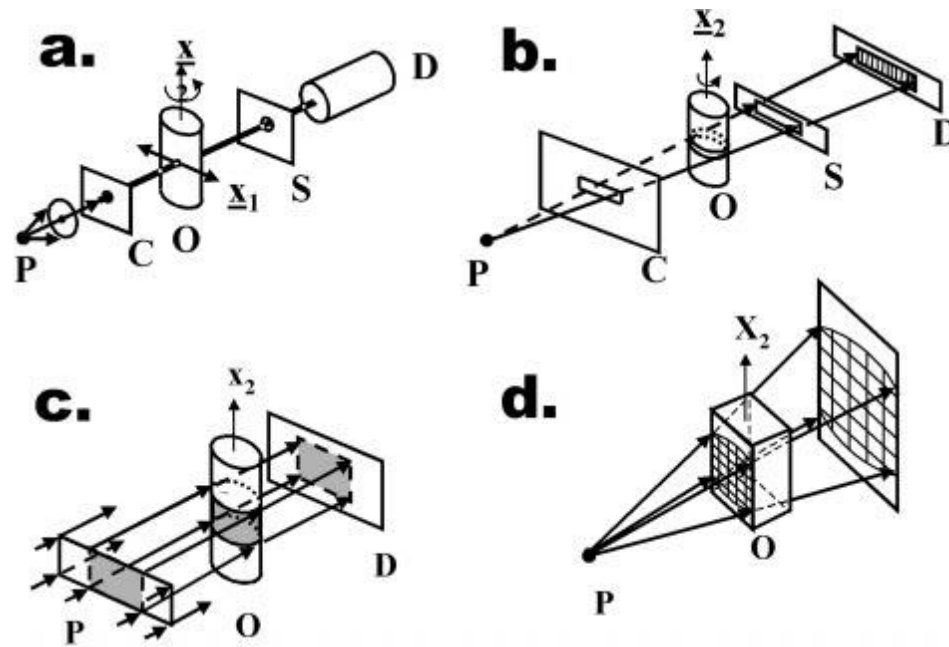


Figure 2.10: Illustration of four experimental approaches to X-ray microtomography data collection: a pencil, b fan, c parallel, and d cone beam methods. P is X-ray source, C is collimator, O is object being imaged,  $X_2$  is axis about which sample is rotated to produce different views required for reconstruction, S is slit, and D is detector<sup>8</sup>.

There are two major types of reconstruction methods, iterative and analytical reconstruction<sup>55</sup>. The preferred algorithm for the reconstruction of cone-beam data appears to be the Feldkamp, Davis, Kress (FDK) technique based upon filtered back projection, an analytical reconstruction method, which is considered to provide the best mix of efficiency and acceptable results<sup>56–58</sup>.

### 2.6.3 Image Resolution

The resolution presented by microCT is then a function of the number of detectors and the field of view (FOV) being analyzed to produce the minimum voxel size. In order to create a complete reconstruction of the object of interest, it is necessary that the FOV is at least as large as the maximum specimen diameter for a given slice<sup>7</sup>. As an example,

if the object of interest is 10 mm in diameter and the detector has one thousand elements, the reconstruction voxel size will be (should be) 10  $\mu\text{m}$ . If only a smaller portion of the object is actually of interest, the FOV can be smaller than the total object so long as the section of interest is in the FOV throughout all rotations<sup>7</sup>. Typical resolution limits for microCT are in the range of in the range of 6 to 15  $\mu\text{m}$  with the smallest reported voxel sizes being on the order of 0.5 to 0.87  $\mu\text{m}$  for the Phoenix X-ray Nanotom equipment though some reports of resolutions less than 8 nm for small scan sizes<sup>7,59</sup>. The nanometer resolutions are typically referred to as nanoCT.

The amount of rotation done while moving from a given  $s$  to the next  $s$  will also have an impact upon the resolution, with a smaller angle leading to a sharper, more (potentially) accurate image. The selection of a smaller angle will also increase the time necessary for scanning and reconstruction as it increases the amount of data collected. For a high resolution image (smallest voxel size) an angular step value of 0.25° (for a 180° degree rotation, this would lead to 720 projections being taken through the object) is found to be an acceptable value beyond which the trade-off between time and improved image quality is usually not significant<sup>7</sup>. This is important in order to avoid aliasing of the image, which can destroy the fine detail that is desired in microCT.

Other factors which will impact the level of contrast in the image include the X-ray intensity flux and the exposure time which will impact the number of X-ray photons striking the detector per pixel<sup>7</sup>. The detectors are limited in the number of photons that can be absorbed before becoming saturated, at which point the reconstructed image

begins to lose integrity. In the case where the X-rays are strong enough to pass through the sample (i.e. the attenuation is not complete) then for a long enough exposure time all detectors will be saturated and the entire image will be white or black (a function of the reconstruction program that assigns black as being either the maximum or minimum intensity). Higher energy X-rays will become less attenuated by passing through the sample, but will also be less sensitive to changes in the density or phase<sup>53</sup>.

Another option for improving the resolution of the image requires the taking of multiple scans and averaging those results to get the final image. This can be an especially useful technique for separating out relatively low density areas which have low attenuation because it improves the signal to noise ratio of the results. The primary drawback to using this averaging technique is that it will significantly increase the necessary scanning time, increasing by a factor equal to the number of scans desired for making the averaged imaged. The usage of a 14-bit detector can result in a similar signal to noise ratio as that obtained from four frame averages from a 12-bit detector while also giving improved contrast<sup>7</sup>.

#### 2.6.4 Potential Image Artifacts

In computed tomography, the potential exists for a variety of artifacts that represent a “systematic discrepancy between the CT numbers in the reconstructed image and the true attenuation coefficients of the object”<sup>54</sup>. The primary types of artifacts which can occur are summarized in Table 2.4 and are categorized by how they appear in the reconstructed image<sup>54</sup>:

Table 2.4: Types of CT Artifacts based upon effect on reconstructed image

Artifact Type	Description
Streaking	Generally due to inconsistency in a single measurement
Shading	A group of channels or views deviating gradually from the true measurement
Rings	Caused by errors in an individual detector calibration
Distortion	Caused by helical reconstruction

while the origin of these artifacts can then be grouped into four categories <sup>54</sup>:

Table 2.5: Groupings of causes for different CT artifacts

Artifact Basis	Cause
Physics Based	Result of the physical processes involved in the acquisition of CT data
Patient (Object) Based	Patient movement or metallic materials in or on the patient
Scanner Based	Imperfections in Scanner Function
Helical & Multi-section	Introduced by the image reconstruction

Some, but not all, of the specific types of artifacts will be discussed here and will include some information on how these issues are dealt with in microCT.

### 2.6.5 Beam Hardening

Beam hardening is a physics-based issue that results from the X-ray beam used for data collection containing individual photons that have a distribution of energies (polychromatic radiation). As the beam travels through the material, the lower energy photons are absorbed faster than the high energy photons, the effect of this is to “harden” the beam, so that the average energy of the beam can actually increase, rather than decrease <sup>8</sup>. Therefore, equations (1)-(4) are no longer perfectly valid. This can lead to two particular types of artifacts: cupping artifacts and dark bands or streaks

between dense objects in the image. These streaks and dark bands result from differences in how much the beam is hardened as it passes through particular objects from different angles. The cupping effect is the result of the beams hardening more when passing through the thickest portions of the sample as opposed to the edges and so the intensity recorded on the detector is higher than expected<sup>54</sup>. Manufacturers have worked to deal with beam hardening in several different ways. Attempts have been made to use a metallic material to “pre-harden” the beam by filtering out low energy photons, by introducing a calibration correction through the use of phantoms in the desired size range which increases the CT values recorded for certain sections (an example of this is seen in Figure 2.11), and the creation of an iterative correction algorithm that can be applied when certain types of regions are being reconstructed (such as bony regions, where this commonly occurs)<sup>54</sup>.

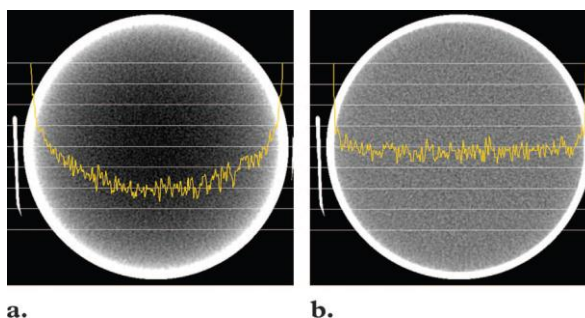


Figure 2.11: CT number profiles obtained across the center of a uniform water phantom without calibration correction (a) and with calibration correction (b)<sup>54</sup>

#### 2.6.6 Photon Starvation

Photon starvation is a physics-based problem in imaging areas of an object with high attenuation potential. The result, as the name implies, is that the detector registers a



low intensity due to the lack of photons reaching it. This results in projections with magnified amounts of noise, which results in horizontal streaks in the image<sup>54</sup>. The simplest way to overcome this is to increase the peak voltage of the beam which increases the mean photon energy to improve the signal to noise ratio in those areas. However, there are several potential issues with merely increasing the energy. The first is that some instruments do not have much variation allowed in the energy source (essentially a low, medium, and high setting) and only operate at certain predetermined settings. A second potential issue is, as previously mentioned, that the use of higher energy X-rays can decrease the image contrast for areas of the object that do not suffer from high amounts of attenuation. The third issue is more of an issue with a biological sample (and especially in imaging of human patients) where increasing the energy level will unnecessarily increase the radiation dosage received by the patient when scanning sections of the body that do not suffer from the higher attenuation. Some manufacturers of equipment have fitted their instruments with systems that automatically vary the energy during the course of the scan to account for the variations in attenuation (generally focused on the changes in sample width) while others have taken to the use of adaptive filtration to smooth out sections in the attenuation data for areas above a certain threshold before the reconstruction occurs<sup>54</sup>.

#### 2.6.7 Metallic Materials

The image artifacts produced by imaging of metallic materials fall into the “patient-based” category. Many typical metal objects have densities beyond the range handled

by traditional equipment which results in an incomplete attenuation profile<sup>54</sup>. This results in an image that lacks detail in certain locations as well as producing many of the previously mentioned artifacts, from beam hardening to aliasing. Metallic materials are one of the most difficult issues to deal with in computerized tomography. The simplest way to deal with this sort of issue would be to increase the voltage of the equipment. However, even at the highest available voltage it can be impossible to obtain anything other than some highly suggestive hints as illustrated in Figure 2.12 which represents an attempt at imaging a single granule formed from zinc oxide particles. There exist some specialized software packages that can deal with the streaking issue through the use of a “variety of interpolation techniques to substitute the over-range values in attenuation profiles”<sup>54</sup>. However, such software packages cannot fix the lost detail that exists in and around the metal object. In many cases, especially in larger areas of interest such as the interface between metal implants and tissue, this area where the detail has been lost was the primary area of interest<sup>54</sup>.

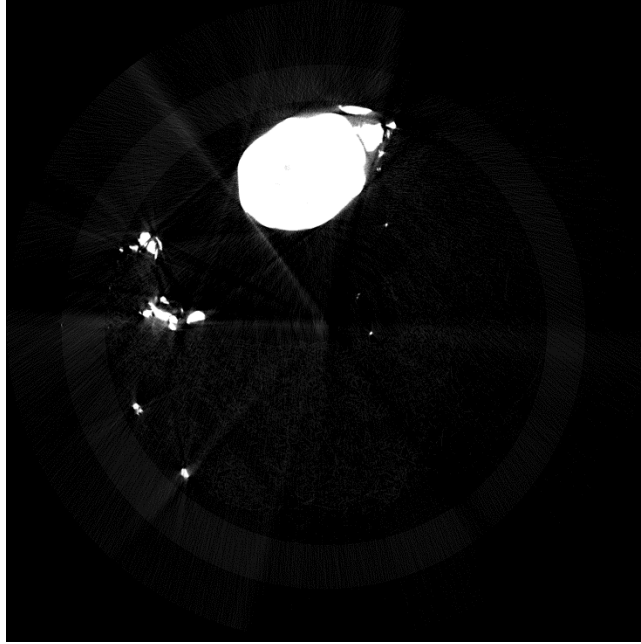


Figure 2.12: An image slice of a granule formed using sub-micron zinc oxide particles.

The material is too dense to be properly imaged by the equipment.

#### 2.6.8 Object Motion Artifacts

A different object based artifact relates to the effects of the object moving while the data collection is taking place. This will typically cause the existence of shading or streaking in the image as the area being scanned changes during scanning. This is typically a greater issue with live patients who are breathing or can be restless, but is an issue which requires some steps in sample preparation for inanimate objects. In the case of inanimate materials, a common step is to pack cotton above and below the material so that it is held in place. This works well for larger samples that completely fill the container. If the object is smaller than the sample holder the cotton has been observed to envelope the sample in question. This can make segmentation of the cotton from the object of interest somewhat difficult if the attenuation coefficients for

the materials are similar. A different technique which has worked well is to cut small wooden plugs of similar diameter to the sample holder to separate and hold the granules in place. Object motion is typically not an issue in microCT for the material sciences.

#### 2.6.9 Ring Artifacts

A ring artifact is the result of one or more detectors being out of calibration and giving incorrect readings at each angular position which causes a circular artifact to appear in the image [5]. The ring artifacts are not necessarily a significant issue from an image analysis standpoint depending on where they occur, but are an indication that the system needs to be recalibrated or repaired. Modern scanners make use of software that can characterize and reduce the potential effects and existence of ring artifacts <sup>54</sup>.

#### 2.7 Quantitative XRCT Analysis of Granule Microstructure

Granule structure and porosity has commonly been measured using mercury porosimetry, gas adsorption, and envelope density measurement, eg. Geopyc. A nondestructive technique that can be used to visualize the true internal morphology of granules is X-ray computed tomography (XRCT). There are a variety of studies which have used XRCT to study microstructure of granules, powder beds, or to describe void space in other types of applications<sup>7-10,59-70</sup>. Readers wishing for a fuller discussion of XRCT and all uses in the literature are directed to a pair of review papers by Stock or the dissertation of Steven Dale<sup>7,8,10</sup>. A variety of methods are used to quantify the granule structures that are observed. Porosity can be measured by XRCT, but several authors

have noted that there are deviations from measurements with other techniques, including mercury porosimetry, that make such determinations suspect for powder materials<sup>42,59,60</sup>. Using particles that are similar in size or smaller than the voxel resolution results in loss of information and can cause either over or under prediction of granule porosity. This will likely be an unavoidable issue when working with ultra-fine powders.

A study by Farber et al examined the differences between porosity measurements and pore size distributions obtained from mercury porosimetry and XRCT results<sup>60</sup>. They determined that mercury porosimetry determines the pore neck size distributions and tomography measures the true size distribution of pores of a size equal to or greater than the voxel size. They determined the XRCT pore size distribution through unfolding from equivalent projected area diameters. The techniques which they describe could be useful where approximate pore diameters are useful, but does nothing to describe the overall structure or the locations of the pore space within the granule.

In their study of high-shear granulation of calcium carbonate (Durcal 65), Rahmanian et al., scanning at less than 1  $\mu\text{m}$  voxel resolution, observed the creation of different microstructures depending upon the impeller tip speed<sup>59</sup>. They attempt to describe the differences in observed XRCT structure through porosity, a solid-phase cord length distribution, and a solid-phase correlation length. They use the chord length as an attempt to measure the average pore size and note that large values of chord length correlate to low measured porosity values. This technique allowed them to show that

certain granules are more heterogeneous in nature but does not do a good job of describing the location of voids or particles.

Eckhard and Nebelung developed a “quotient H” to correlate the average diameter of the largest pore to the average granule diameter by the following <sup>42</sup>:

$$H = \frac{\frac{(d_{\text{void1}} + d_{\text{void2}} + d_{\text{void3}})}{3}}{\frac{(d_{\text{granule1}} + d_{\text{granule2}} + d_{\text{granule3}})}{3}}$$

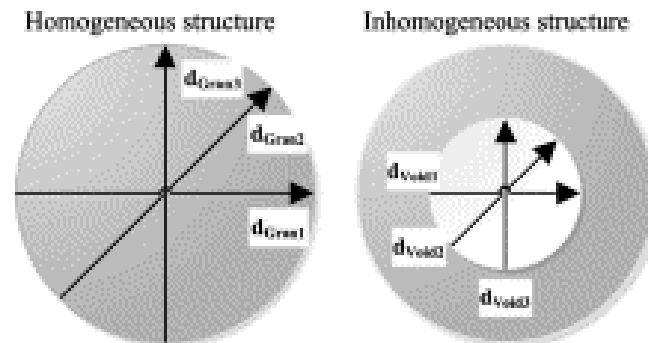


Figure 2.13: Quotient H and the different diameters for quotient H calculation. Dgran for homogeneous, dgran and dvoid for inhomogeneous. D1 is horizontal, D2 is diagonal, D3 is vertical<sup>42</sup>

where D1 is the horizontal, D2 is the diagonal and D3 is the vertical measurement from direct measurement of approximately 20 granules. A value of H near 1 represents a void almost as large as the granule, such that the solid material is packed into a solid shell. This would make it a hollow granule. A value of H near zero would indicate a homogeneous structure. For the purposes of their work, a pore was treated as a void if the average diameter was >10% of the granule diameter. This method does provide a sense of granule structure and comparisons between “H” values can be used to indicate

differences in structure. However, it does not address the actual location of void space within the granule and cannot tell if the void space is evenly distributed or not.

The creation of the medial axis, or “skeletonizing”, has been proposed by Lindquist and Venkatarangan as a method to determine the locations of throats in the pore pathways and pore pathway lengths<sup>61</sup>. They do this by uniformly dilating the medial axis segment as a solid cylinder until it contacts the grain surface of a pore pathway at which point dilation there ceases. They create a closed loop along the segment that defines the perimeter of a minimal cross sectional area that then is identified as the throat surface area. This is yet another method for describing the size of the granule pores, but it also fails to address the location of the void space within the granule.

An examination of differences in granule growth mechanisms by Le et al. used XRCT to identify differences between coalescence and consolidation growth granules<sup>62</sup>. They measure porosity and binder content of the granules. A study of Ni agglomerates from heap leaching by Nosrati et al. focuses on qualitative (not quantitative) information regarding granule composition, but does note that they used a minimum of 10 granules at each data point to account for variations from one granule to the next<sup>63</sup>. Two different papers describe the internal granule structure for a discrete void structure by fitting volume equivalent sphere diameters to the measured void volumes and creating a void size distribution<sup>64,65</sup>. Both studies make use of erosion followed by dilation to remove potential interconnections between voids with minimal loss of information, with one using spray-dried nanometric alumina granules and the other using iron ore

pellets. This technique does provide quantitative measures of void structure, but again does nothing for describing the location of void space.

A recently developed technique from Dale et al. (2014) shows great utility for describing a wide variety of microstructures in terms of void size and location<sup>9,10</sup>. His work developed a microstructure analysis method performed using MATLAB which uses binary thresholding to segment out and identify different phases of interest, in particular a binder and particle phase. For this work, glass ballotini makes up the particle phase and polyvinylpyrrolidone solution is used for liquid binder. The Dale method uses a convex hull wrapping to define the outline of the granule. The granule is wrapped in the XY, YZ, and XZ planes with only the overlapping sections kept to identify the void space. Individual connected objects can be isolated, identified and the object volume can be measured. Dale et. al. use a watershed transform to separate the glass ballotini primary particles and then determines contact numbers for each particle. The method of Dale et. al. also looks at axial distributions of the three phases as well as identifying the distance of the various phases from the edge of the granule. The use of the axial distribution is proved by showing how the binder phase is preferentially distributed to the top of the granule. These positional distributions, when coupled with volume/size measurements, make this method a uniquely robust approach amongst the current literature. The Dale method does not attempt to identify void shapes or to differentiate between different types of microstructure. This limits its usefulness as a



tool for analyzing granule microstructures on the basis of void space rather than binder distributions.

This microstructure information has been incorporated into DEM simulations to test various models of granule strength and predict how granules will break<sup>70</sup>. The Dale method was successfully used to create simulations of granule microstructure that are quantitatively accurate for spherical fluid bed granules. They do note that the simulations do not exactly describe experimental results because the DEM simulation used pure elastic solid bonds which cause brittle behavior and are not accurate representations of the elasto-plastic deformation of bonds in experimental testing.

XRCT is a powerful technique which has been widely used in attempts to describe granule microstructure. The current literature has a variety of techniques which attempt to characterize the structure through measurement of various properties, including porosity and various equivalent diameters. The Dale method is the best method currently available for describing both the size and location of the void phase within granules<sup>9</sup>. Other methods are lacking in one or both capabilities. The location of the void space is a key point in attempting to predict future behaviors. If the voids are not evenly distributed throughout a granule, certain sections can be expected to behave in different fashions. There is not currently a published method which quantitatively describes differences in void size and location for determining if changes in structure have occurred. XRCT has been shown to be a poor choice for determining the true

porosity of granules formed from ultra-fine powders due to loss of information, so other applications of XRCT should be used for this work.

## 2.8 Loss-in-Weight Feeding

A significant shift in recent industrial granulation processes is the push towards continuous manufacturing. This has resulted in a significant amount of research since 2005<sup>71</sup>. A subset of this research has begun, since 2012, to focus on the use of loss-in-weight feeders (LIW) as a method for continuous feeding of raw materials, specifically for pharmaceutical applications<sup>72-75</sup>. These papers cover studies of performance of different feeder systems, a methodology for characterizing tooling and feeder performance, issues related to hopper refill, and a case study of pharmaceutical formulation feeding.

The LIW feeders are of industrial interest because the use of control systems have improved the ability to control federate and minimize flow rate variability, however much of the information on feeder performance appears to lie with the equipment manufacturers and is not publicly available<sup>73</sup>. Engisch et. al. have noted specifically a lack of available information on (i) the effect of powder properties on flow rate intermittence and (ii) the effects of feeder design and operation on output powder properties<sup>73</sup>.

The first issue is important because pharmaceutical companies are interested in designing and feeding new active pharmaceutical ingredients (API's) into granulation processes. Pharmaceutical API's are commonly cohesive and poorly flowing which

results in potential failures during feeding, such as ratholing/arching in the hopper, compaction in the feed screws and surface buildup resulting in blockages<sup>74</sup>. There is industrial interest in predicting which materials will have this issue based on small scale material property tests, but this information is currently unavailable. A secondary issue to this is that most API's are fed at low flow rates and at the lower end of the equipment feeding capability which means inaccuracies in feed rates themselves are a proportionately larger issue<sup>74</sup>.

Engisch et. al. have developed and validated a method for evaluating the performance of a twin-screw LIW feeder (K-Tron KT35) for three pharmaceutical grade powders. Their method is to use relative standard deviation (RSD) of the mass flowrate data (calculated every second) determined by a catch-scale for the feeder running in gravimetric mode (with the control systems operational). This method is used to determine the effectiveness of various feed tooling selections, specifically the combinations of speed, screw type and screen. They propose that a predictive model for tooling selection could be created using this method if combined with a database of feeder performance and powder properties<sup>73</sup>.

A more recent publication by Meier et. al. claims that this methodology may lead to inaccurate conclusions because it only assesses momentary fluctuations of the feed rate<sup>76</sup>. It does not account for time intervals in fluctuations and therefore will ignore if the wrong mass is fed only during certain intervals but is within tolerances for most of the time. They note that the choice of sampling time for feed rate calculation is

incredibly important in whether or not the system appears to be feeding at the desired rate. Fast sampling rates (1s), which are desirable for control systems, shows a sinusoidal variation around the set-value while sampling times of 30s would suggest only small deviations from the set-value<sup>76</sup>.

## 2.9 Critical Summary

There have been significant advancements in the understanding of granulation in the last twenty years. However, there have not been many studies which have focused upon the complex behaviors that exist when working with ultra-fine powders. Literature exists for suspension-based agglomeration and fluidized bed granulation of ultra-fine powders which shows that the bed structure and granule microstructure can be changed through changes in pre-granulation handling. These studies indicate that it should be possible to control the formation of microstructure in other applications. Such studies have not been performed for wet-granulation of ultra-fine powders, nor has a systematic study of pre-granulation effects been attempted.

There are also no published techniques for describing a wide range of microstructures in useful terms of the size, shape and location of void space within granules. The recent method developed by Steven Dale shows great promise for describing the granule microstructure, including size and positional distribution of the various phases. However, his methodology is incapable of distinguishing between distinctly different types of microstructures and has not been used to evaluate changes in microstructure as a function of time. It is unclear if his methodology is sufficiently sensitive to observe

these differences. There is space for both a systematic study of the formation of microstructure in ultra-fine powder granules and an analytical methodology for describing the different microstructures.

Several studies describe and model the nucleation behavior of granules. The key properties identified from these studies are the particle size, liquid viscosity, surface tension, and contact angle. The primary issues with these models is that they rely either on average primary particle or average aggregate sizes. These studies have not attempted to link granule microstructure with nucleation kinetics. In addition, these studies have not evaluated the use of ultra-fine powders. The nucleation immersion mechanism of Hounslow may be an accurate descriptor of ultra-fine powder behavior.

There is space in the literature for an experimental study of the growth kinetics of ultrafine powders, with the intention of developing quantitative models and/or regime maps which can predict specific behaviors. The available literature on imaging techniques shows that it is possible to examine changes in granule structure through the use of XRCT. The desired size range of study will make the examination of individual powder bed structures unlikely due to resolution limits, but should not pose a problem to granules formed from such powders.

The study of continuous feeding of raw materials using LIW feeders, both for ultra-fine powders and otherwise, is exceptionally sparse in current literature. The relevant information for predicting behavior of new materials based upon measurement of specific properties appears to reside in the proprietary hands of equipment

manufacturers. Currently, we cannot predict when, how and if powders, especially pharmaceutical API's, will fail during continuous feeding.

## CHAPTER 3. MATERIAL AND METHODS

### 3.1 Material Properties

$\alpha$ -alumina particles with 4 different median particles sizes ( $d_{50} = 0.5 \mu\text{m}$ ,  $5 \mu\text{m}$ ,  $25 \mu\text{m}$ ,  $108 \mu\text{m}$ ) supplied by Inframat Advanced Materials) were used to form granules for study. Particle size characterization was performed by wet dispersed laser diffraction (Malvern Mastersizer 2000). The  $25 \mu\text{m}$  alumina dispersed easily in water with ultrasonics set at 50%. A dispersant solution of IGEPAL CA-630 in water (1 g/1000 g water) was used for the other particles. Particles were in the ultrasonic bath for 5 minutes at 50% ultrasonic intensity before measurements were taken. The  $108 \mu\text{m}$  particles broke up when dispersed (to around  $1 \mu\text{m}$ ) so a Tyler Ro-Tap Model E Sieve Shaker was used for sizing. The sieve shaker was run for 5 minutes using the Fine Analysis option using a sieve stack sized  $500 \mu\text{m}$ ,  $355 \mu\text{m}$ ,  $250 \mu\text{m}$ ,  $180 \mu\text{m}$ ,  $125 \mu\text{m}$ ,  $90 \mu\text{m}$ ,  $63 \mu\text{m}$ . The powder flow properties have been evaluated using a Jenike & Johanson RST-XS Schulze Ring Shear Tester using 1,2,4 kPa pre-shear values with automatic shear point selection. Primary particle size and shear cell results are in Chapter 4.3.1. A Nikon SMZ1500 microscope was used for optical imaging of granules from each experiment. Ten granules from each experiment were randomly selected for XRCT imaging to examine and analyze the internal structure. Additional granule images are found in Appendix B.

Water was used as the liquid binder either as a solution with dissolved polyvinylpyrrolidone (PVP) or as pure water. PVP K29/32 was used in concentrations of 5g/100g H<sub>2</sub>O or 10g/100g H<sub>2</sub>O and PVP K90 in concentrations of 5g/100g H<sub>2</sub>O or 7g/100g H<sub>2</sub>O to create binder solutions. Solution concentrations are referred to hereafter in the form 5% PVP K32 or 7% PVP K90 as appropriate. PVP K32 solutions were mixed in 20 mL scintillation vials using 10 mL of H<sub>2</sub>O and shaken for 10 minutes to dissolve PVP K32 in solution and left for 10 minutes for bubbles to rise to the surface. PVP K90 solutions were mixed in 250 mL Erlenmeyer flasks using 100 mL of H<sub>2</sub>O and mixed overnight (minimum of 12 hours) using a magnetic stir bar at setting 7. Solution surface tensions were measured using a Kruss K100 Tensiometer. Solution viscosities were measured using a Brookfield DV2+ Pro Viscometer. Droplet size measurements were performed using a Photron FASTCAM high speed camera and assuming the droplet is an ellipse axisymmetric about the axis parallel to gravitational pull. Measured liquid properties can be found in Table 5.1 in Chapter 5.2. Raw data for liquid properties and droplet size measurements are found in Appendix C.

### 3.2 Static Bed Granulation

A Tyler-Ro Tap Model E Sieve Shaker was used (when necessary) to pass powder through 1.4 mm, 710  $\mu$ m, and 500  $\mu$ m sieves into a bottom pan containing three Pyrex dishes. A 22 gauge needle and a Hamilton Gastight 1700 syringe were used. A 3-prong adjustable clamp was used to hold the syringe in place while attached to a ring stand.



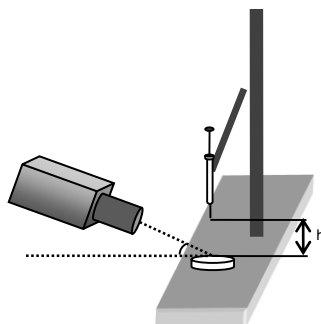


Figure 3.1: Diagram of Static Bed Granule Production, provided by Heather Emady

All powders were sifted using 1.4 mm, 710  $\mu\text{m}$  and 500  $\mu\text{m}$  sieves. All beds were created by placing 3 Pyrex dishes in a bottom pan and sifting powder directly into the dishes. The beds were smoothed with a straight edge after filling. 108  $\mu\text{m}$  primary particles were sifted by running a scoop along the outer edges of the sieve which caused the primary particles to pass through into the dishes. 25  $\mu\text{m}$  alumina passed through the 1.4 mm sieve by the same procedure. The sieve shaker was used on a fine analysis for 710 and 500  $\mu\text{m}$  sieves and run until the sieve had cleared (approximately 20 seconds). Additional material was added to the sieve and the process repeated if the dishes were not full. 0.5  $\mu\text{m}$  alumina was passed through the 1.4 mm and 710  $\mu\text{m}$  sieves on a fines analysis 5 times before sifting into powder beds. The material began to form stable agglomerates after the first pass. The 500  $\mu\text{m}$  sieve for 0.5  $\mu\text{m}$  alumina required a coarse sieving in small quantities (100 g or less) because of sieve blinding. The stable agglomerates were collected and the cake was broken up. The caked material and fresh powder were then passed again through the 500  $\mu\text{m}$  sieve on a coarse analysis. This was repeated until 600 g of agglomerates were collected. The 500  $\mu\text{m}$  sieve agglomerates were passed through on a fine sieve analysis to form the powder bed. 5

$\mu\text{m}$  alumina was passed through the sieves on a fines analysis (for 1 minute) 5 times before sifting into the petri dishes to break up large agglomerates.

Approximately 300 granules were created in each experiment using a 22 gauge needle from each powder/sieve combination. Liquid droplets were created by hand with a release point 0.7 cm above the powder bed surface. The powder bed was rotated to find a clean surface for a new granule to be formed. Water ( $2.71 \pm 0.03$  mm diameter droplets) was used as the binding liquid for submicron and 5  $\mu\text{m}$  alumina powder beds. A solution of 0.05 g PVP K32 (manufacturer name) per gram of distilled water ( $2.76 \pm 0.06$  mm diameter droplets) was used for the 25  $\mu\text{m}$  and 108  $\mu\text{m}$  granules because the granules did not have sufficient dry strength for analysis with water as the binder.

Granules were collected on a 2.8 mm and 2.00 mm sieve and dried in a Mettler Toledo Halogen Moisture Analyzer at 100 °C for 20 minutes. A Nikon SMZ1500 microscope was used for optical imaging of granules. The material and sieve preparation will be referenced using the letter and number combinations in Table 3.1, such as Powder A1 or Powder B3.

Alumina Size ( $\mu\text{m}$ )	1.4 mm Sieved	710 $\mu\text{m}$ Sieved	500 $\mu\text{m}$ Sieved
0.5	A1	A2	A3
5	B1	B2	B3
25	C1	C2	C3
108	D1	D2	D3

### 3.3 Drum Granulation

A tumbling drum with a 30 cm internal diameter was run at 25 rpm to evaluate time based effects on granule microstructure. A 22 gauge needle with a release point 12 cm above the drum wall was used to create granules. The needle and syringe were the same items used in the Static Bed Granulation, described in Chapter 3.2. A 2.8 mm was used for granule retrieval and a 1.4 mm sieve was used for powder preparation.

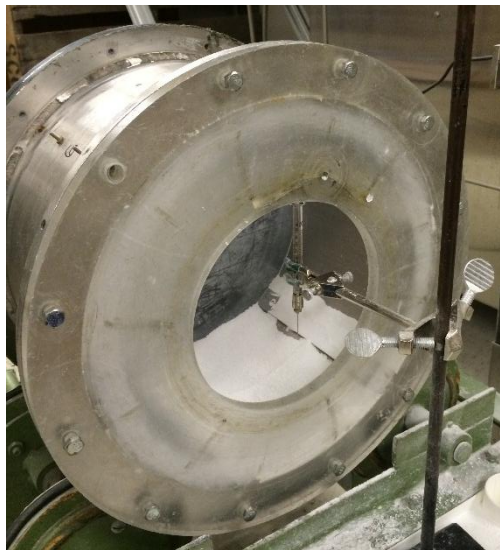


Figure 3.2: Drum Granulation Experimental Setup

Powders A and C were passed through a 1.4 mm sieve prior to being placed in the drum. Powder A was also run for 5 minutes at 25 rpm, forced through a 1.4 mm sieve, and repeated again because of dry agglomerate growth. Large agglomerates which remained on top of the sieve were forced through using a flat bladed scoop. After the 2<sup>nd</sup> run and sieve procedure, the growth of dry agglomerates was observed to be minimal over a 15 minute period. Fresh Powder A was processed for each liquid binder

using a 5 minute run in the tumbling drum and 1.4 mm sieving to prepare a consistent bed.

Powder A experiments were performed with 650-700 grams in the drum. Powder A granules were retrieved after 3 seconds, 5 minutes, 10 minutes, and 15 minutes. The drum was run for 1 minute at 25 rpm before liquid addition. This was done to return the bed to a consistent starting state after agglomerate breakage from sieving. One granule was created for the 3 second experiment and 5 granules were created for each 5, 10 and 15 minute experiments. Liquid addition took 12-15 seconds for the 5, 10, and 15 minute experiments. Granules were retrieved by scooping and pouring powder onto a 2.8 mm sieve and then placed into a petri dish for drying and storage. Powder A was passed through a 1.4 mm sieve by hitting the sieve/pan on the floor and the remaining agglomerates were forced through by hand before being returned to the drum for another experimental run. This was done to create a controlled and consistent starting bed condition.

Powder C experiments were performed using 1.1-1.2 kg of powder. The drum was run for 30 seconds at 25 RPM before liquid addition. There was no agglomerate formation observed for Powder C. Powder C granules were retrieved after 3 seconds, 10 seconds, 1 minute, and 5 minutes. One granule was created for 3 and 10 second experiments. Two granules were created for the 1 minute experiments and 5 granules were created for 5 minute experiments.

Liquid binder solutions of 10% PVP K32, 5% PVP K90, and 7% PVP K90 were used to create granules. A minimum of 15 granules were created and retrieved for each time point and liquid binder combination and 10 granules were selected for XRCT analysis. Extra granules were created to account for potential breakage during handling/transport or issues during the XRCT analysis process.

### 3.4 Image Analysis

#### 3.4.1 XRCT Analysis

A Scanco Medical microCT 40 (Purdue University) has been used for XRCT scanning. All alumina scans were performed at 70 kVp and 115  $\mu$ A at the high resolution settings resulting in a 6x6x6 $\mu$ m voxel resolution. Granules were placed in a 4 mm sample holder with ~4mm diameter wooden pegs used as spacers and allowing 5 granules to be scanned sequentially. Cotton wadding was used for Powder B granules because their fragility resulted in breakage when used with the wooden pegs.

#### 3.4.2 Image Preparation

The analysis of XRCT granule images was performed using ImageJ and MATLAB based on the analysis methodology developed by Dale et al.<sup>9</sup>. A copy of all the code can be found in Appendix A. An order of operations and which program is used can be found in Table 3.2. A binary threshold was applied in ImageJ using the automatic threshold selector to determine a specific value for separation of the granule from the background and is used consistently with granules of a specific material. Non-granule pixels above the threshold limit were manually removed for a majority of the static bed analysis. An

automatic method for noise elimination was developed in MATLAB which eliminates all voxels which are not part of the largest concentration of black voxels in the image. The large voxel concentration is assumed to be the granule while other voxels are assumed to be noise. A manual inspection of each automatically processed image stack was performed to ensure proper image cleaning. Some examples of extra objects that needed to be removed by hand are the cylinder walls or parts of the wooden peg which touched the granule surface and had some alumina dust from surface attrition. The automated image cleaning code was developed by Kelly Wang working as an undergraduate researcher under my supervision. The automated cleaning code takes 10-15 minutes granule and reduces manual cleaning time from 30-45 minutes down to 5-10 minutes per image stack.

Table 3.2: Image Processing Order of Operations and Program

Image Stack Cropping	ImageJ
Binary Thresholding	ImageJ
Automated Noise Removal	MATLAB
Visual Inspection/Manual Noise Removal (ImageJ	ImageJ
Feret's Diameter Measurement Parameter Selection	MATLAB
Convex Hull Wrapping	MATLAB
Macro-Void/Pore Space Separation in XY, YZ, ZX Planes	MATLAB
Macro-Void Labeling	MATLAB
Macro-Void Property Calculation (Volume, Center of Mass, Size, Surface Area)	MATLAB
Sphericity Calculation	MATLAB
Positional Distribution Measurement	MATLAB
Data Export	MATLAB to Excel

### 3.4.3 Granule Analysis

The edited images were processed in MATLAB to create a convex hull wrapping in the XY, YZ, and XZ planes and define the interior and exterior granule void space. Each distinct void was then labeled, with the center of mass, volume, volume equivalent sphere diameter, Feret's diameter, and surface area calculated. The measured and labeled voids were further processed to exclude small voids below a specific limit (micro-voids), either when viewing the stack as a 3-D reconstruction in ImageJ or during numerical evaluation of the voids. The 3-D reconstructions have been created using the 3D Image Viewer plugin in ImageJ. An example of this process is shown in Figure 3.3.

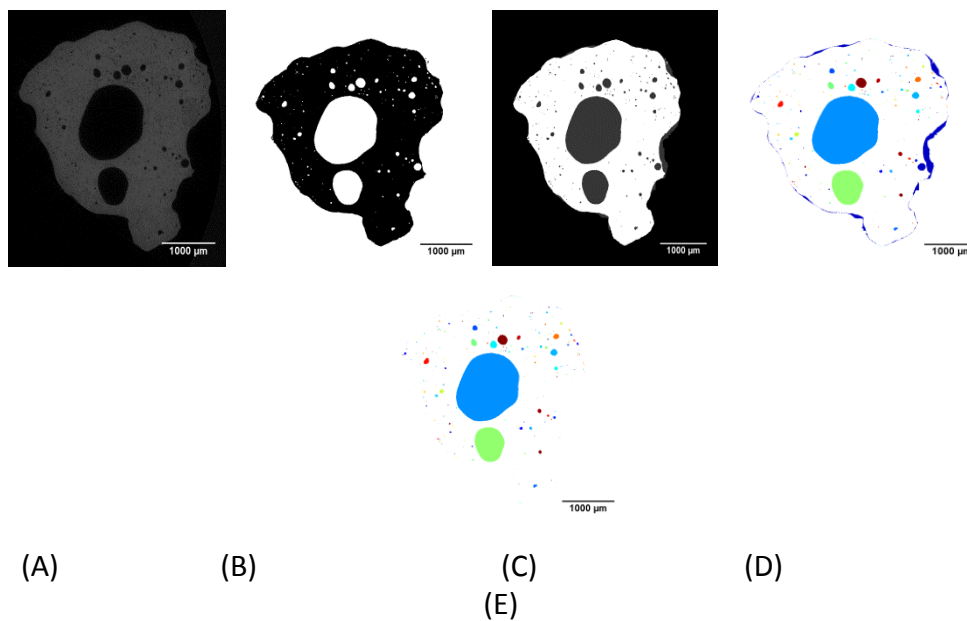


Figure 3.3: (A) Raw XRCT Slice, 0.5  $\mu\text{m}$  Alumina (B) Segmented Particles (black) (C) Convex Hull Output (white=particle) (D) Color Coded Connectivity Map (E) External Void Removed

The processed stacks were evaluated using MATLAB. The minimum Feret's diameter of each 2D void was calculated in the XY plane and were removed if the diameter was less than 30  $\mu\text{m}$  for the Powder A and Powder B granules and D granules. A 78  $\mu\text{m}$  limit was used for void exclusion in the Powder C granules and 224  $\mu\text{m}$  limit for Powder D granules which is 3 times the powder  $d_{50}$ . This value is set at 5 times the voxel resolution because that is approximately the minimum value needed to properly identify an object from noise. The value at 3 times the primary particle size is selected on the assumption that this is an appropriate amount of space which a single particle could easily fill. The Feret's diameter is the distance between two parallel planes restricting the object perpendicular to a given direction (see Figure 3.4). Figure 3.5 shows the effect of the 30  $\mu\text{m}$  limit for a static bed Powder A1 granule. The minimum value was determined by calculating 7 different values at 30 degree intervals from 0 to 180 degrees and selecting the smallest value. This was repeated, in order, for the YZ and ZX planes.



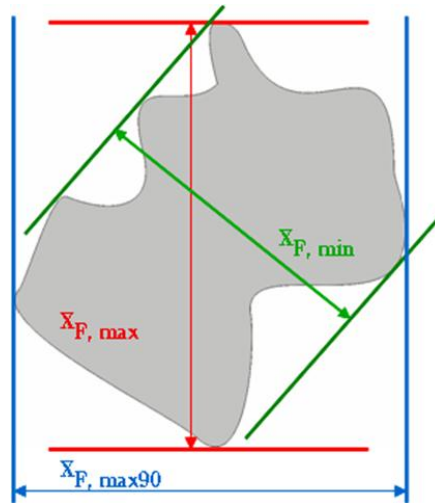


Figure 3.4: Diagram of Feret's Diameter, including minimum and maximum measurements<sup>77</sup>

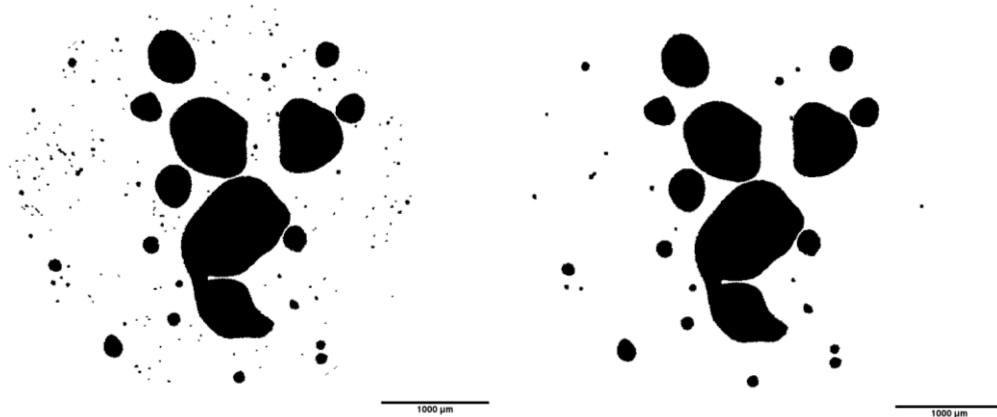


Figure 3.5: Demonstration of Feret's Diameter process for separating macro-voids from pore space. (A) Raw Void Space Image (B) Processed with objects below 30  $\mu\text{m}$  excluded

The total volume (in 3D) of the remaining voids, called macro-voids, was calculated by summing the voxel volume and converted to volume equivalent sphere diameters. The surface area of the 3D voids was also calculated using a simple counting of the number of exposed voxel faces for each labeled void. The sphericity of the labeled objects was then calculated to evaluate the shape of the macro-voids using the following equation:

$$\Psi_{void} = \frac{\pi^{\frac{1}{3}}(6V_{void})^{2/3}}{SA_{void}} \quad (3.1)$$

Where  $V_{void}$  is the volume of the labeled void in  $\mu\text{m}^3$  as determined by counting voxels in the void and  $SA_{void}$  is the surface area in  $\mu\text{m}^2$ . The voids with  $\Psi_{void} < 0.3$  were found to describe cracks and other non-macro-void objects. Objects with  $\Psi_{void} < 0.1$  have been excluded from further analysis because the objects are either external voids or pore networks. The fraction of granule volume ( $\epsilon_{void}$ ) contained in macro-voids was also calculated after the exclusions.

A radial distribution of the void space from the granule center of mass was created to look for repeating trends in the locations of the void volumes using 50  $\mu\text{m}$  bins. An ellipsoid fitting function was used to look for preferential concentrations of voids at the top or bottom of granules using 50  $\mu\text{m}$  bins. The distributions were normalized to the size of each granule to account for variations in granule size. The normalization was done by defining a maximum voxel distance (size) for each granule using the particle and void bins with more than 50 voxels. This maximum voxel distance was used to create normalized bins which represent 5% of the total axial or radial distance. Typical processing times are partially dependent on the size of granule image stack. Drum granules from Chapter 5 take 4-5 hours to process an image stack with the current code structure. Powder C1-C3 granules take 3-4 days to process 10 granules.

## CHAPTER 4. STATIC BED GRANULATION

### 4.1 Introduction

The XRCT image analysis techniques described in Chapter 3 were developed and applied to the evaluation of the microstructure of single-droplet granules formed in a static bed. The developed techniques provide evaluation of macro-void structures formed from primary particles of various sizes using one of three powder bed preparation techniques. The presented results validate the usefulness of the developed techniques in describing granule macro-voids of varying shape and size.

### 4.2 Experimental Plan

Four different size fractions of alumina (0.5  $\mu\text{m}$ , 5  $\mu\text{m}$ , 25  $\mu\text{m}$  and 108  $\mu\text{m}$ ) were sifted into petri dishes through either a 1.4 mm, 710  $\mu\text{m}$ , or 500  $\mu\text{m}$  sieve and then the powder bed surfaces were leveled. Single droplet nuclei were created from the static beds, up to 300 at a time, and dried at 100 °C for 20 minutes. 10 granules were processed using the XRCT analysis techniques described in Chapter 3.

### 4.3 Results

#### 4.3.1 Powder Characteristics

The differences in size and shape between the powders can be observed using the SEM micrographs in Figure 4.1. Powder A in Figure 4.1A is a series of small particles which

naturally cluster together and are viewed at resolutions of 1  $\mu\text{m}$  or less. Powder B particles in Figure 4.1B are sintered clusters of smaller particles that resemble Powder A and are viewed at resolution of 5  $\mu\text{m}$ . Powder C in Figure 4.1C is composed of jagged, irregularly shaped particles that are significantly different from Powders A and B. Powder D particles in Figure 4.1D are actually spray-dried agglomerates that are composed of smaller primary particles.

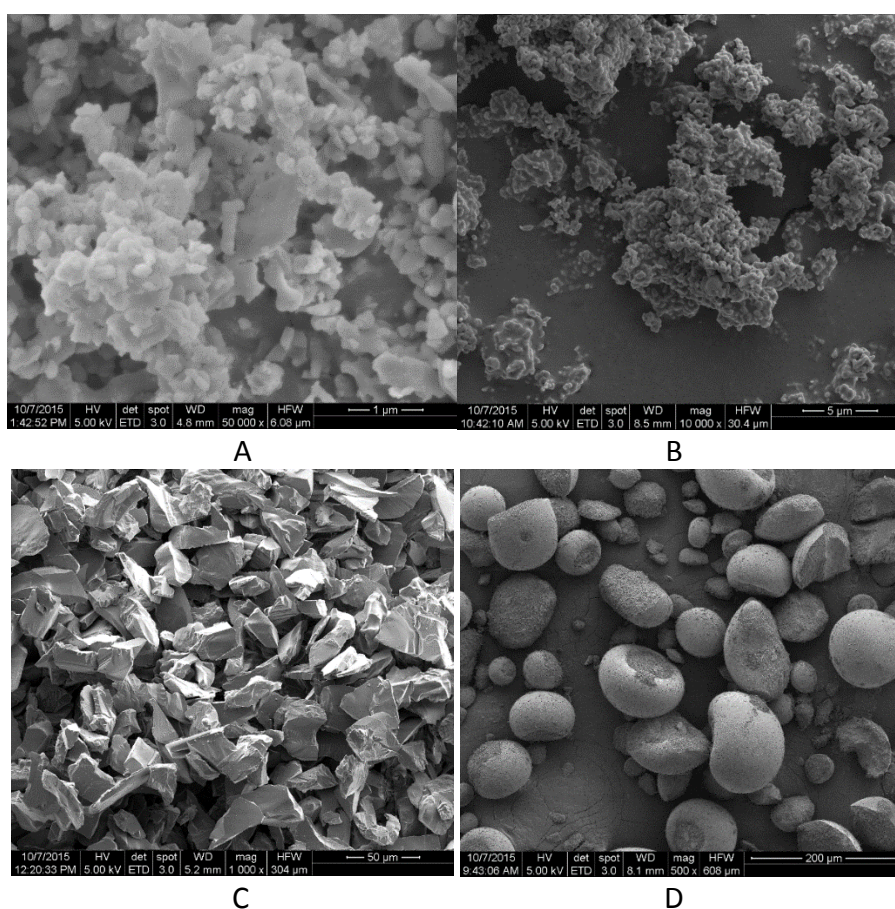


Figure 4.1: SEM Micrographs (A) Powder A (B) Powder B (C) Powder C (D) Powder D  
[Note: The scales are different in each figure as denoted by the scale bar in the bottom right of each figure.]

The light diffraction sizing data is shown in Figure 4.2 and generally confirms the estimates from observations made in Figure 4.1 regarding the size of primary particles.

The sieve data for 108  $\mu\text{m}$  particles in Figure 4.3 is also consistent with Figure 4.1D.

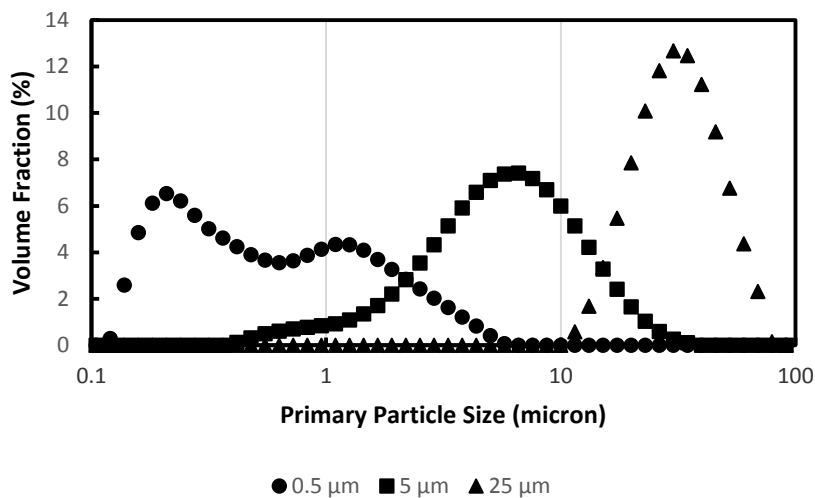


Figure 4.2: Particle size distribution of well dispersed alumina particles (Malvern Mastersizer 2000)

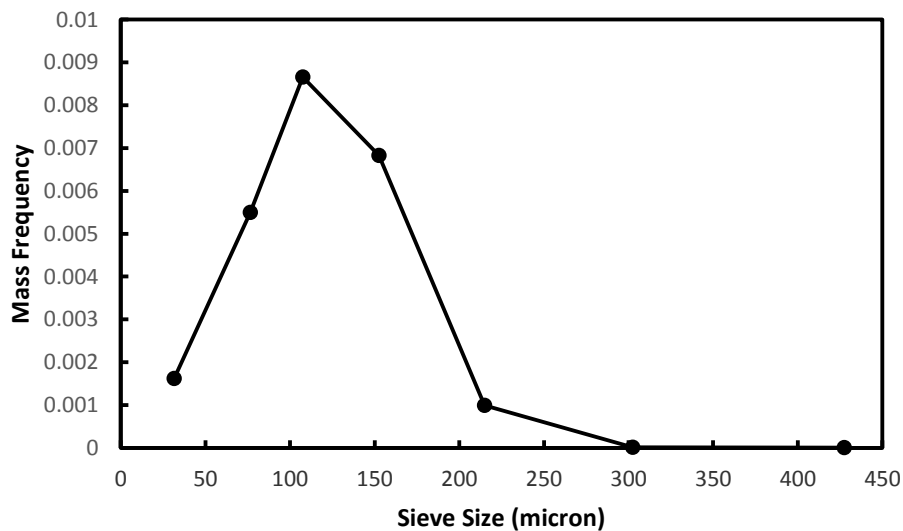


Figure 4.3: Particle size distribution of 108 micron aggregates (sieve size analysis)

The sizing data in Figure 4.2 shows that there are a large number of fine primary particles less than 1  $\mu\text{m}$  and 0.5  $\mu\text{m}$  in diameter for Powder A. The small particles were observed to aggregate over time when dispersed in water with ultrasonics, i.e. the size distribution shifted to the right with time. This is also consistent with Figure 4.1A where the primary particles naturally cluster. Typical  $d_{50}$  sizing for the 108  $\mu\text{m}$  agglomerate primary particles are between 0.8-1.0  $\mu\text{m}$ . The shear cell results in Figure 4.4 indicate that the 25  $\mu\text{m}$  and 108  $\mu\text{m}$  are exceptionally free flowing and the 0.5  $\mu\text{m}$  and 5  $\mu\text{m}$  particles are essentially non-flowing.

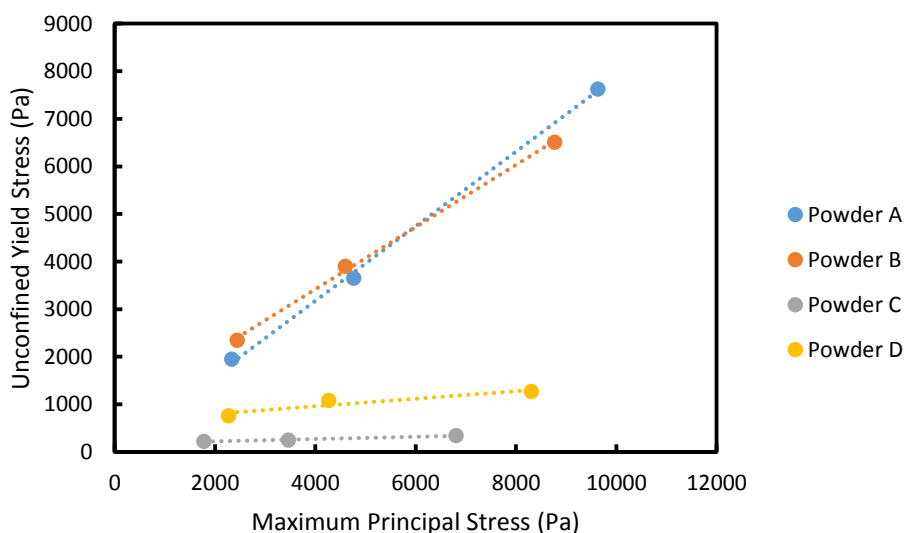


Figure 4.4: Ring Shear Tester results for alumina oxide powders

#### 4.3.2 Analysis of Granule Structure

The various granule structures can be evaluated visually using an optical microscope, visual examination of XRCT image stacks and/or 3-D reconstructions of the void spaces. Representative images of the different visualization options are shown in Figures 4.5 to

4.7. The different size fractions of alumina each produce a distinct granule shape and internal structure when viewed through optical microscopy and XRCT image stacks.

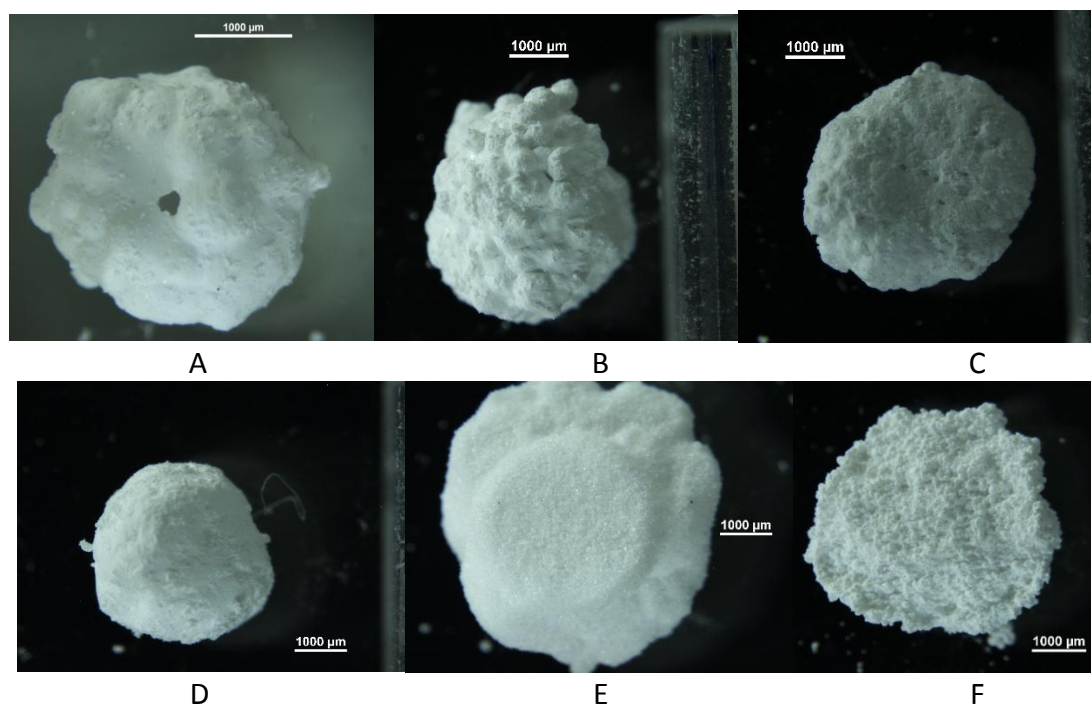


Figure 4.5: Microscope Images of alumina granules. (A) Powder A1 (B) Powder A2 (C) Powder A3

(D) Power B1 (E) Powder C1 (F) Powder D1

The optical microscopy images in Figure 4.5 shows the different alumina particle sizes produce different granule shapes. Powder A granules (Figure 4.5A to C) are made up of smaller spherical aggregates that appear to vary in size with sieve preparation. Powder A1 in Figure 4.5A has the largest aggregates and Powder A3 in Figure 4.5C has the smallest aggregates. The granules are round with a concave indentation at the top, relative to the orientation when formed. Powder B granules (Figure 4.5D) are rounded and smoothed with no visual differentiations from orientation or sieve preparation.

Powder A and Powder B granule shapes are consistent with the Tunneling mechanism as expected from primary particle size<sup>6</sup>. Powder C granules (Figure 4.5E) resemble a mushroom with a circular core at the top of the granule and a rounded cap at the bottom. Powder D granules (Figure 4.5F) are a thin relatively thin concave disk less than 2 mm thick. Powder C and Powder D granule shapes are consistent with the Spreading/Crater mechanism of drop granule formation<sup>6</sup>.

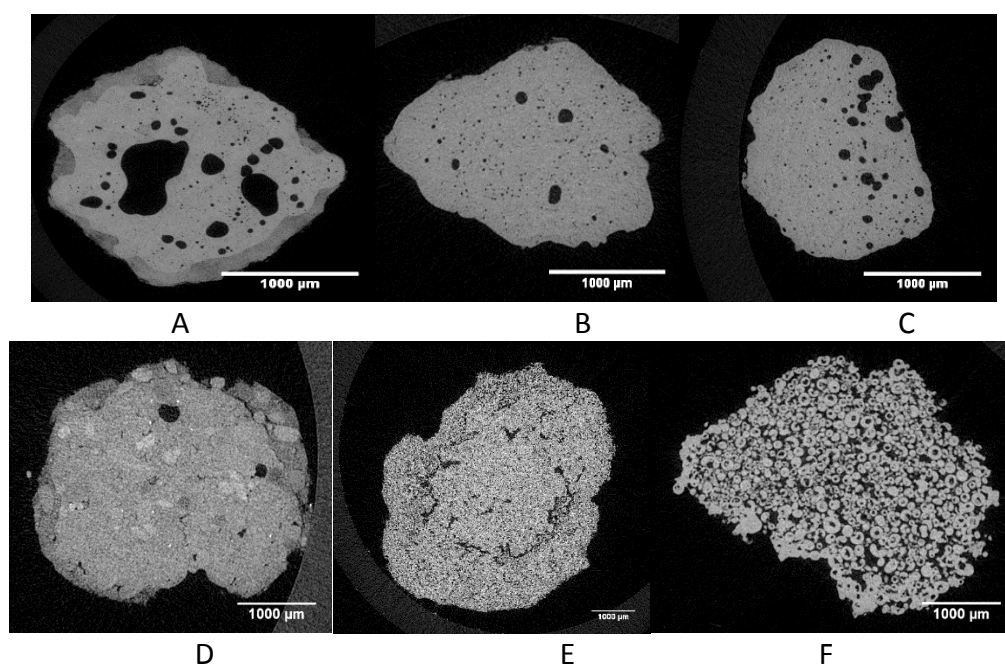


Figure 4.6: Representative XRCT Slices. (A) Powder A1 (B) Powder A2 (C) Powder A3 (D) Power B1 (E) Powder C1 (F) Powder D1 [Note: Granule contrast has been increased to improve object visibility in print]

The XRCT slices in Figure 4.6 show that there is also a significant effect from changing the primary particle size on the granule internal structure. Powder A granules in Figure 4.6A-C show a number of large, discrete void spaces surrounded by a dense particle matrix. Powder A primary particles are smaller than the 6  $\mu\text{m}$  voxel resolution so the



“solid” areas of the image are actually made up of a mixture of primary particles and pores. The voids in Figure 4.6A are noticeably larger than those in Figure 4.6B-C. The largest voids in Figure 4.6C appear to cluster to one side of the granule, which is the top of the granule when the formation orientation is considered. The Powder B granule microstructures in Figure 4.6D have a few large, discrete void spaces, but is mostly the particle matrix. Powder C granules in Figure 4.6E have some large cracks that are centered on the circular core observed in Figure 4.5E and a diffuse pore network elsewhere. Fig 4.6F shows that the Powder D particles are actually hollow spray-dried aggregates which is consistent with SEM images of the powder (Figure 4.1D). The internal structure is that of a pore network which is expected with large particles and a handful of isolated voids within the spray-dried agglomerates.

The use of 3D representations in Figure 4.7 allows for better visualization of the entire structure of a granule than looking at the 2D slices individually. Here, the colored objects are the macro-voids within the granule. Objects of identical color are connected in 3D. Powder A granules in Figure 4.7A-C can be fully visualized because the external void space is easily removed. Figure 4.7A shows the existence of large macro-voids that travel throughout the granules formed from Powder A1. The large orange void at the center and the light blue void in the lower right of Figure 4.7A are prime examples of this behavior with large necks formed between sections. The macro-voids are clearly smaller in Figure 4.7B & 4.7C but are still large distinct objects within the granules. This shows that preparing Powder A using different sieves creates differences in the

maximum size of macro-voids within the granules. Figure 4.7D & 4.7E show the results of external void fragmentation during processing. The large voids at the edges of Figure 4.7E have all been removed manually from the size/volume analysis as discussed in Chapter 3. Figure 4.7F shows that the external void fragmentation for Powder D granules the remaining macro-voids are thin slices at the outer edges and are not internal to the granule.

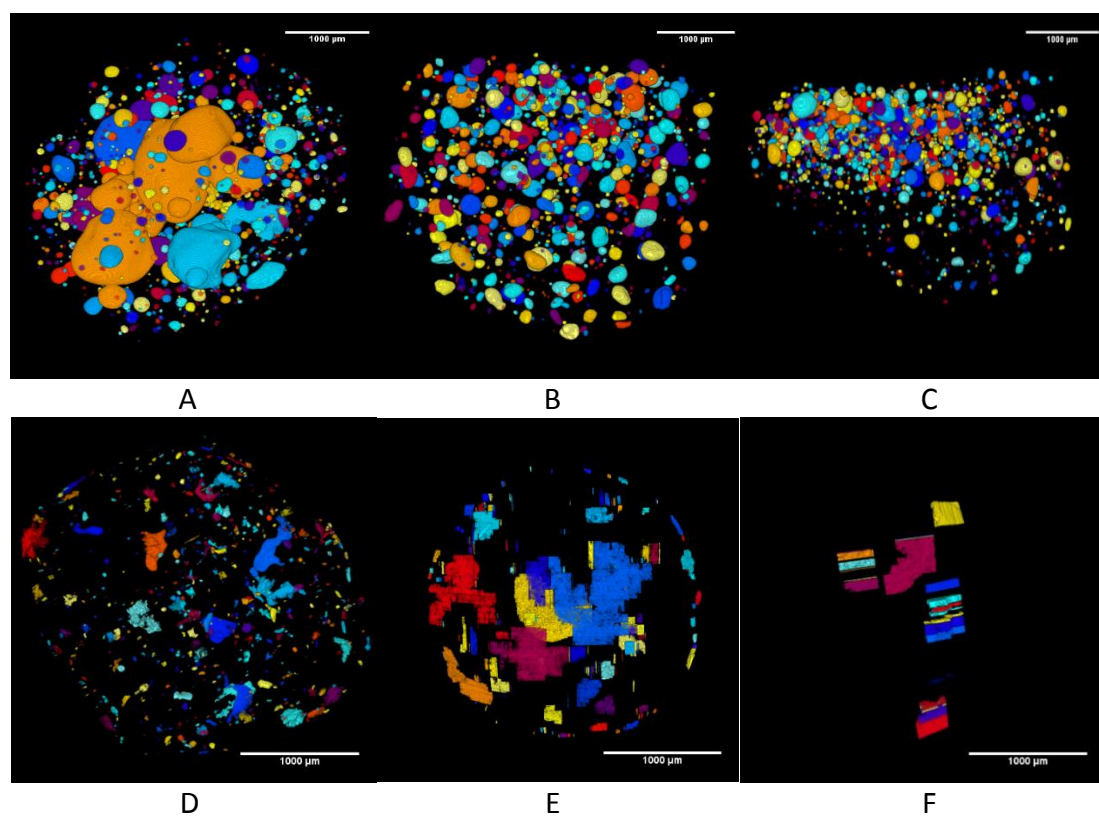


Figure 4.7: 3D macro-void reconstructions (A) Powder A1 (B) Powder A2 (C) Powder A3  
(D) Power B1 (E) Powder C3 (F) Powder D1

### 4.3.3 Effects of Primary Particle Size

Changing the primary particle size clearly shifts the nuclei formation mechanism.

Powder A (Fig 4.4A, B, C) and Powder B (Figure 4.4D) granules are formed through the Tunneling mechanism and are mostly round. The Powder C (Figure 4.4E) and Powder D (Figure 4.4F) granules are clearly formed through the Crater/Spreading mechanism. Powder C granules have the expected mushroom stalk/cape shape. Powder D granules are best described as thin, concave wafers formed only at the surface of the powder bed. The large particles can clearly be seen for Powder D granules and large agglomerates can also be seen in some of Powder A granules.

There are also clear qualitative effects on granule microstructure from the changes in particle size. Powder A granules (Fig 4.5A,B,C) show a solid matrix filled with many large, discrete macro-voids. The dense solid phase, which appears continuous with some variations in density, is actually a mixture of primary particles and pores. The primary particles are an order of magnitude smaller than the voxel resolution ( $6 \times 6 \times 6 \mu\text{m}$ ) so small spaces between particles are not visible. Powder B (Fig 4.5D) granules have a solid matrix with some large, discrete macro-voids but fewer and smaller visually than those in the  $0.5 \mu\text{m}$  granules. Powder C (Figure 4.5E) produces a discrete network of particles and voids with the spaces between particles visible. There are some large macro-voids that are best described as cracks in the Powder C and the rest of the voids are micro-voids. Powder D (Figure 4.5F) shows that it has a pore network made up of small spray-dried granules.

There is a strong effect from changing primary particle size on the percentage of the total granule volume contained in macro-voids or  $\epsilon_{\text{void}}$  (Figure 4.8). Powder A1 granules have average  $\epsilon_{\text{void}}$  values of 7% and as high as 9% while the Powder D granules have average  $\epsilon_{\text{void}}$  values of at or near 0. Decreasing the primary particle size causes the  $\epsilon_{\text{void}}$  value to increase both for ultra-fine powders and non-ultra-fine powders. The change in the granule nucleation mechanism from Powder B to Powder C results in an increase in  $\epsilon_{\text{void}}$  due to crack formation around the “stem” of the mushroom shaped granule.

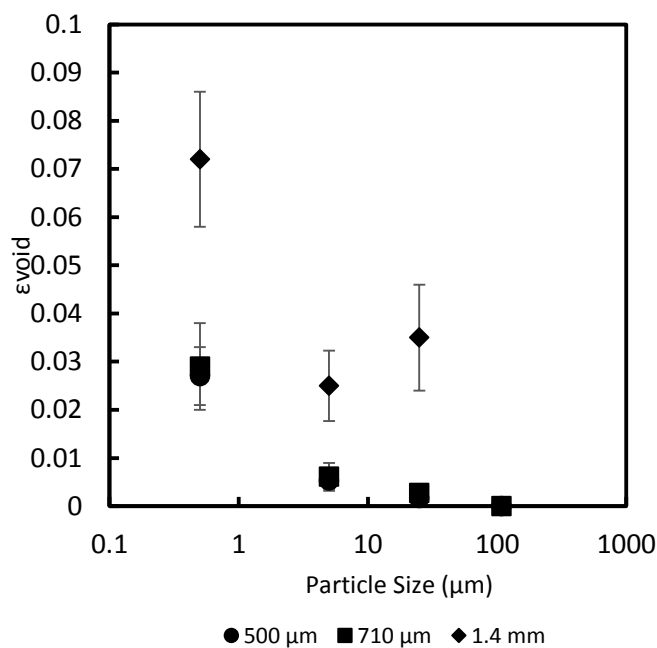


Figure 4.8: Fraction of total granule volume contained in macro-voids

A similar effect exists for the maximum macro-void size in each granule shown in Figure 4.9. A decrease in primary particle size strongly correlates with an increase in the maximum volume/size of macro-voids. The largest macro-voids in Powder B1 are two thirds the size of those in Powder A1 which have an equivalent sphere volume diameter

of 883  $\mu\text{m}$ . A typical Powder A or Powder B granule is approximately 3 mm in diameter. A typical powder C granule is approximately 4 mm across. There are no macro-voids in Powder D granules.

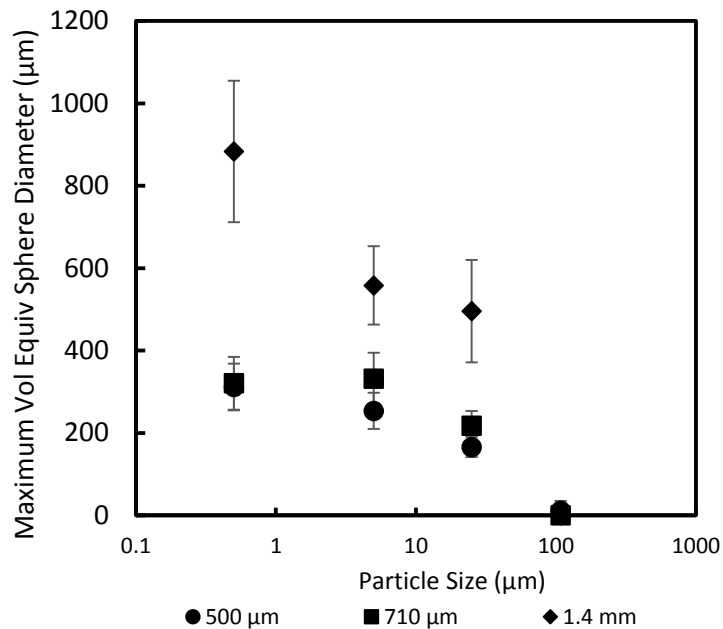


Figure 4.9: Average size of largest individual macro-voids

#### 4.3.4 Effects of Bed Preparation

There is a strong effect of bed preparation on the microstructure of Powder A granules. The effects on  $\epsilon_{\text{void}}$  and macro-void size can be seen in Figures 4.5 to 4.9 in both the visual observations and the quantitative measurements. Powder A1 granules have macro-voids which are visibly larger than the macro-voids in Powders A2 & A3 and this is confirmed by the measurements reported in Figure 4.9. There is a similar effect seen in the  $\epsilon_{\text{void}}$  values in Figure 4.8. The change in sieve size from Powder A2 to Powder A3 does not cause a similar change as seen in  $\epsilon_{\text{void}}$  and macro-void volume. The maximum

macro-void sizes in Powder A2 & A3 are 310 and 320  $\mu\text{m}$  respectively, although the shape of these voids is very different.

There is a weaker effect of bed preparation on the microstructure of Powder B granules than for Powder A. The data in Figure 4.8 and Figure 4.9 show that Powder B1 granules have more and larger macro-voids than Powders B2 & B3. There are no discernible effects of bed preparation on the volume or size of macro-voids in Powder C and Powder D granules. There are no changes to the nuclei formation mechanism from changes in bed preparation.

The observed effect of bed preparation on the distribution of macro-voids in Powder A granules must be evaluated in a different fashion. The 3D reconstruction of the Powder A3 Granule in Figure 4.7C shows that the macro-voids appear to cluster towards the top of the granule based upon the granule orientation during formation. The top of the granule is identified by the concave surface seen in Figure 4.4C. The axial distribution is the appropriate method for evaluating this. Figure 4.10 shows the axial distribution results for macro-void volume in Powders A1-A3 and it does show the clustering near the top of the granule for Powder A3. However, there is not a clear transition of macro-void position from Powder A1 to Powder A3. It is unclear if this is random variation as a result of small sample size or evidence of a second-order effect related to the bed preparation. However, the available information is sufficient to show that the analysis method is capable of picking up such a change in distribution if it should occur.

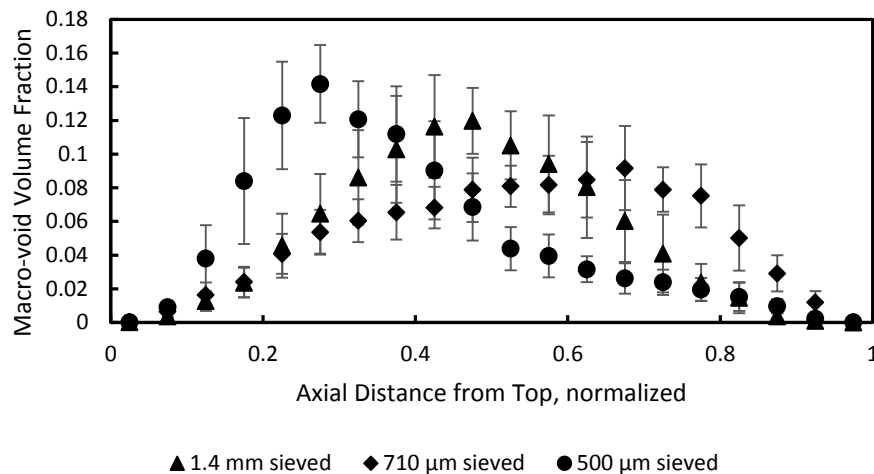


Figure 4.10: Axial distribution for 0.5  $\mu\text{m}$  alumina granules. Describes distribution of macro-void volume as a fraction of total macro-void volume.

#### 4.4 Discussion

The presence of strong cohesive forces creates the potential for complex structures in the powder bed. Large particles with little or no cohesive strength will form a relatively homogenous bed with consistent macro-, meso-, and micro-structures throughout the bed. The presence of strong cohesive forces creates the potential for complex structures of varying size within the powder bed. This leads to volumes of differing density and the creation of large scale, relatively stable structures. These different structures will interact with the liquid droplet at different rates and can break up if the larger structures have cracks or other flaws in them.

The quantifiable differences in structure come, in part, from the dominant attractive forces present in the 0.5  $\mu\text{m}$  particles. The images of the sieved 0.5  $\mu\text{m}$  particles in Figure 4.11 show that the powder forms small agglomerates of varying sizes. The different sizes of small aggregates likely contribute to differences in the packing of the

bed structure. The maximum size is greatest for the 1.4 mm sieving which also has a wider size distribution of the aggregates. The larger size distribution of agglomerates should result in a more varied bed structure. The larger structures are also more likely to break up because of flaws in the agglomerate. One possible explanation for the existence of macro-voids in the drop formed granules is that they already exist in the bed before liquid is added. Another is that the voids are primarily formed during drying in a fashion similar to that described by Pagnoux et. al.<sup>43</sup>. Certain high density, closely packed areas would have low particle mobility and lower density areas would have high particle mobility which leads to the existence of many macro-voids rather than a completely hollow structure. The former theory, that the structures existed to begin with, is more likely because the “drying” model would mean granule structure is independent of bed preparation. The rate at which the granules are dried may also have some second-order effects on the precise macro-void size.

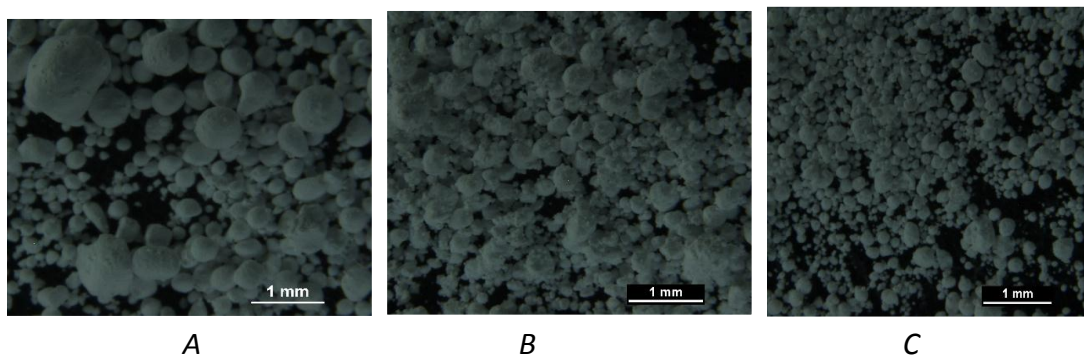


Figure 4.11: 0.5  $\mu\text{m}$  primary particles that have been passed through different sieve sizes. (A) A1 (B) A2 (C) A3



The more cohesive the material is, the greater the potential for complex structures to exist in the bed. The more complex structures that exist in the powder bed, the larger the number and/or size of macro-voids within a granule. This would explain why Powder A granules have more and larger macro-voids than Powder B granules after sieving. The switch to larger, non-cohesive reduces the opportunity for complex structures.

The study and description of granule microstructure in literature is sparse, especially for small particles. However, there are a variety of different microstructures that exist dependent up on the particle size. These different structures need to be characterized in a quantitative manner to describe the effects of process changes. There is the potential for very large macro-voids to exist within granules separate from the hollow structures that can be found in certain processes. The methods described in this paper show how to quantify changes in these macro-voids based on size, shape, and location within the granule. It also allows for identification and differentiation between external void space, pore networks, cracks, and macro-voids. The powder handling history has also been shown to have an impact on the structure of the smallest materials. The handling history has little observed effect for the larger powders. These observations explain the anecdotal evidence from industrial application that granulation of ultra-fine powders can be highly variable and highly sensitive to powder history.

The impacts of these different microstructures and handling histories on the further processing of granules is unstudied. However, granules that are hollow or have large macro-voids can be considered to have flaws of various sizes. These flaws introduce

potential weaknesses into the granule structure and can affect granule properties, such as strength, and make the granules more likely to break than their homogenous counterparts. This is the theory promoted by Kendall which argues that fracture strength scales with the negative square root of the flaw size<sup>78</sup>. Thus, the strength of granules formed from ultra-fine powders can be much lower than expected unless care is taken to remove macro-voids.

It is also possible that additional, vigorous processing of the powders will override the previous history. The exact method of handling is likely to be important as a fluidized bed provides very different stresses on the granules than a screw feeder or vibrating tray and all are different from a static, sieved bed. A fluidized bed arrangement is the one most likely to show effects similar to those from bed preparation shown in this paper<sup>44-51</sup>. Powder mixtures including particles that differ in size by orders of magnitude are not expected to show effects from handling history because the larger particles are expected to be dominant.

#### 4.5 Conclusions

This work successfully demonstrates a method for analyzing 3D granular meso-structures using XRCT. This method has been shown to be capable of distinguishing between pore networks, cracks and large, discrete macro-voids that exist in granules formed from various sized primary particles that are chemically similar. It makes use of macro-void sphericity and volume as primary descriptors to distinguish between different types of structures. This method can be used to distinguish between

differences brought on by changes in the production process. It can be used, for example, to supplement porosity measurements by giving a precise explanation of how structure has changed in the granule.

Ultra-fine powders have complex behaviors and the effect becomes stronger as the particle size is decreased. Chemically similar materials can be made to form radically different granule structures simply by changing the size of the primary particles. Sub-micron primary particles are capable of forming granules with large, discrete macrovoids of various shapes and sizes. Therefore, the size distribution of feed materials must be tightly controlled to ensure consistent and reproducible behaviors in granulated materials.

The prior handling of ultra-fine powders has also been shown to have an impact on material behavior and the granule microstructure. Due to natural agglomeration of fine particles, this effect is strongest for the smallest particles. Other systems which can induce this behavior, such as fluidized beds, have the potential for similar issues. It also shows the need for greater care with the handling and preparation histories of ultra-fine powders because it can profoundly affect final granule properties.

## CHAPTER 5. DRUM GRANULATION

### 5.1 Introduction

Chapter 3 and Chapter 4 showed the development of an XRCT based granule microstructure analysis technique using static bed granules. This technique has proven useful in identifying and measuring the existence of large macro-voids as well as size, shape, and macro-void volume fraction. The analysis method was also successful in measuring the effects of powder history and powder bed preparation on granule microstructure. In Chapter 5, this analysis technique is applied to granules formed in a tumbling drum. This study addresses the ability of XRCT analysis to evaluate the industrially relevant question of the effects of consolidation with and changes in liquid viscosity on granule microstructure. The created granules are compared to the predictions of the surface-tension-driven flow model of the nucleation immersion mechanism.

### 5.2 Theory

The surface-tension-driven flow model of the nucleation immersion mechanism, discussed in Section 2.2, predicts the creation of initially hollow granules wherein the hollow center shrinks with time until the granule core is solid. Ideal nucleation immersion mechanism granules are referred to as Hounslow Granules in this study.

Equation 2.15 predicts the time at which the void should disappear and was used to determine reasonable experimental run times. The parameter values used are found in Table 5.1 and model predictions are found in Table 5.2. Additional time estimates at one and two orders of magnitude larger are also included because of the conclusions from experimental testing of the nucleation immersion mechanism by Pitt et. al<sup>26</sup>.

Table 5.1: Surface-Tension Model Parameters and Predictions

Model Parameter Values			
	10% PVP K32	5% PVP K90	7% PVP K90
Droplet Size (mm)	2.83 ± 0.047	2.87 ± 0.043	2.91 ± 0.03
$\phi_{cp}$	0.5	0.5	0.5
$\gamma$ (mN/m)	57	62	68
$\mu$ (Pa*s)	0.07	0.036	0.0055

Nucleation Immersion Surface-Tension Model Predictions

	Model Predictions (s)	Model*10 (min)	Model*100 (min)
Powder A 5.5 cP	47.8	7.97	79.7
Powder A 35 cP	288	47	479
Powder A 70 cP	540	89.9	899
Powder C 5.5 cP	0.96	0.16	1.6
Powder C 35 cP	5.6	0.96	9.6
Powder C 70 cP	10.8	1.8	18

### 5.3 Experimental Plan

Granules were formed in the tumbling drum according to the procedure outline in Chapter 3.3 using four different experimental time points. The original model predictions in Table 5.2 would suggest experimental run times of less than 1 minute for Powder A and 1 second for Powder C. Powder A experimental times were selected as three seconds and five, ten, and fifteen minutes. The selected run times place Powder A

experimental times between one and two orders of magnitude greater than original predictions for the low viscosity binder. Powder C run times were selected as three seconds, ten seconds, one minute, and five minutes which places times between one and two orders of magnitude. Fifteen single-droplet granules were created at each time point and powder/binder combination and ten granules were used for XRCT analysis. Binder liquid usage is referenced according to liquid viscosity, Powder A 5.5 cP, etc.

## 5.4 Results & Discussion

### 5.4.1 Analysis of Granule Structure

The various granule structures can be evaluated visually using an optical microscope, visual examination of XRCT image stacks and/or 3-D reconstructions of the void spaces. Representative images of the different visualization options are shown in Figures 5.1 and 5.3. Diagrams of the observed internal structures are shown in Figure 5.2. The different size fractions of alumina each produce a distinct shape and internal structure when viewed through optical microscopy and XRCT image stacks.

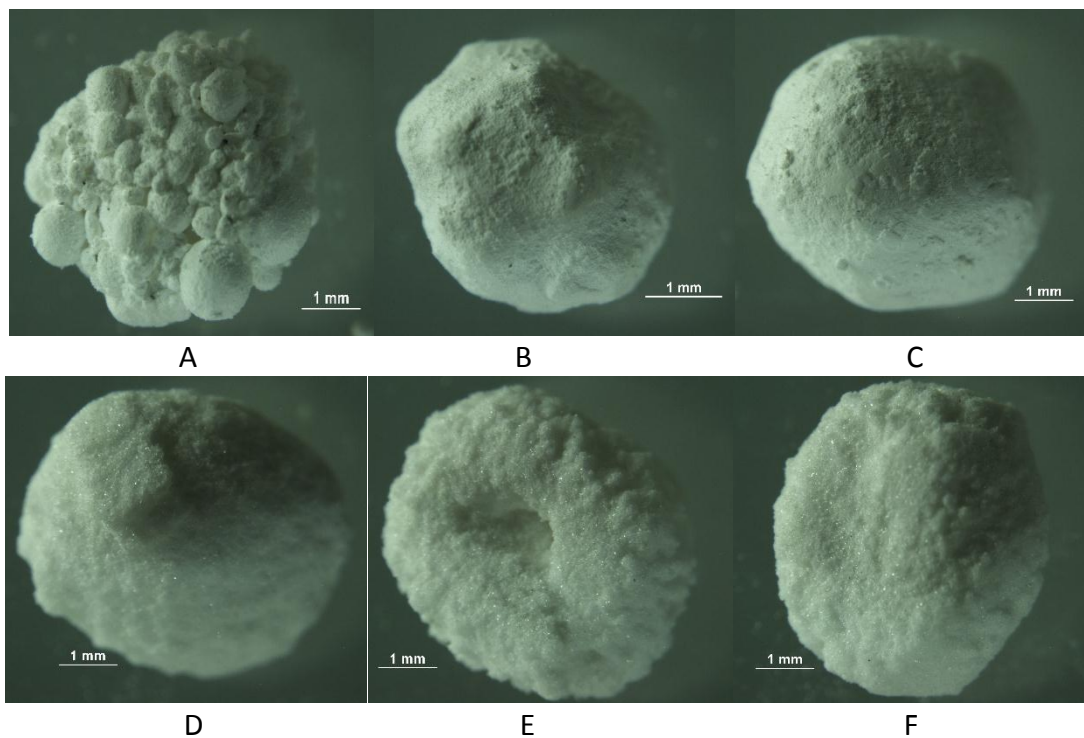


Figure 5.1: Microscope Images of Alumina Granules. (A): Powder A5 3 sec (B): Powder A5 5 min (C): Powder A5 15 min (D): Powder C5 3 sec (E): Powder C5 1 min (F): Powder C5 5 min

The optical microscopy images in Figure 5.1 shows that Powder A and Powder C can produce granules with similarly rounded shapes after 5 minutes. However, there are clear differences between the two types of granules in terms of surface morphology. Powder A granules are clearly formed from a combination of large, stable agglomerates (Figure 5.1A) after 3 seconds with a powder shell growing to cover the agglomerates at 5 minutes (Figure 5.1B). The shell remains intact and the granule surface is smoother after fifteen minutes (Figure 5.1C) This basic structural format is consistent for Powder A granules irrespective of binder viscosity. Powder C granules have a smoother surface and can be either the round shape seen in Figure 5.1D or the flatter, pancake shaped granule seen in Figure 5.1F. Some of the Powder C granules also have holes or

indentations in the surface like the hole in Figure 5.1E. The flat and rounded shapes are both present in all sets of Powder C granules.

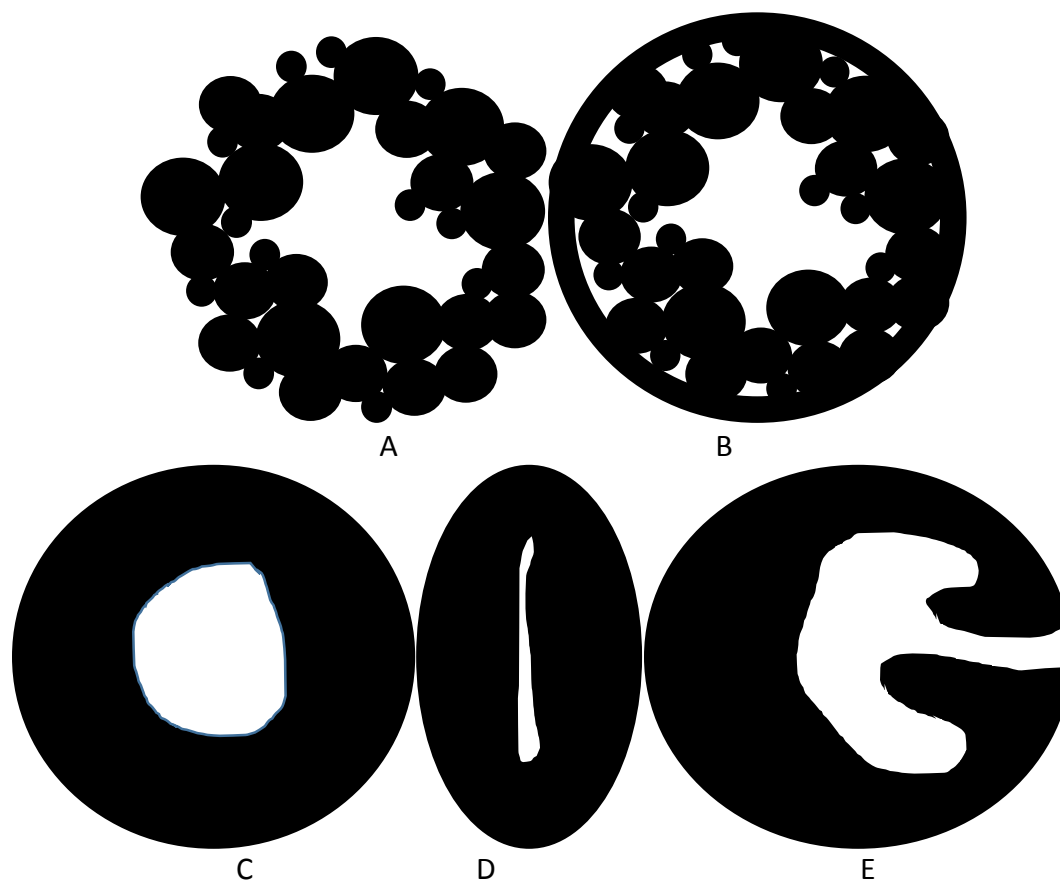


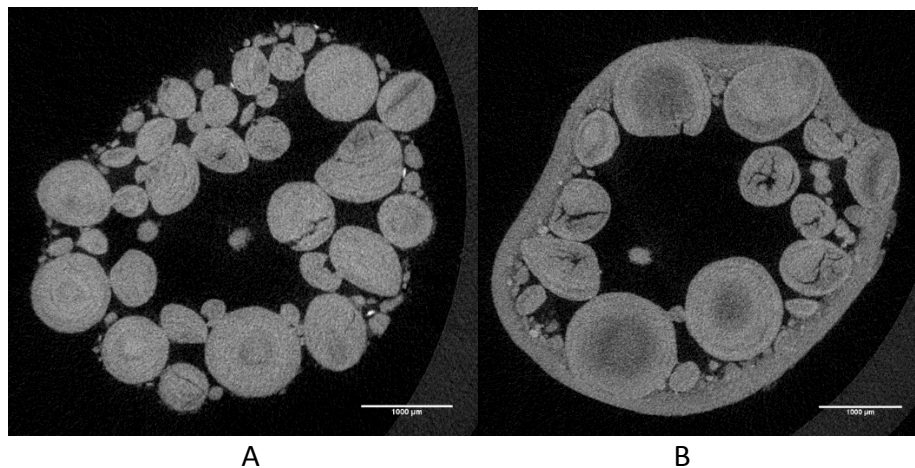
Figure 5.2: Representative diagrams of granule microstructures. (A): Powder A Nuclei (B): Powder A with powder shell (C): Powder C with large enclosed macro-void (D): Powder C with squished macro-void (E): Powder C with internal void connecting to granule exterior

The diagrams in Figure 5.2 are idealized forms of the observed internal granule structures. Powder A granules show a hollow structure surrounded by a clustering of large agglomerates either with or without a powder shell (see Figure 5.2A & Figure 5.2B). Powder C granules presents one of three idealized structures. Powder C granule void spaces can be roughly spherical (Figure 5.2C) or the spherical hollow can appear to have been compressed into a narrow opening (Figure 5.2D) which typically corresponds



to a flatter, pancake shaped granule. Granules can also be somewhere inbetween the idealized diagrams of Figure 5.2C & D. A third structure also occurs in which the internal void space connects via a large channel to the surface of granule (see Figure 6.2E).

Representative XRCT slices of different Powder A and Powder C granules in Figure 5.3 show what these idealized structures look like inside of real granules. The Powder A granules are formed from agglomerates of a higher comparative density than the powder shell. This is concluded from the higher intensity seen in the agglomerates in Figure 5.3A&B (white is an intensity value of 255 and black is intensity value of zero). The stable agglomerates and the shell are both powder matrices that mix primary particles and the spaces between primary particles. Powder C granules are formed of primary particles with a primary hole near the center of the granule. The outer layer is a high density area with majority of void space being pore space between primary particles and a few voids that might be macro-voids near the surface.



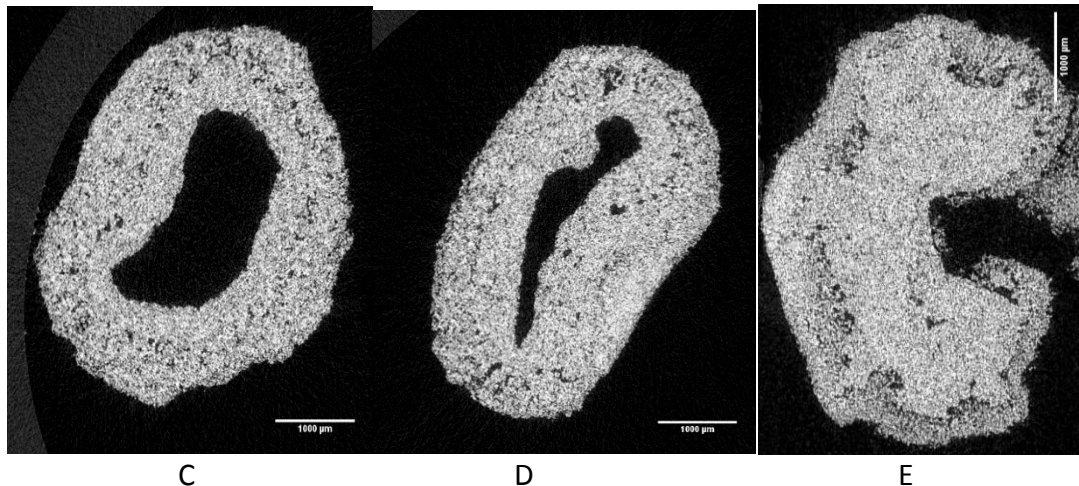


Figure 5.3: XRCT Slices of Powder A & C Granules. (A): Powder A5 3 seconds (B): Powder A5 5 minutes (C): Powder C6 1 minute (D) Powder C6 1 minute (E) Powder C4 1 minute

#### 5.4.2 Granule Consolidation with Time

Figure 5.4 shows the change in macro-void volume fraction,  $\epsilon_{\text{void}}$ , with time. The plot of  $\epsilon_{\text{void}}$  shows a clear effect for between the first and second time points for Powder A granules. This is the creation of the powder shell shown in Figure 5.1 A&B. The powder shell remains at ten and 15 minutes, but the bumpy surface texture caused by the stable agglomerates remains visible. The plot of macro-void volume fraction in Figure 5.5 clearly shows this outcome as  $\epsilon_{\text{void}}$  drops approximately 40% from a value of 0.26-0.31 at 3 seconds to 0.16-0.19 at five minutes. The decrease in  $\epsilon_{\text{void}}$  continues after five minutes but is noticeably slower than in the first 5 minutes. Variation in results between individual granules The 95% confidence intervals on the mean value for the Powder A results are small for all data points and data sets for different binders do not overlap.

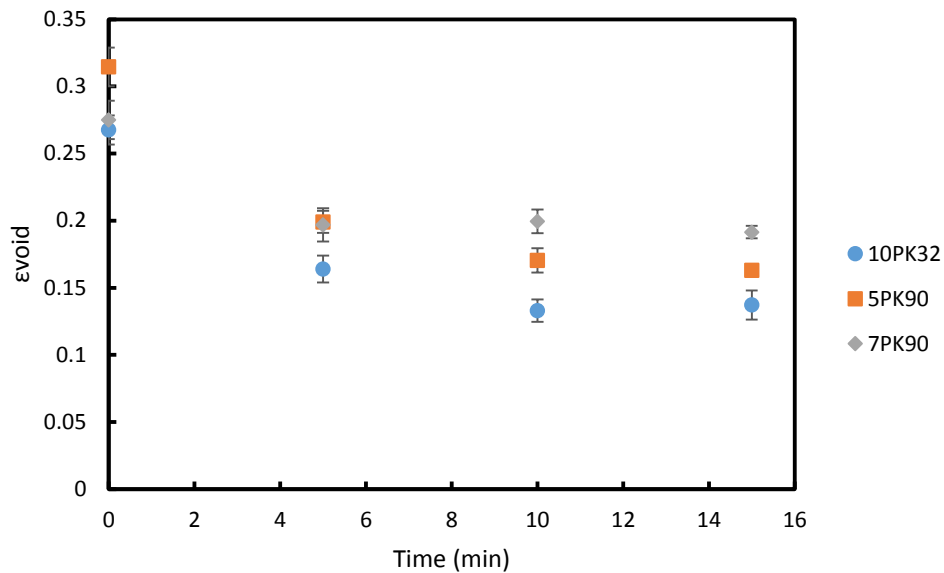


Figure 5.4 Macro-void Volume Fraction of Powder A granules

There are no clear changes in  $\epsilon_{\text{void}}$  as a function of consolidation time for the Powder C granules. The shapes of the granules, described in Section 5.3, do not. The results in Figure 5.5 shows that  $\epsilon_{\text{void}}$  clusters between 0.07 and 0.13 for all time points with significant overlap of the 95% confidence intervals. There are no consistent trends indicating a change in  $\epsilon_{\text{void}}$  with time. The 95% CI on the mean are significantly larger for Powder C granules

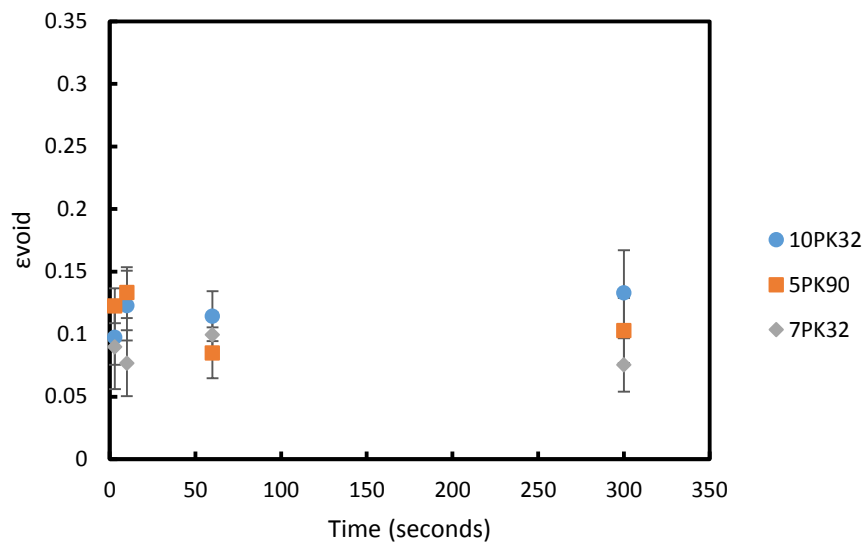


Figure 5.5: Macro-Void Volume Fraction of Powder C Granules

The results shown in Figure 5.5 and Figure 5.6 are consistent with the measurement of the size of the largest macro-void shown in Figure 5.7 and Figure 5.8. These results do show that there is a single void of exceptional size in each granule, with Powder A granules having macro-voids between 2 and 2.5 mm in equivalent sphere volume diameter while Powder C granules are between 1.5 and 2 mm in diameter. Powder A and Powder C granules have average diameters between 3.9 and 4.4 mm. Most of the macro-void volume is contained in these single, large macro-voids.

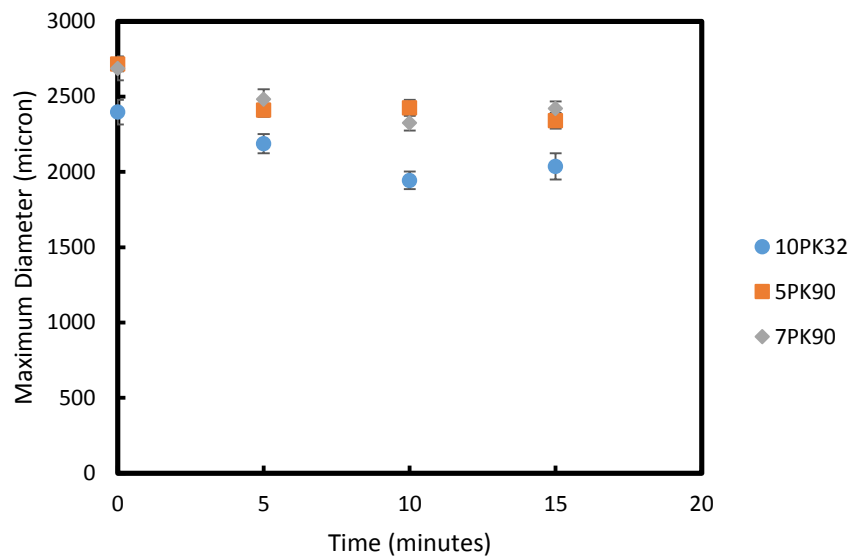


Figure 5.6: Diameter of Largest Macro-Void in Powder A granules expressed as Equivalent Sphere Volume Diameter

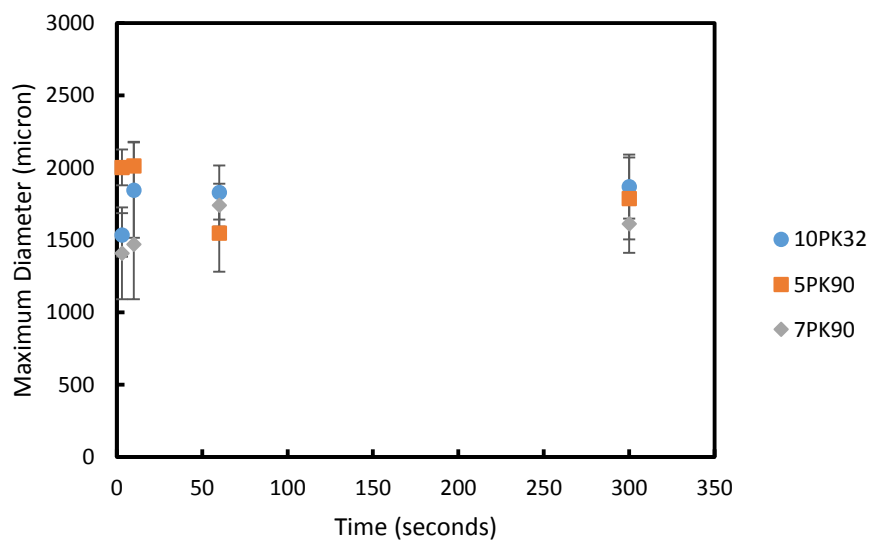


Figure 5.7: Diameter of Largest Macro-Void in Powder C granules expressed as Equivalent Sphere Volume Diameter

The radial distribution of macro-void volume fraction for Powder A 5.5 cP in Figure 5.9 also shows the change that results from addition of the powder shell. The macro-void volume fraction is the fraction of granule volume in a given radial bin which is a macro-void. The radial distribution after 3 seconds shows an increase in macro-void volume

fraction about halfway out from the radial center of mass (COM) while the other time points converge and continue a smooth decline to the outer granule edge. The bins at the center and the edges (0.0-0.2 and 0.8-1.0) have large 95% CI when compared to the middle range of radial distances. The large CI values confuse the analysis of shifts in macro-void volume as a function of time at the center of mass. Powder C 5.5 cP granules have high variability in the first third of the radial distribution which confounds evaluating any change as a function of time. The 95% CI values near the center are between  $\pm 0.25$  to  $\pm 0.37$ , covering the entire available range between the different time points. The time based distributions converge at a normalized distance of 0.4 and remain so to the edge of the granule. The Powder C 5.5 cP granule radial distributions of macro-void fraction do show an increase in overall macro-void volume at the edges of the granule.

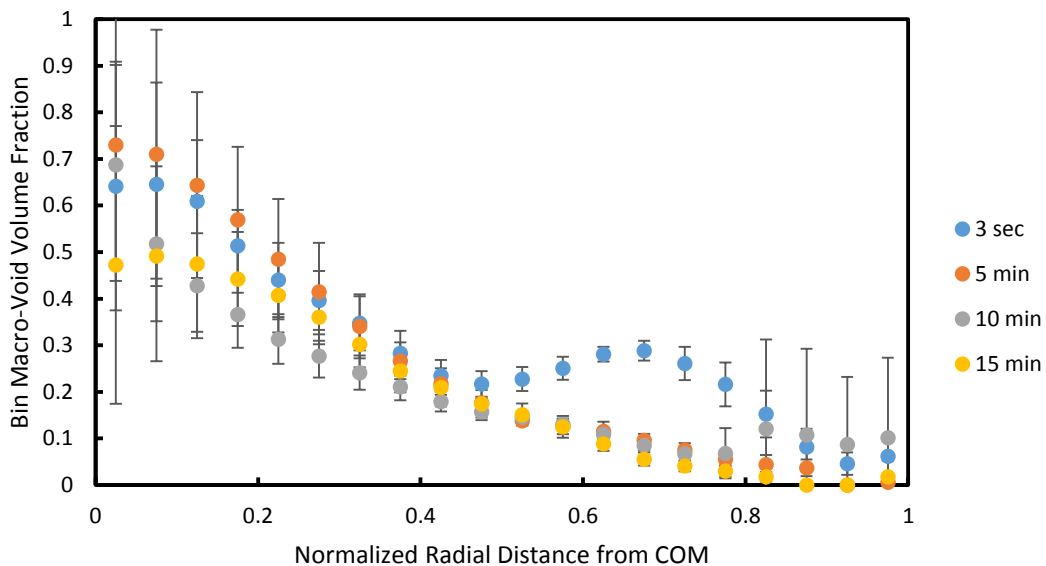


Figure 5.8 Powder A 5.5 cP Radial Analysis of macro-void volume fraction

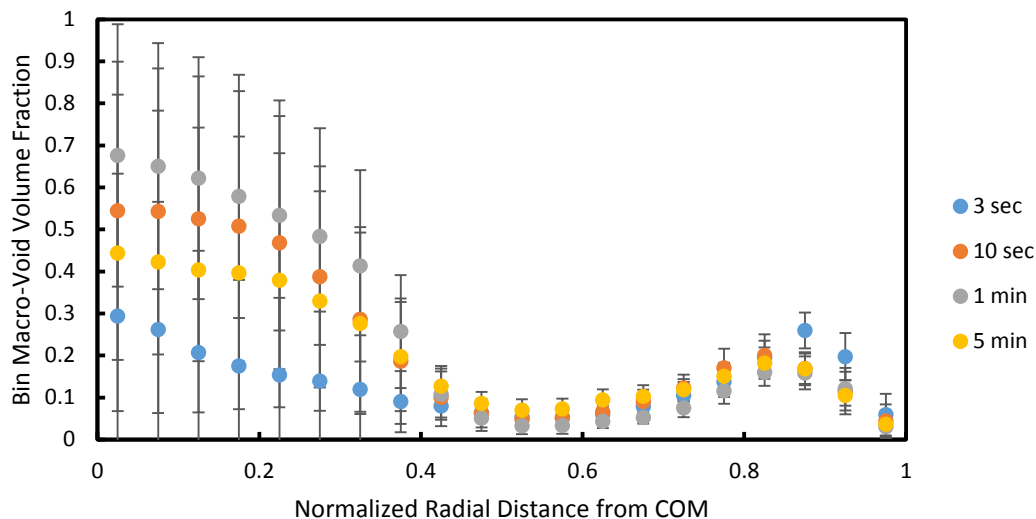


Figure 5.9: Powder C 5.5 cP Radial Analysis of macro-void volume fraction

#### 5.4.3 Effect of Binder Viscosity

The effect of binder viscosity on Powder A macro-void shape and size is shown in Figure 5.5 and 5.7. The increase in viscosity causes  $\epsilon_{\text{void}}$  to increase from a value of 0.13 to 0.19 for Powder A 5.5 cP and Powder A 70 cP respectively at the 15 minute time point. The increase in viscosity results in  $\epsilon_{\text{void}}$  being higher at all time points after formation of the powder shell. Increasing binder viscosity leads to granules with a larger volume fraction of macro-voids. In contrast, within experimental error there is no effect of binder viscosity on  $\epsilon_{\text{void}}$  or diameter for Powder C granules.

The radial distributions of macro-void volume fraction for Powder A 35 cP and Powder A 70 cP granules in Figure 5.11 and 5.12 have the same shape as that of Powder A 5.5 cP in Figure 5.9. Increasing binder viscosity results in a slower rate of decrease in macro-void volume fraction as radial distance increases which is not accounted for by the 95% CI and is predicted in part by the  $\epsilon_{\text{void}}$  values in Figure 5.5. There is also an apparent

increase in macro-void volume fraction near the center of the granule, but large CI values make it unclear if it is a true increase. The radial distributions for Powder C 35 cP and Powder C 70 cP granules in Figure 5.13 and 5.14 seem to show an overall increase in macro-void volume fraction near the center of the granule and a smaller amount of macro-void volume at the edge of the granule compared to the Powder C results in Figure 5.10. The large CI values for bins at the granule COM overlap strongly with the Powder C 5.5 cP granule results in Figure 5.10. The values at the granule edge do not overlap with those in Figure 5.10 and indicate that increasing viscosity causes a denser granule surface.

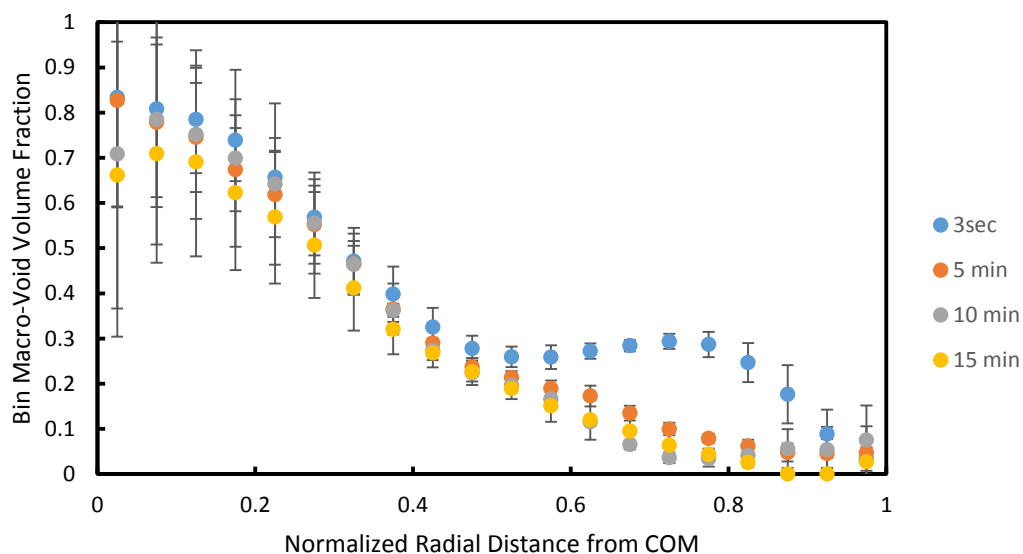


Figure 5.10: Powder A 35 cP Radial Analysis of macro-void volume fraction



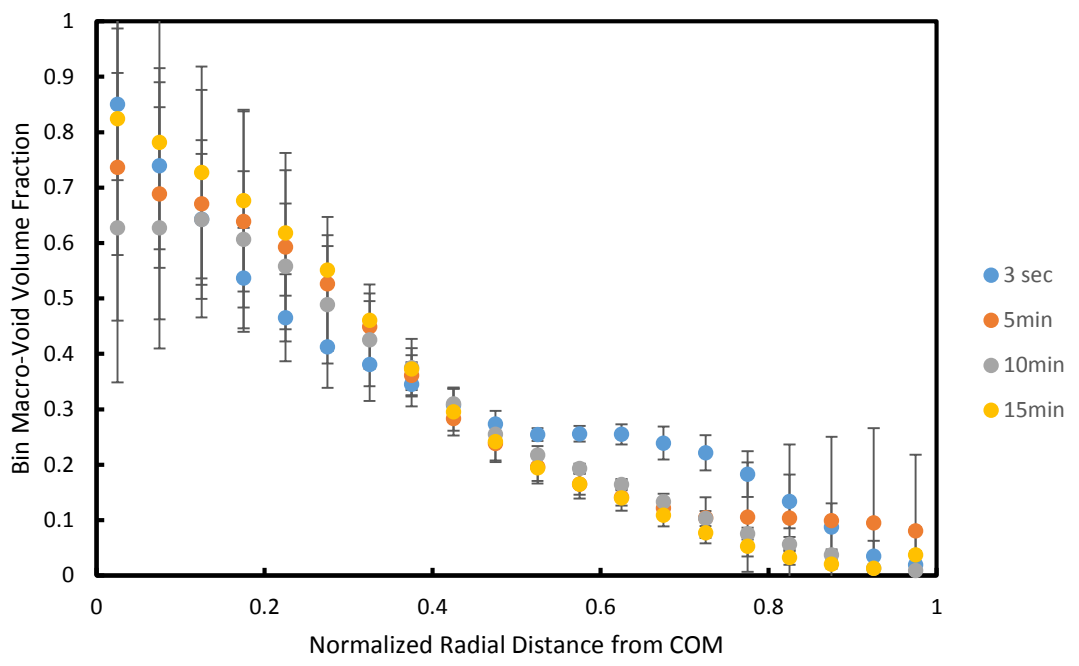


Figure 5.11: Powder A 70 cP Radial Analysis of macro-void volume fraction

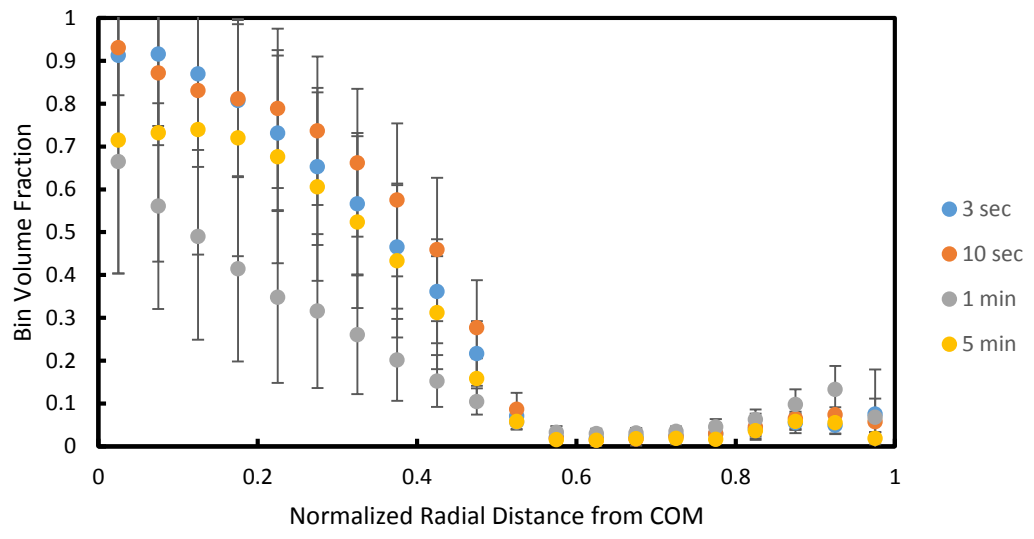


Figure 5.12: Powder C 35 cP Radial Analysis of macro-void volume fraction

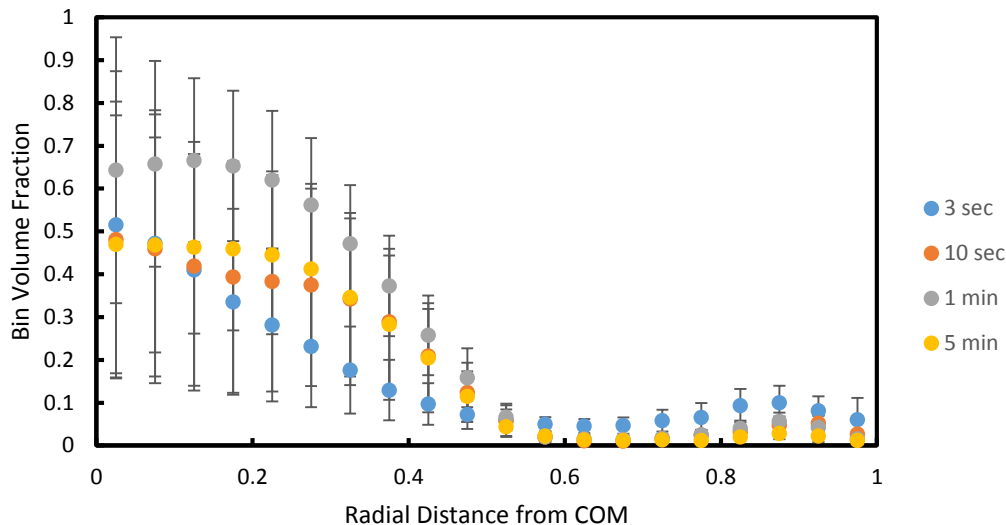


Figure 5.13: Powder C 70 cP Radial Analysis of macro-void volume fraction

#### 5.4.4 Discussion

The Powder A drum granulation process results in a bed of powder which creates good powder circulation. The addition of PVP binder to the Powder A granules has resulted in the large, stable agglomerates behaving as larger, single particles within the structure of the granule. The three second granules show that solid bridges form rapidly between the large agglomerates and that the granules have already attained a basic hollow structure seen in Figure 5.1A. The growth of the powder shell covers the open spaces and fills in surface spaces between the agglomerates but the internal agglomerates do not show evidence of significant rearrangement or consolidation with time. The broken Powder A 35 cP granule at fifteen minutes in Figure 5.15 shows that the shell is coating the stable agglomerate structure. The observed drop in macro-void volume fraction and size from three seconds to five minutes is primarily caused by the creation of the powder shell. The convex hull wrapping procedure for the three second granules

encompasses volume that is between agglomerates at the outer edges of the granule which is then filled in when the powder shell forms. A majority of the volume change can then be attributed to a decrease in the amount of external void volume counted by the wrapping procedure. Figure 5.16 contains slices of labeled macro-void volume for three second (Figure 5.16A) and five minute (Figure 5.16B) granules to show the change in wrapping from powder shell formation. It is unclear if the powder shell continues to grow with time or reaches a stable size.



Figure 5.14: Powder A5, 15 minutes, broken shell showing hollow agglomerate structure

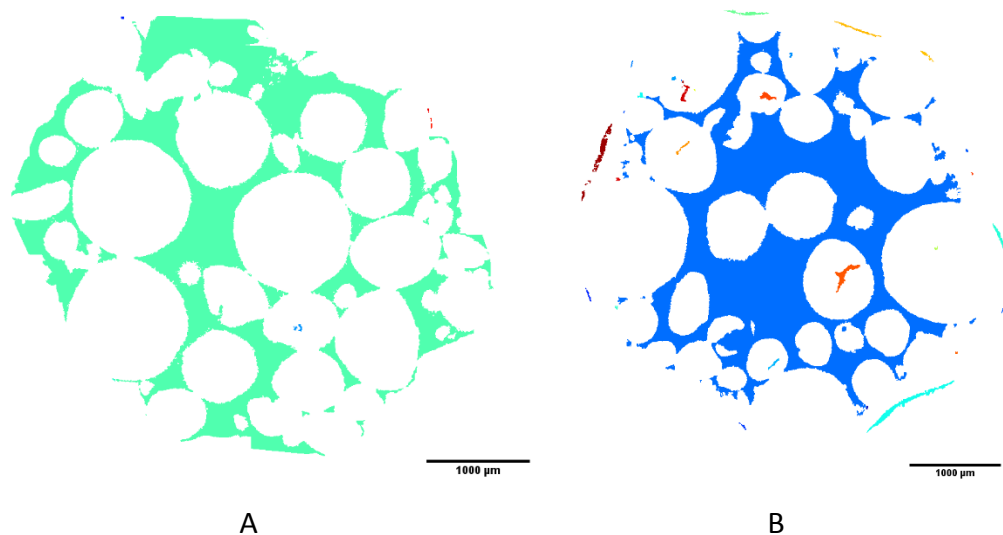


Figure 5.15: Labeled Connected Macro-Void Volumes for Powder A6 granules (A) 3 seconds (B) 5 minutes

The created granules do not match the ideal Hounslow granule because the predicted time scales are inaccurate for the complex system created by the ultra-fine powders. The droplet should have a rapid penetration time into the bed of large stable agglomerates and quickly form a granule nuclei, which does occur as seen with the three second granules. Liquid does go to the surface and take up excess powder to form the powder shell which is predicted for a Hounslow Granule. However, there is a second time-scale that is not considered for the Hounslow Granule, the penetration time of liquid into the large agglomerates made up of primary particles. This penetration time is 2000 times slower into the agglomerates using Equation 2.1 & 2.3 if it assumed that all other terms are constant and the particle sizes are  $0.5 \mu\text{m}$  and  $1 \text{mm}$ . Some of the liquid is then available at the granule surface for building the powder shell and the rest is consumed by infiltrating the powder agglomerates. The existence of multiple time scales is something that is not considered in existing nucleation and growth models. The creation of an interconnected network of large agglomerates makes rearrangement and consolidation unlikely through either surface tension forces or diffusion from the relatively low energy impacts within the drum granulator. The large void at the center of the granule remains largely intact throughout the granulation because of the interconnected network. An similar application of the microstructure analysis for breakage performed by Dale et. al with glass ballotini could be used to create an estimate of the forces need to cause consolidation by looking at the network of particle contacts<sup>70</sup>.

The radial distributions show clearly the effects of the powder shell with time, but the changes in liquid viscosity result in only small shifts in the distribution. The overlap of the 95% CI values near the granule center mean that changes in the radial distribution could easily be the result of experimental variability. The high variability is caused by variations in the exact location of the COM and arrangement of the agglomerates. The small bin size at the center means that if the COM bin contains part of an agglomerate there will be a disproportionate decrease in  $\epsilon_{\text{void}}$  for those bins. The small 95% CI values for the macro-void volume and size suggest that experimental variability and small sample size are not the cause of the changes for Powder A granules but are instead representative of real effects. The large 95% CI values at the edges of the radial distribution are the result of the external void artifacts resulting from the convex hull wrapping.

Powder C does not circulate well in the tumbling drum. The powder instead rides up the wall, slumps back towards the bottom, and repeats the process. The Powder C granules appear to achieve a stable condition within the first few seconds of droplet addition to the bed which aligns with the original and one order of magnitude model predictions. The granules, when retrieved, are still wet and can be squeezed into a flatter shape. The likely cause of flat granules is early contact of the forming granule with either the drum wall or one of the bars attached to the drum wall. The question of whether or not a droplet/granule will impact the wall is likely related to the position and direction of travel of the powder bed when the droplet reaches it. No granule was

retrieved in a handful of droplet releases because a mass of hardened powder was found attached to the wall after the drum was emptied.

The large 95% CI values that exist in the volume fraction, macro-void size, and radial distributions indicate that experimental variability is the primary cause of changes in measured parameters. The existence of 3 different granule structures while only testing 10 granules for each experiment cloud evaluating any meaningful effects that might occur. It may also be that no changes occur after the early granule formation and powder uptake. The surface-tension-driven flow model may not be applicable for Powder C in a tumbling drum and the impact forces in the drum are insufficient to cause consolidation.

Granules formed in the tumbling drum by drop controlled nucleation match the basic predictions for internal structure of Hounslow Granules. The created granules do not, however, reach that ideal term, with each of the granules retaining a significant void near the center after five or fifteen minutes, depending on material. The Hounslow Granule model is too simple for the tested powder materials so quantitative comparison is a moot point. Gas intrusion into the granule could replace the liquid as it moves to the surface and create a long lasting cavity within the granule.

## 5.5 Conclusions

Powder A granules form a complex internal structure with a hollow core surrounded by large agglomerates created from ultra-fine primary particles. A powder shell forms around the granule as granulation time increases, but the hollow core and agglomerate

structure does not change or shrink in significant fashion. The hollow structure is stable and difficult to consolidate.

Powder C granules rapidly achieve a final state and consolidation is not caused by surface-tension forces. Impact forces in the drum are insufficient to cause consolidation. There are no measureable changes in  $\epsilon_{\text{void}}$  or macro-void size as a function of time or liquid viscosity.

Powder A granules have a complex internal structure of large agglomerates created from ultra-fine primary particles that the Hounslow Granule does not consider or account for. This results in two separate time-scales that must be considered in order to describe granule nucleation and growth. Current literature models do not account for the existence of multiple time scales that exist in granulation of ultra-fine powders because of the capability to create large, complex, and stable structures like the millimeter sized agglomerates observed in this study. Future models for ultra-fine powders and other cohesive materials will have increased accuracy by considering the existence of time scales for primary particles and complex structures.

The microstructure analysis technique is capable of picking up some of the effects of consolidation on granule microstructure. It successfully measures the change in  $\epsilon_{\text{void}}$  that comes from creation of the shell for Powder A granules, as does the radial distribution. The drum granulation system with this combination of materials was insufficient to cause consolidation. The different powders were shown to achieve a consistent granule microstructure within a few seconds of droplet addition.

The surface-tension-driven flow model for the nucleation immersion mechanism does not accurately describe the behavior of Powder A and Powder C granules formed in the tumbling drum at 25 RPM. The created granules do initially form with large voids at the center as predicted for Hounslow Granules. However, the center void does not significantly decrease in size with time for either Powder A or Powder C granules. The granule consolidation behavior may be dominated by impact forces which are insufficiently strong in the tumbling drum to cause particle rearrangement.



## CHAPTER 6. POWDER FEEDING

### 6.1 Introduction

The work presented in Chapters 3, 4, and 5 evaluate and quantify the granulation behavior of ultra-fine powders. Another area of interest is in continuous feeding of ultra-fine powders for pharmaceutical manufacturing applications. One method of feeding in industrial use for feeding ultra-fine powders are loss-in-weight feeders (LIW). In Chapter 6, the relationship between powder properties and quality of feeding behavior is evaluated for several ultra-fine and coarser powders.

### 6.2 Materials & Methods

The powder materials used for this project are crosscarmellose sodium (NaCMC), sodium stearyl fumarate (SSF), magnesium stearate, semi-fine APAP, micronized APAP, and micronized APAP with SiO<sub>2</sub>. Materials were supplied by Eli Lilly. A Freeman FT4 Powder Rheometer has been used for powder property measurements. A KTRON KT-20 twin screw feeder (provided by Eli Lilly) was used for the powder feeding experiments. Coarse and fine screw sets were used with the feeder. Materials have been stored at 22-23 °C with relative humidity levels below 20% for the duration of the project.

The FT4 measurements were performed on both raw and fed powder samples. A raw powder was defined as material as received and stored at Purdue. A fed powder is one

that has passed once through the KT-20 feeder. These fed materials refer to coarse screw feeding, except for magnesium stearate which was only fed with a fine screw set. A new sample was used for each test to avoid potential powder history effects. The FT4 measurements were made using default test programs for BFE/Stability, compressibility, and 3 kPa pre-shear consolidation pressure for the shear cell. Shear cell tests at different pre-shear consolidation stresses (1, 2, 4 kPa) were created by scaling proportionately from the 3 kPa test program. BFE tests were performed 3 times each, compressibility once, and 1-4 kPa pre-shear consolidation pressure were performed once each.

Powder feeding experiments were performed at 2%, 5%, 10%, and 25% of the available drive command (% DC) while running in volumetric mode. The settings for starting fill level, screw configuration, and feeder flow factor are contained in Table 6.1. The target length for each experimental run was 35 minutes. Some experiments did not reach that target time due to disruptions, typically material backup at the feeder exit. The feeder was filled to target net weights of 2.8-3 kg for NaCMC, 2-2.5 kg for APAP materials, and 1.5 kg for SSF and magnesium stearate. The material flow rate was calibrated via the auto calibration function to determine the appropriate feeder flow factor for each material. That net weight value  $\pm 100$  g for each material was used as the starting set point for all experiments. The system was emptied and cleaned for new materials but not for changes in screw speed. The feeder was run for approximately 3 minutes at each new speed prior to starting a new experimental run. This was done to remove any

feeding history related to the previous screw speeds. Powder materials were collected in a bag attached to the outlet to reduce dust production. One experiment was performed at each screw speed using fresh raw material. Two additional experiments were performed at 5% and 10% DC using previously fed material.

Table 6.1: Feeding Experiment Parameters

Material	Fill Level	Screw Set	Feeder Flow Factor (kg/hr) Raw	Feeder Flow Factor (kg/hr) Fed
Crosscarmellose Sodium (NaCMC)	2.8-3 kg	Coarse/Fine	20.2/12.1	22.2/12.04
Semi-Fine APAP	2.3-2.5 kg	Coarse	16.8	16.7
Micronized APAP	2.2-2.3 kg	Coarse	10.5	9.5
Micronized APAP/SiO <sub>2</sub>	2.1-2.2 kg	Coarse	11	15.8
Sodium Stearyl Fumarate (SSF)	1.5-1.7 kg	Coarse/Fine	9.5/4.05	11.2/2.4
Magnesium Stearate	1.4 kg	Fine	2.6	1.94

The feeder has a control box which calculated mass flow rates using built-in smoothing functions. These values, including raw net weight data, are recorded every 1 second and the previous hour of data can be retrieved as trace data. The control box did not record new values if the data variability triggers certain internal alarms and instead reports a steady value, shown in Figure 1 for the 10% DC experiments. The available sampling times for the control box ( $\leq 8000$  mS) proved insufficient for most experiments. Material mass flow rates have been manually calculated from the raw net

weights at 30s intervals, with outliers removed, using the following equation of simple differences in net weight:

$$MF \left( \frac{kg}{hr} \right) = \frac{(NW_{t=1} - NW_{t=30})}{30} * 3600 \quad (6.1)$$

The starting point of the 30s samplings have been shifted up to 10s to avoid outliers.

Relative standard deviations (RSD) have been calculated in a simple fashion using the mean flow rate for each experiment:

$$\%RSD = \frac{\sigma_{data}}{\bar{x}_{data}} * 100 \quad (6.2)$$

A fitting spline program in MATLAB has also been used to fit the data and measure variability. The RSD has been calculated as in equation 3:

$$\%RSD_{spline} = \frac{\sqrt{SSE_{fit}}}{N * \bar{x}_{data}} * 100 \quad (6.3)$$

where the SSE is the sum of squares error for the fitted line and N is the number of samples. The sum of squares error uses the difference between the data point and the fitted point at that time value rather than the global mean mass flow rate for the system. This is useful and provides a more accurate representation for materials with a clear time dependency where the mass flow rates at the beginning and end of the experiment are significantly different.

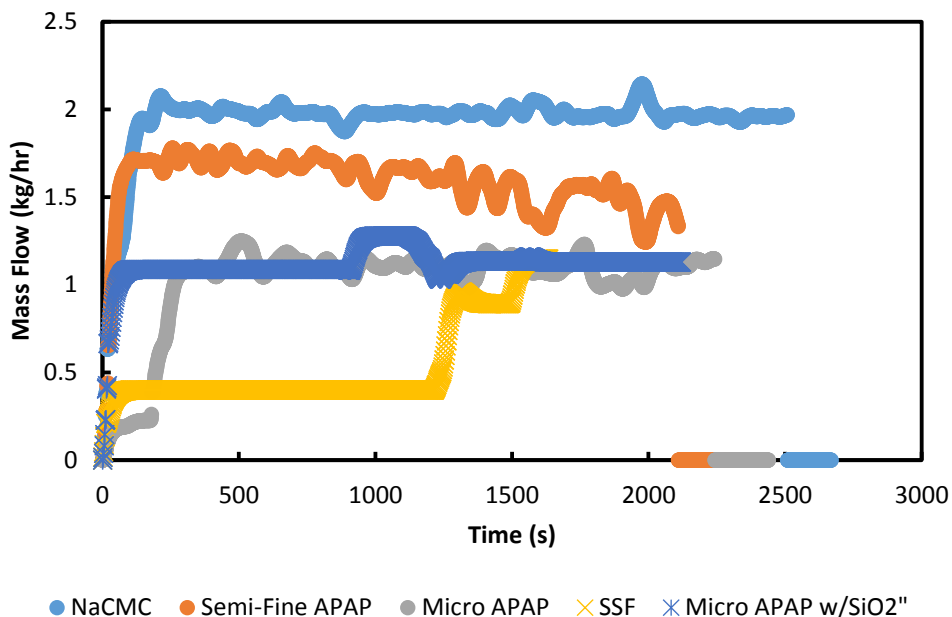


Figure 6.1: Mass flow rates reported by control box using 1 second sampling times for 10% DC. Large amounts of data are lost for SSF and MicroAPAP w/ SiO<sub>2</sub>

### 6.3 Results & Discussion

#### 6.3.1 FT4 Results

##### 6.3.1.1 Shear Cell

The primary outputs from shear cell tests are the maximum principle stress (MPS) and the unconfined yield stress (UYS) which can be used to define a powder flow factor (ffc). Additional calculated values from the FT4 include cohesion and angles of internal friction. Data from the shear cell tests is given in Appendix C. A plot of UYS against MPS using measurements at three pre-consolidation pressures is sufficient to describe powder behavior. Cohesive materials, which typically have poor flow behavior, will have a slope closer to 1 and have higher unconfined yield stress than free-flowing materials. The results in Figure 6.2 (fresh material) indicate that crosscarmellose

sodium and sodium stearyl fumarate are free-flowing while the APAP materials are more cohesive. Micronized APAP is the most cohesive material by this measure. The addition of nanosilica to micronized APAP has changed its behavior closer to that of Semi-Fine APAP. The Figure 6.3 results show that the effects of feeding are negligible at higher pressures for most materials. There may be an effect for micronized APAP, but it is unclear because the experimental variability is unknown.

The expected feeding behavior based on shear cell results would be that crosscarmellose sodium and sodium stearyl fumarate have the lowest feed rate variability and micronized APAP has the highest feed rate variability.

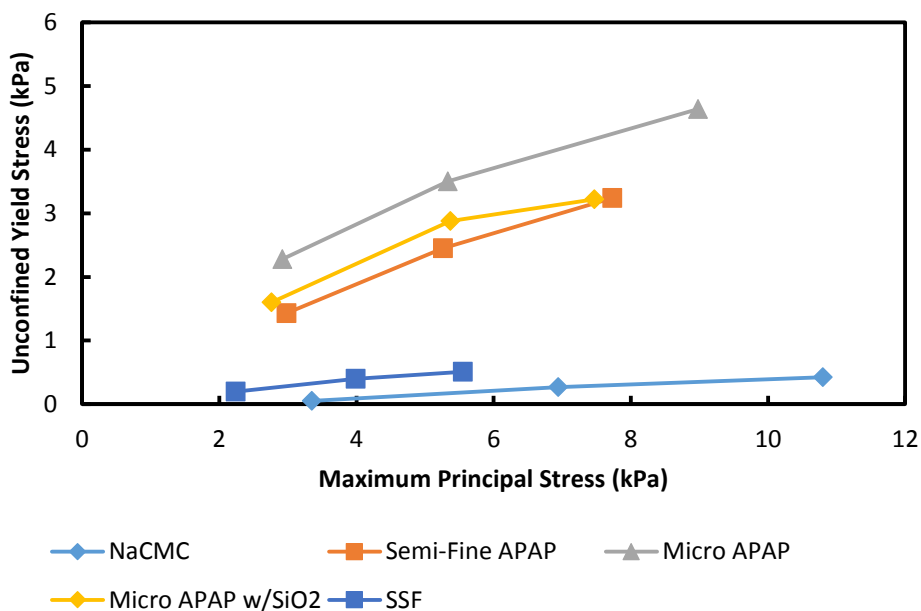


Figure 6.2: Shear Cell results for raw materials, unconfined yield strength and maximum principal stress. Scaled so that x and y axes are proportionately sized

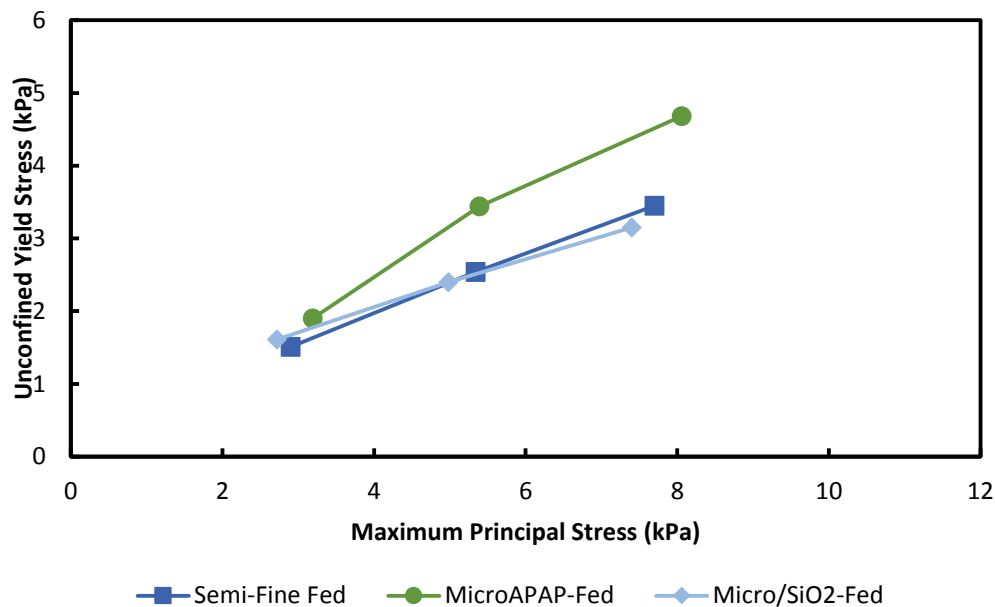


Figure 6.3: Shear cell results for previously fed APAP materials. Scaled so that x and y axes are proportionately sized

#### 6.3.1.2 Basic Flowability Energy

The primary Variable Flow Rate (VFR) and Stability test outputs are the Basic Flowability Energy (BFE) and Specific Energy (SE). The SE value is intended to describe unconfined flow behavior and is not representative of a twin-screw feeder. The BFE is intended to describe confined flow behavior which does describe a twin-screw feeder. Figure 6.4 shows the BFE for the material sin this study. Primary data from the VFR tests are given in Appendix C. One might expect that SSF will have the best flow behavior from BFE and crosscarmellose sodium will have the worst flow behavior from BFE. This differs strongly from the shear cell data which predicted that both crosscarmellose sodium and SSF will have good flow behavior. There is no clear effect from feeding on BFE.

Crosscarmellose sodium and micronized APAP show some decrease in BFE, but the other material values are essentially unchanged.

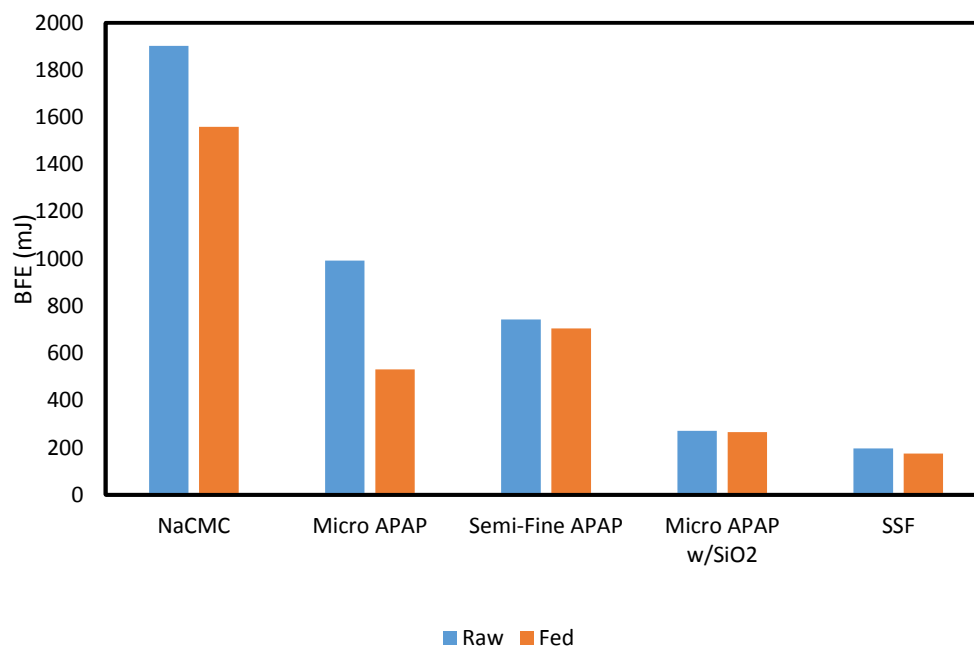


Figure 6.4: Basic flowability energy results for coarse screw materials, covering Raw and Fed materials

### 6.3.1.3 Compressibility

Figure 6.5 shows the measured compressibility at 20 kPa axial compaction stress. The compressibility of materials correlates to qualitative descriptions in powder behavior where free-flowing materials typically have low compressibility and cohesive materials have higher compressibility. Crosscarmellose sodium is predicted to have the best feeding behavior and the APAP materials are expected to have the worst feeding behavior based on the compressibility data. With the exception of micronized APAP



with SiO<sub>2</sub>, there was no significant effect of feeding on compressibility. Compressibility does not distinguish between the behavior of semi-fine and micronized APAP.

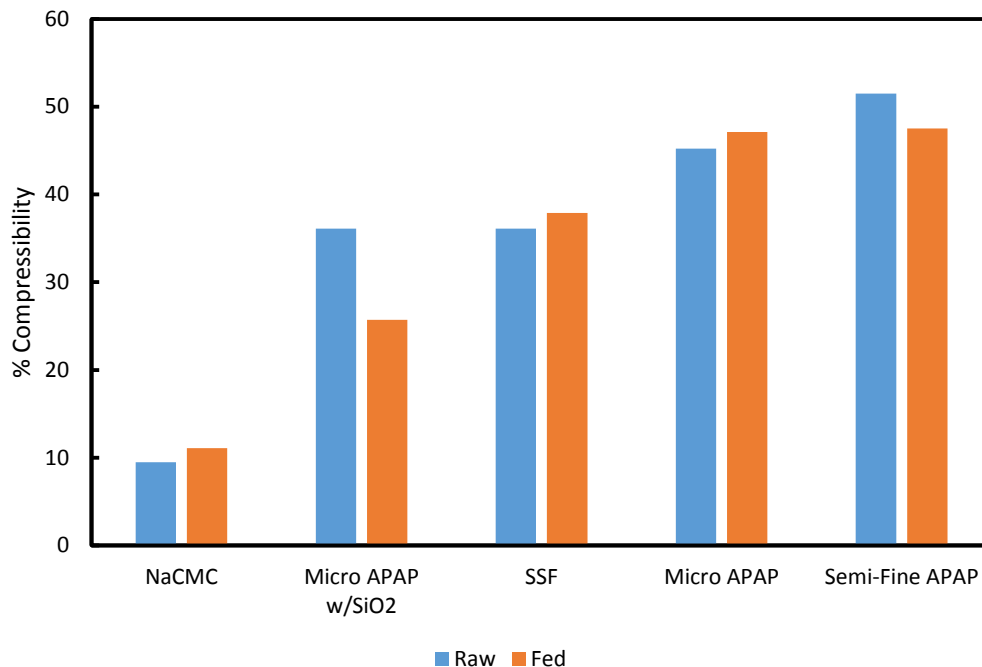


Figure 6.5: % Compressibility for raw and fed powders

### 6.3.2 Feeder Experiments

Primary data for all feeder experiments is stored in an electronic appendix, referenced in Appendix D.

#### 6.3.2.1 Qualitative Observations

The crosscarmellose sodium exits the feeder in a fairly continuous stream and all other materials exit as clumps of varying sizes. The micronized APAP w/ SiO<sub>2</sub> and sodium stearyl fumarate exit as small clumps which break up on impact. The micronized APAP and semi-fine APAP form clumps of various sizes and some of the material sticks inside the exit. The material does not leave the area that is measured by the balance and so is

not counted. This indicates lost material and presents a potential disturbance if a lump breaks off later on. Material also coats metal surfaces further from the exit, so this material is counted as leaving the feeder but does not actually reach its destination. In this case, the measured (or calculated) material flow rates are not fully representative of how much material is leaving the screws and would induce problems for runs in gravimetric mode. The other materials do not stick to surfaces in noticeable quantities.

Material exiting as clumps is a significant problem at 2% and 5% DC because the screw speed is low (3 and 7.5 RPM) and clumps only fall out every few seconds. This makes accurate calculations of the mass flow rate difficult, especially if using small sampling times, because the mass existing the hopper consists almost entirely of isolated events.

At 10% and 25% DC the screws turn quickly enough so clumps exit in a continuous stream and alleviate this concern.

The mass flow rate of the system can vary with time because the feeder is run in volumetric mode. Figure 6.6 shows the mean mass flow rate against the setpoint mass flow rate with dashed lines representing  $\pm 10\%$  of the setpoint. The mean mass flow rates are typically within  $\pm 10\%$  of the setpoint. The micronized APAP w/  $\text{SiO}_2$  at 25% DC has a mean flow rate that is 86% of the setpoint.

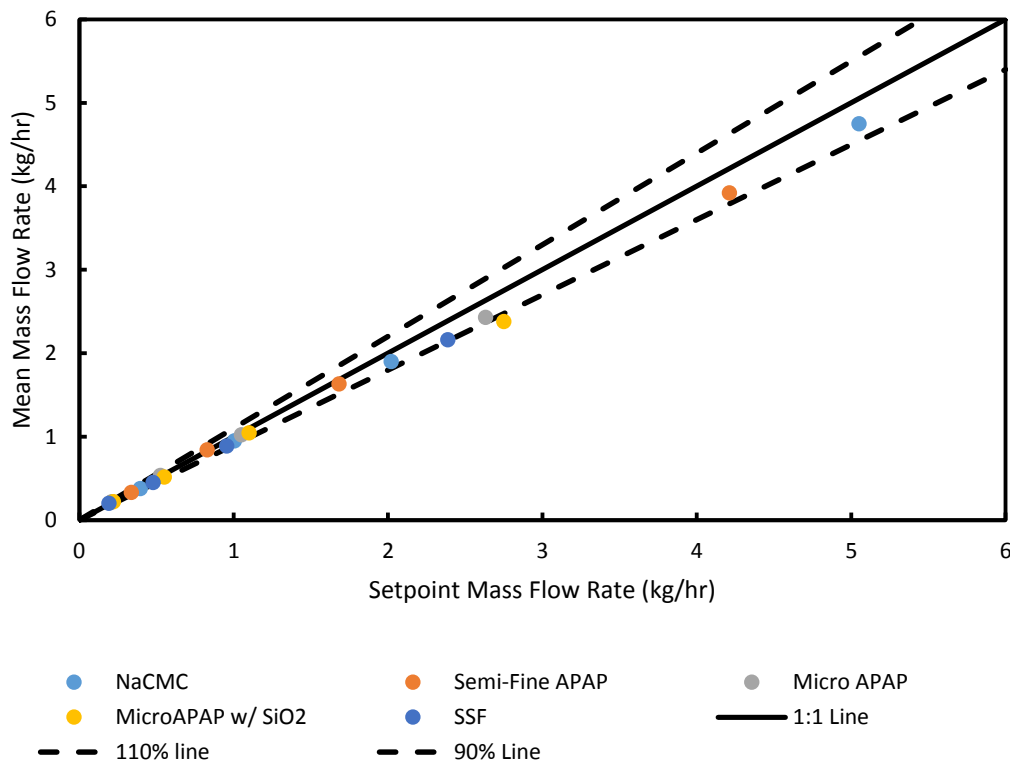


Figure 6.6: Calculated Mass Flow Rate against Setpoint Mass Flow Rate with 1:1 line and +/- 10% lines

The 25% DC results in Figure 6.7 show that the flowrate decreases at a rate which varies with material. The effect is significant for semi-fine APAP, where flowrate decreases from 4 kg/hr to 2 kg/hr over 35 minutes, but is smaller for other materials.

Crosscarmellose sodium flowrate decreases slightly with time. Micronized APAP, micronized APAP with SiO<sub>2</sub>, and sodium stearyl fumarate all experienced some user related disruptions. The data after these recorded disruptions have been removed. The disruptions include an overflowing bag causing material to back up in the hopper exit and bumping the balance.

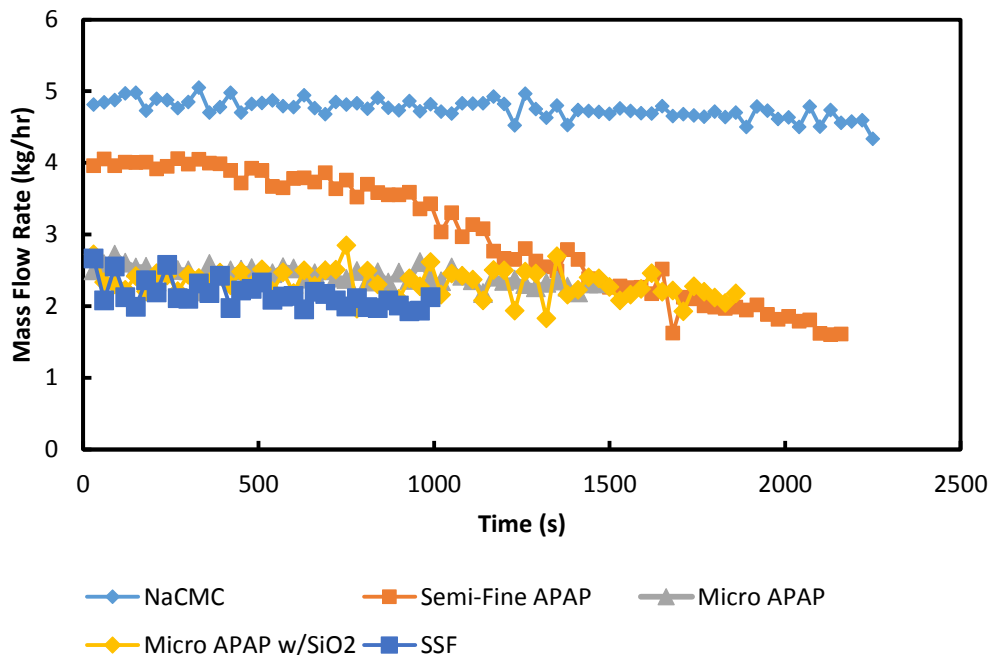


Figure 6.7: 25% DC Coarse Screw Mass Flow Rates calculated from net weight data

The mass flow rate calculations for 10%, 5%, and 2% DC are shown in Figures 6.8-6.10, respectively. The micronized APAP w/ SiO<sub>2</sub> and sodium stearyl fumarate show significant variations between maximum and minimum values on either side of the target flow rate, especially at 2% and 5% DC. This is likely to be the result of the clump feeding behavior described earlier. The existence of negative flow rate values at 2% DC at is difficult to explain (seen in Figure 6.10 for micronized APAP w/ SiO<sub>2</sub>) but persists regardless of the sampling start point. The negative flow rates appear to be a systematic issue with the data reported by the balance.

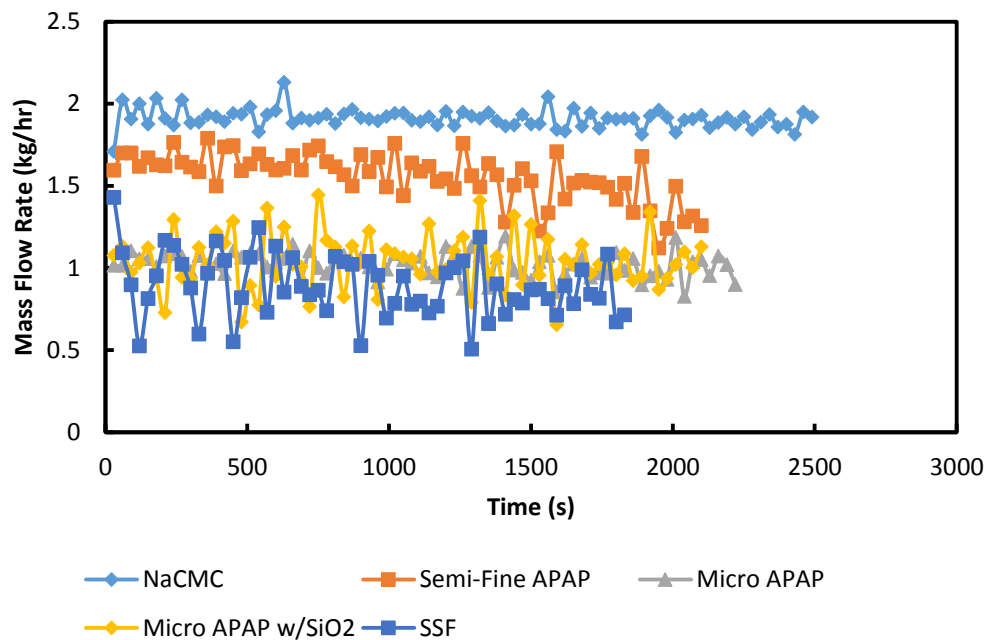


Figure 6.8: 10% DC coarse screw mass flow rates calculated from net weight data

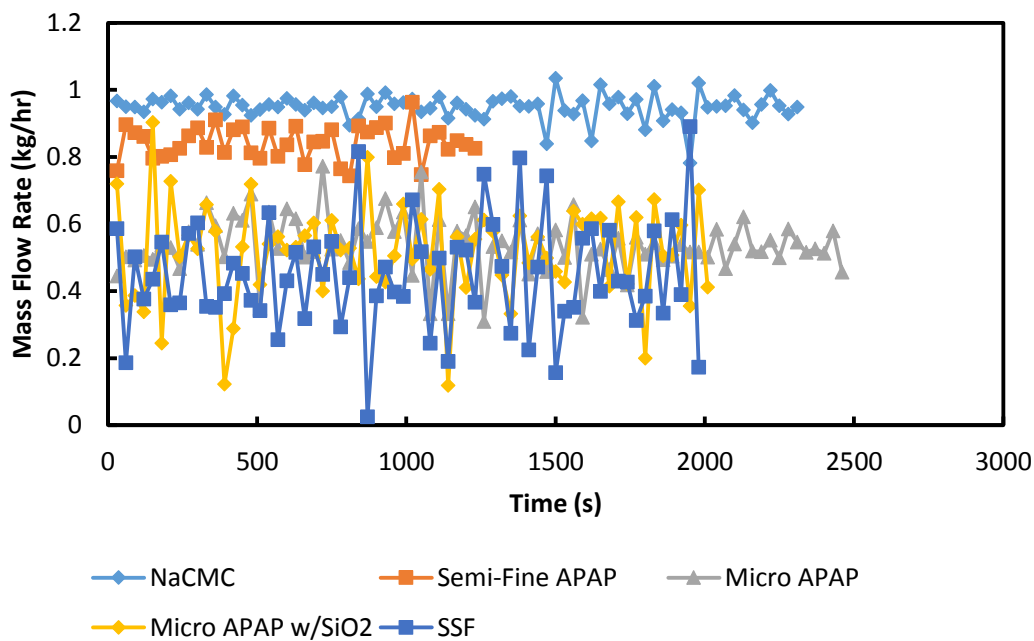


Figure 6.9: 5% DC coarse screw mass flow rates calculated from net weight data

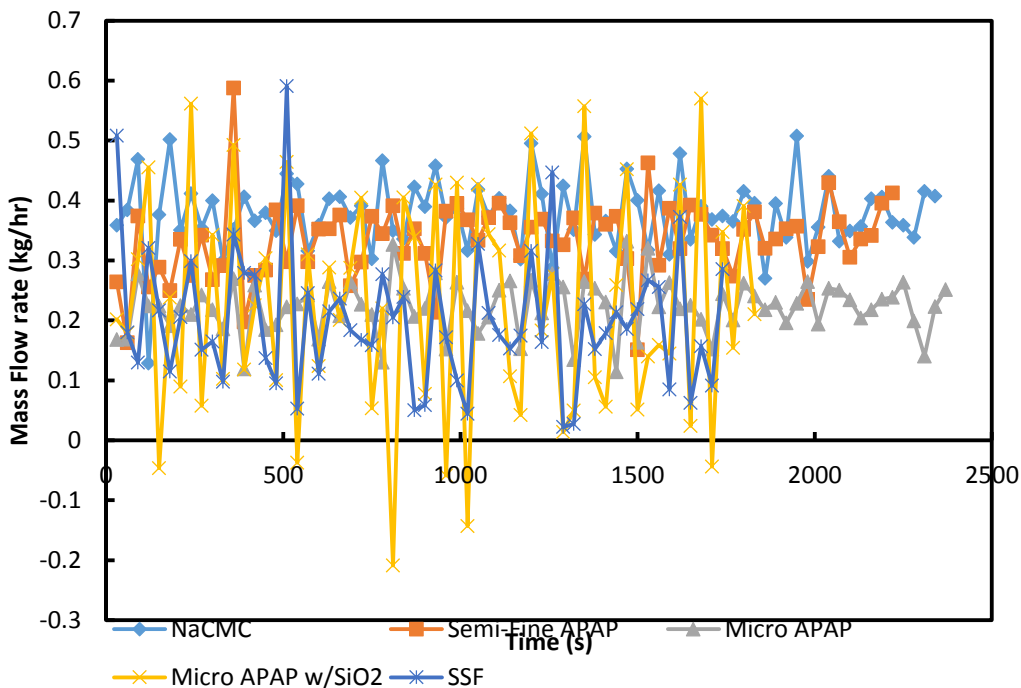


Figure 6.10: 2% DC coarse screw mass flow rates calculated from net weight data

### 6.3.3 Variability Results

#### 6.3.3.1 RSD Results

The relative standard deviation (RSD) values were calculated using Eqn. 6-2. Figure 6.11 shows the RSD values for the 25% DC experiments arranged from low to high.

Crosscarmellose sodium has the lowest variability and sodium stearyl fumarate has the largest. The APAP materials are arranged in order of increasing RSD as follows; semi-fine, micronized, and micronized w/ SiO<sub>2</sub>. Figure 6.12 shows the RSD values at all screw speeds. RSD increases as speed decreases and the rank order of materials does not change. The first 10 minutes of the 25% DC experiments have been used for RSD calculations of semi-fine APAP and micronized APAP with SiO<sub>2</sub>. The strong decreasing

trend in mass flow rate with time for semi-fine APAP at 25% screw speed would make the data otherwise unusable for this method (see Figure 6.7). There is an unexplained disturbance for the micronized APAP with SiO<sub>2</sub> in the middle of the experiment that is smoothed out by the end of the experiment. The mass flow rate % RSD for micronized APAP with SiO<sub>2</sub> when using all the results is 8-9%.

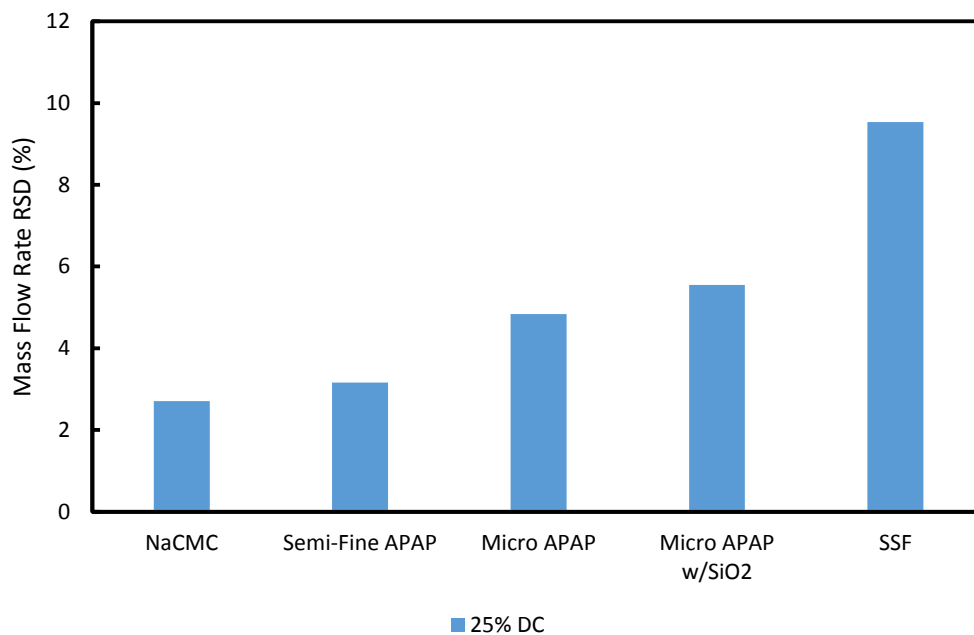


Figure 6.11: Mass Flow Rate RSDs sorted low-high for 25% DC, non-spline fitting

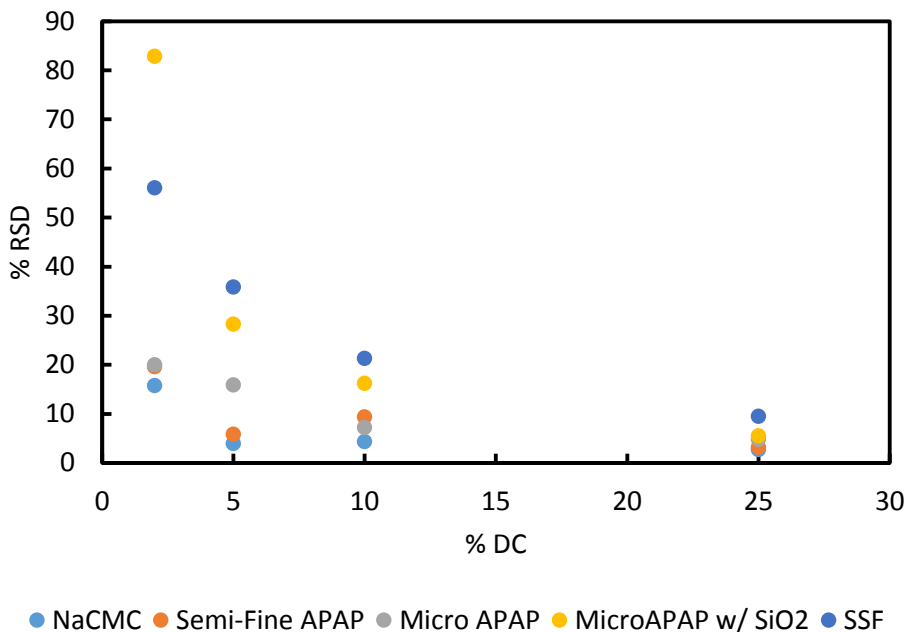


Figure 6.12: %RSD against different materials and screw speeds for coarse screw experiments.

The comparisons of % RSD with material properties are performed first at the 25% DC level as it has the lowest variability. This approach shows a potential relationship between the BFE and the % RSD. BFE results in Figure 6.9 show that BFE has a negative correlation with RSD. There are no apparent relationships between % RSD and unconfined yield stress or % compressibility shows in Figure 6.10 and Figure 6.11. The same correlation for BFE and % RSD does exist at the lower screw speeds and the same lack of trend exists for the other properties. However, the BFE results contradict the expectation that low BFE values would result in low RSD. The minor calculated variables from the compressibility and shear cell tests are not being heavily evaluated because their primary properties of interest show no relationship for the cohesive materials studied.



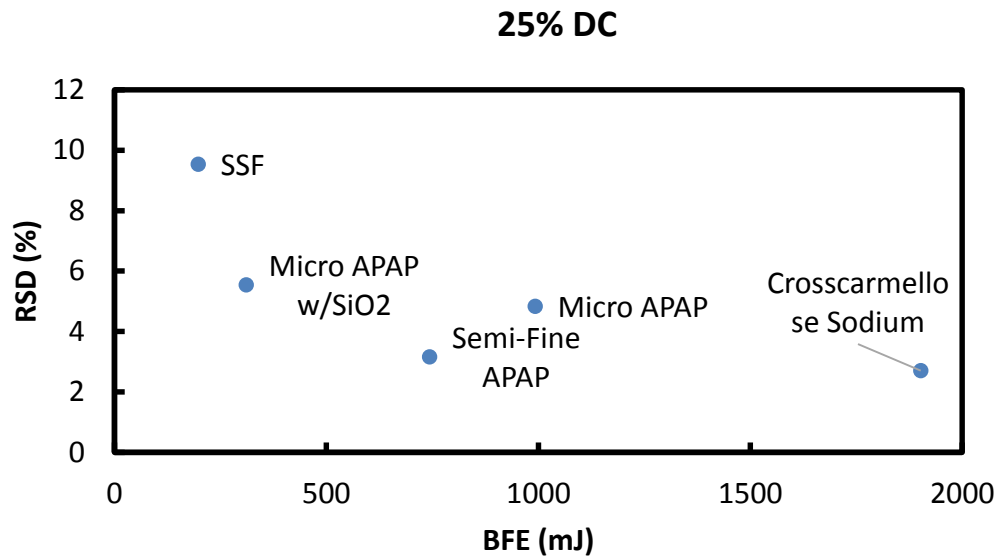


Figure 6.13: %RSD as a function of the BFE at 25% DC for coarse screw experiments

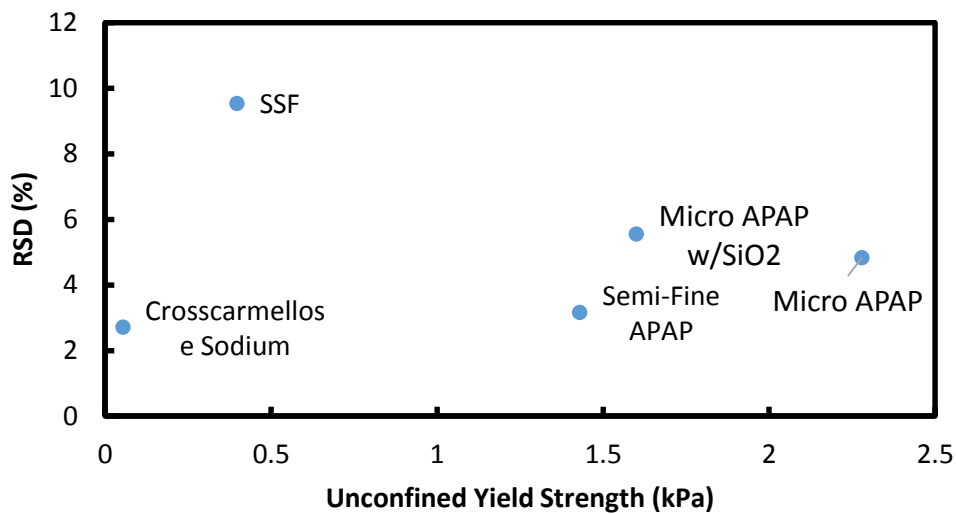


Figure 6.14: %RSD as a function of unconfined yield stress (kPa) at 25% DC for coarse screw experiments

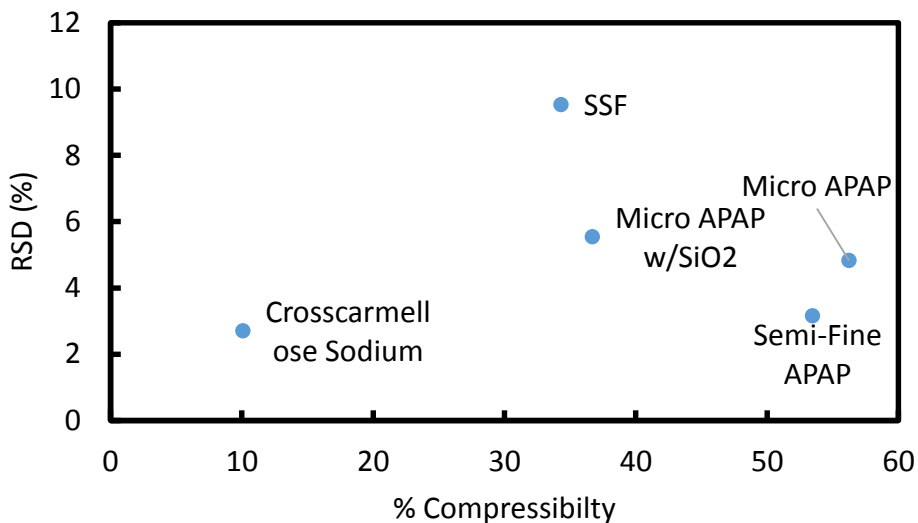


Figure 6.15: %RSD as a function of compressibility for 25% DC coarse screw experiments

#### 6.3.4 Spline Fitting

The RSD values from spline fitting are plotted against screw speed in Figure 6.16. The spline fit RSD values are significantly smaller than the RSD values from Figure 6.12. However, the rank order of materials with increasing RSD values is unchanged (Figure 6.11 and 6.17). The spline fitting makes use of the entire dataset except where data has been removed because of recorded disruptions or outliers. The rank order using RSD is surprising because the addition of SiO<sub>2</sub> is expected to improve flow behavior. However, the micronized APAP w/ SiO<sub>2</sub> has a higher variability than the micronized APAP and semi-fine APAP.

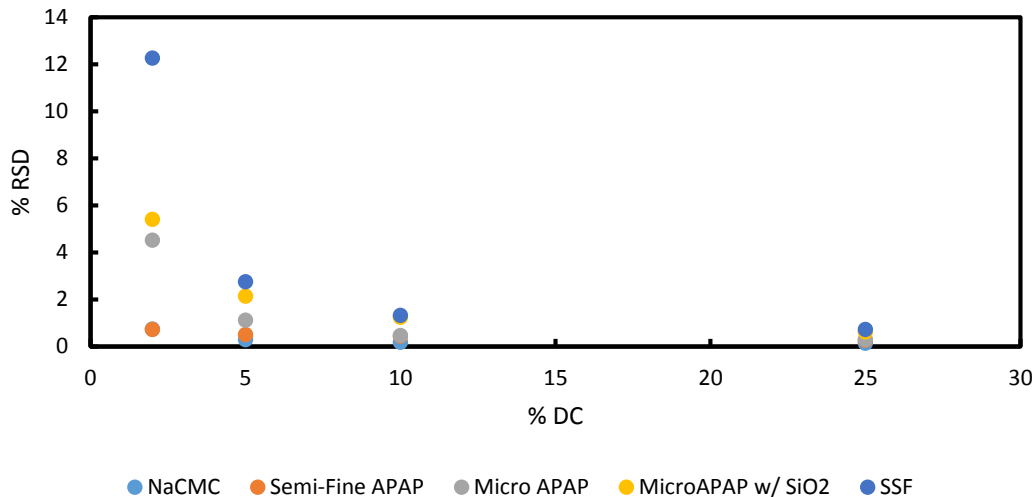


Figure 6.16: %RSD against % DC for spline fit calculated RSDs

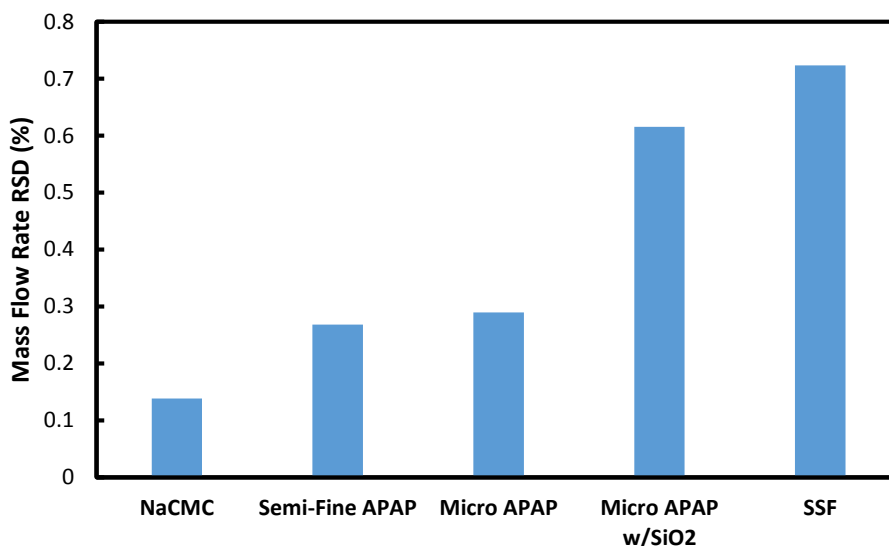


Figure 6.17: Spline Fitting RSD values for 25% DC

### 6.3.5 Fine Screw Experiments

Feeding experiments were performed using crosscarmellose sodium, sodium stearyl fumarate, and magnesium stearate using a set of fine screws provided by Eli Lilly. The experimental methodology is identical to the coarse screw experiments. Fine screw results show similar trends to the coarse screw results. The increase in speed reduces

the measured RSD values in Figure 6.18 and in Figure 6.19 the higher RSD values are associated with smaller BFE values. The measured RSD values are somewhat higher for the fine screw experiments and RSD values for comparable mass flow rates (not screw speeds) are slightly higher. The difference in RSD may be caused by experimental variability and suggests that running at a higher screw speed with smaller screws does not provide a more consistent delivery of raw materials.

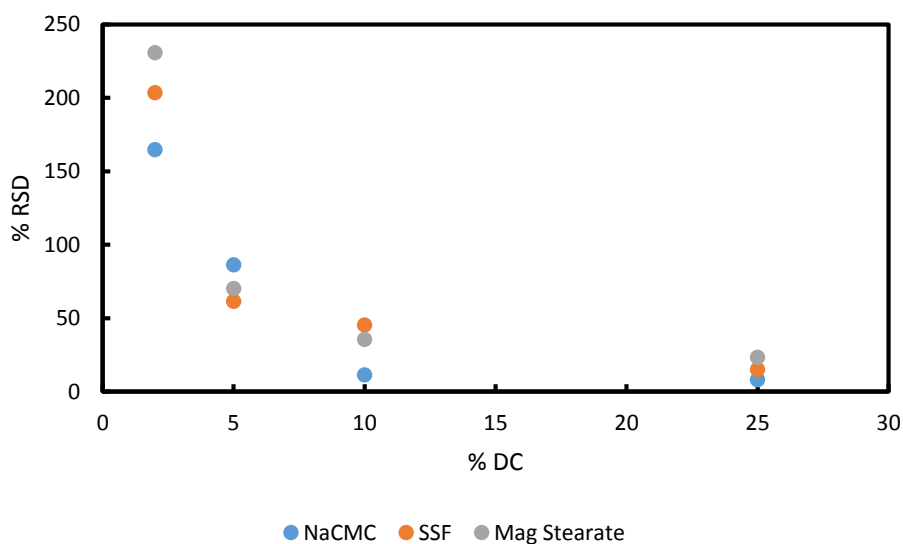


Figure 6.18: RSD results for fine screw experiments

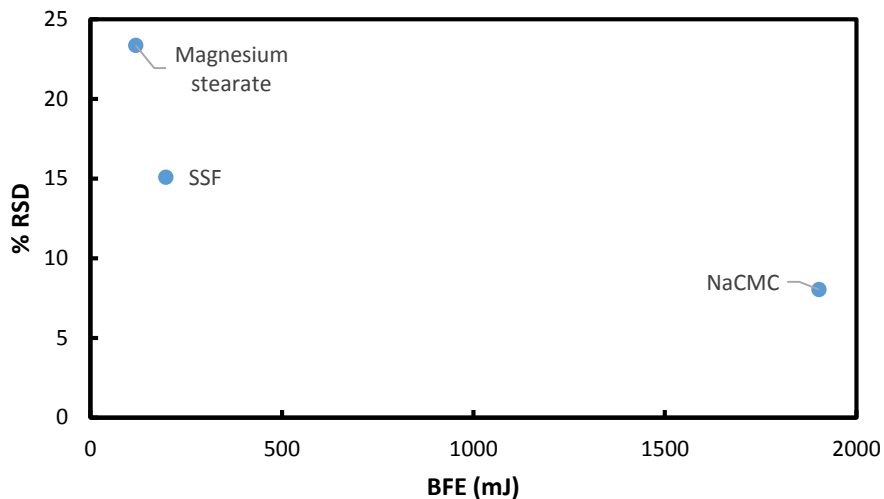


Figure 6.19: RSD against BFE for fine screw experiments at 25% DC

### 6.3.6 Fed Material Experiments

The fed material was collected and reused for two experiments each at 5% and 10% DC to evaluate if the previous feeding has an effect on material behavior. The results in Figure 6.20 show that high variability materials (micro APAP w/SiO<sub>2</sub> and sodium stearyl fumarate) show significant decreases (25-50%) in calculated %RSD at both speeds while other materials show either no change or an increase in variability at only one speed. This does not align well with any of the FT4 results which generally showed little or no change for powder properties.

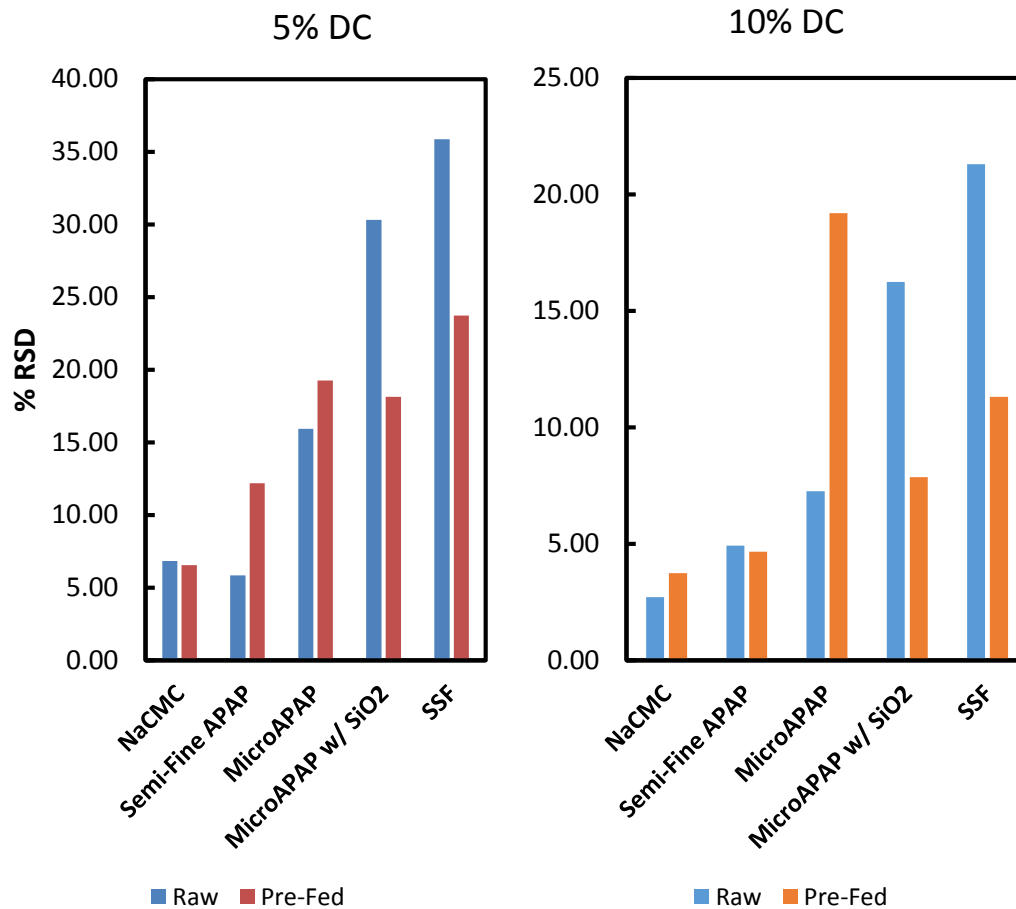


Figure 6.20: % RSD for raw and fed material coarse screw experiments at 5% and 10%

DC

#### 6.4 Conclusions

The use of RSD to describe material feeding behavior does not provide the expected results. Sodium stearyl fumarate is ranked as the worst material using RSD, but gives the best feeding behavior (of those tested) based upon industrial experience.

Crosscarmellose sodium provides the opposite effect by having the lowest RSD and many difficulties in practice. The behavior of micronized APAP with SiO<sub>2</sub> is actually

worse based upon RSD than unadulterated micronized APAP. It has been shown that increasing screw speed does result in lower mass flow rate variability.

The powder flow function measured by shear cell tests does correlate with observations of material problems due to caking and powder adhesion to surfaces. To predict caking and consolidation of as received material (crosscarmellose sodium), longer time tests in controlled humidity environments are suggested, e.g. measurement of the time dependent powder flow function.

The use of the FT4 shows that individual tests will rank materials differently and give conflicting results when used to make behavioral predictions. There are no clear effects of single-pass feeding on measured material properties, with a possible exception of the micronized APAP with SiO<sub>2</sub>. Ultra-fine materials pose unique challenges in the measurement of flow rates, especially at low feed rates, because of the tendency for the material to form clumps.

This work shows that there is currently not a single material measurement which will accurately describe material behavior and likelihood of various failure during feeding. Fed materials can exhibit a wide variety of failure modes in hoppers and screw systems, including arching and rat holing in the hopper, or compaction in screw flights or surface adhesion at the feeder exit leading to blockages. A single parameter may be capable of predicting the likelihood of one failure method, such as compaction, but not of all failure modes.

## CHAPTER 7. CONCLUSIONS AND FUTURE DIRECTIONS

### 7.1 Conclusions

The main goal of this work was to quantify the complex behavior of ultra-fine powder materials in wet granulation using 3D measurements of granule microstructure. In Chapter 4, the complex behavior of ultra-fine powders was first observed from creation of large, stable agglomerates through sifting of 0.5  $\mu\text{m}$  alumina (Powder A). The average agglomerate sized decreased as the sieve opening decreased from 1.4 mm to 710  $\mu\text{m}$  to 500  $\mu\text{m}$ . The presence of stable agglomerates was not observed in static powder beds formed from sifting 5  $\mu\text{m}$  (Powder B), 25  $\mu\text{m}$  (Powder C), or 108  $\mu\text{m}$  (Powder D) alumina through the various sieves. Single-droplet granules formed from these powder beds matched literature predictions for the nucleation behavior, with Powder A and Powder B forming Tunneling granules while Powder C and Powder D formed Spreading/Crater granules.

Existing and newly developed image analysis techniques, discussed in Chapter 3 and applied in Chapter 4, were able to distinguish between different granule meso-structures. The granule internal structures for Powder A and Powder B were composed of large spheroidal macro-voids and a particle matrix. Powder C granules were found to have large macro-voids but the macro-voids were identified as “cracks” rather than spheroids. The



difference between spheroids and “cracks” was identified by granule sphericity, accomplished by creation of a new measurement technique for measuring the surface area of labeled 3D objects within the granule. Powder D granules were found to have no macro-voids at all. The macro-voids were found to take up to 7% of the granule volume and the largest macro-voids are 200-800  $\mu\text{m}$  volume equivalent sphere diameter. Decreasing granule size resulted in larger macro-voids and a larger  $\epsilon_{\text{void}}$  value. Decreasing the opening of the sieve used for powder bed preparation decrease the macro-void size and  $\epsilon_{\text{void}}$ . This work showed the need for greater care in handling and preparation of ultra-fine powders because it can profoundly affect final granule properties.

Single-droplet granules formed in a tumbling drum were found to have a hollow core surrounded by either a dense powder layer or a packed structure of large, stable agglomerates depending on primary particle size. The Powder C granules rapidly reached a consistent hollow structure which remained stable even after five minutes. Within experimental error, there were no effects of consolidation time or liquid binder viscosity on macro-void size or  $\epsilon_{\text{void}}$ . The Powder A granules initially produced an open structure of large agglomerates linked by solid bridges surrounding a hollow center. After 5 minutes, Powder A granules develop a powder shell which coats the previously observed open structure of large agglomerates and the shell remains up to 15 minutes. Increasing binder viscosity was observed to increase the size and volume fraction of macro-voids in Powder A granules. The observed decrease in macro-void size and  $\epsilon_{\text{void}}$  from 3 seconds to

5 minutes consolidation time for Powder A granules in the drum is most likely caused by the creation of the powder shell.

The Powder A and Powder C granules form the initial granule structure predicted by the nucleation immersion mechanism, but do not match the model predictions for a shrinking hollow core. Gas infiltration into the granule center is one possible explanation for the creation of a stable center as liquid moves to the exterior of the granule. For Powder A granules, the model is also too simple to account for the complex behavior of ultra-fine powders.

The feeding behavior of ultra-fine powders and other cohesive materials (chapter 6) is difficult to quantify nor does a single material property predict the existence of issues with powder feeding. The RSD of the mass flow rate exiting a twin screw feeder does not correlate with the unconfined yield stress, powder compressibility, or the basic flowability energy. Powders were observed to have multiple modes of failure which creates significant complications for a single parameter measurement. Shear cell results were observed to predict simple flow issues, but not the RSD or likelihood of feeding failure. Additional tests at varied moisture content and time consolidation tests would provide additional information for comparison. A single material property may be capable of predicting a particular failure mode during powder feeding.

## 7.2 Recommendations for further work

The studies presented in this thesis are the first to do a detailed study of the behavior of ultra-fine powders during wet granulation and related processes. Therefore, there is a

large amount of work that remains in order to understand and improve the processes using them. The proposed areas of further study are broken into four main areas. These are improvements in the analytical tools and run time, consolidation studies of ultra-fine powders, new models accounting for complex structures and multiple length scales, and new analytical techniques for quantifying twin screw feeder behavior and linking it to material properties.

There is a need for continuing improvements and additions to the microstructure analysis techniques used here for quantifying granule microstructure. Techniques to identify and remove external void fragments from the analysis would be especially useful. This can likely be accomplished by linking positional information about the labeled fragments to some sort of edge detection. A second area for improvement is in formatting the code to better use advances in computing, such as parallel processing and multi-core systems, to significantly decrease the time required to process a granule. Improving the run time will allow for additional analysis without sacrificing the quantity of granules that can be processed, leaving the XRCT scan time as the primary bottleneck without creating a new one. This will require a rewrite of the existing code to take advantage of the faster architecture. Tools for linking granule microstructure to consolidation effects would be useful for determining the kinetics of the consolidation and/or layering processes.

Studies of consolidation behavior involving ultra-fine powders, both as single materials, and as mixtures, are needed in a variety of equipment. Fluidized beds can show formation of complex agglomerate structures similar to those observed in Chapter 4, while high

shear mixers may overwhelm the cohesive material nature. Powder mixtures using ultra-fine powders are industrially relevant, but the effect on microstructure of combining these materials has not been evaluated.

The drum granulation study clearly shows the need to account for complex behaviors when modeling ultra-fine powders. Current literature models only include a single length scale for predictions and do not consider the possibility of multiple relevant scales. This oversight explains why ultra-fine powders typically do not fit the existing models, such as for drop penetration time and immersion nucleation kinetics. Additional studies are needed to determine the types and relevant length scales of structures formed in different processes as fluidized beds and high-shear mixers have significantly different mechanics.

Studies of how to properly quantify the performance of continuous feeding systems is sorely needed. While techniques exist for comparing different types of feeders and attachments, the existing techniques are insufficient for comparing individual powder behaviors on a single piece of equipment and new ones need to be developed. Additional work is needed to properly measure the mass flow rate at low speeds, which the existing LIW feeding systems struggle. These issues apply to all powders that may be used in small quantities, not just ultra-fine powders.

## REFERENCES

## REFERENCES

1. Visser J. Van der Waals and Other Cohesive Forces Affecting Powder Fluidization. *Powder Technol.* 1989;58:1–10.
2. Litster J, Ennis BJ. *The science and engineering of granulation processes*. Kluwer Academic Publishers; 2004.
3. Hapgood KP, Litster JD, Biggs SR, Howes T. Drop penetration into porous powder beds. *J Colloid Interface Sci.* 2002;253(2):353–66. doi:10.1006/jcis.2002.8527.
4. Armstrong B, Brockbank K, Clayton J. Understand the effects of moisture on powder behavior.pdf. *AIChE J.* 2014;(October):25–30.
5. Emady H, Kayrak-Talay D, Litster J. A regime map for granule formation by drop impact on powder beds. *AIChE J.* 2013;59(1). doi:10.1002/aic.
6. Emady HN, Kayrak-Talay D, Schwerin WC, Litster JD. Granule formation mechanisms and morphology from single drop impact on powder beds. *Powder Technol.* 2011;212(1):69–79. doi:10.1016/j.powtec.2011.04.030.
7. Stock SR. Recent advances in X-ray microtomography applied to materials. *Int Mater Rev.* 2008;53(3):129–181. doi:10.1179/174328008X277803.
8. Stock SR. X-ray microtomography of materials. *Int Mater Rev.* 1999;44(4):141–164. doi:10.1179/095066099101528261.
9. Dale S, Wassgren C, Litster J. Measuring granule phase volume distributions using X-ray microtomography. *Powder Technol.* 2014;264:550–560. doi:10.1016/j.powtec.2014.06.009.

10. Dale, Steven Andrew. INTERNAL STRUCTURES OF WET GRANULATION AGGLOMERATES: MEASUREMENT, FORMATION, AND INFLUENCE ON GRANULE BREAKAGE. *Purdue Univ.* 2014.
11. Hounslow MJ, Oullion M, Reynolds GK. Kinetic models for granule nucleation by the immersion mechanism. *Powder Technol.* 2009;189(2):177–189. doi:10.1016/j.powtec.2008.04.008.
12. Iveson SM, Litster JD, Hapgood K, Ennis BJ. Nucleation, growth and breakage phenomena in agitated wet granulation processes: a review. *Powder Technol.* 2001;117(1-2):3–39. doi:10.1016/S0032-5910(01)00313-8.
13. Hapgood K., Iveson SM, Litster JD, Liu LX. Granulation Rate Processes. In: Salman AD, Hounslow MJ, Seville JPK, eds. *Handbook of Powder Technology*. Volume 11. Elsevier Science B.V.; 2007:897–977.
14. Litster J., Hapgood K., Michaels J., et al. Liquid distribution in wet granulation: dimensionless spray flux. *Powder Technol.* 2001;114(1-3):32–39. doi:10.1016/S0032-5910(00)00259-X.
15. Hapgood K., Litster JD, Smith R. Nucleation regime map for liquid bound granules. *AIChE J.* 2003;49(2):350–361.
16. Emady HN, Kayrak-Talay D, Litster JD. Modeling the granule formation mechanism from single drop impact on a powder bed. *J Colloid Interface Sci.* 2013;393:369–76. doi:10.1016/j.jcis.2012.10.038.
17. Charles-Williams HR, Wengeler R, Flore K, Feise H, Hounslow MJ, Salman AD. Granule nucleation and growth: Competing drop spreading and infiltration processes. *Powder Technol.* 2011;206(1-2):63–71. doi:10.1016/j.powtec.2010.06.013.
18. Iveson SM, Wauters P a. L, Forrest S, Litster JD, Meesters GMH, Scarlett B. Growth regime map for liquid-bound granules: further development and experimental validation. *Powder Technol.* 2001;117(1-2):83–97. doi:10.1016/S0032-5910(01)00317-5.

19. Iveson SM, Litster JD, Ennis BJ. Fundamental studies of granule consolidation Part 1: Effects of binder content and binder viscosity. *Powder Technol.* 1996;88(1):15–20. doi:10.1016/0032-5910(96)03096-3.
20. Iveson SM, Litster JD. Fundamental studies of granule consolidation part 2: Quantifying the effects of particle and binder properties. *Powder Technol.* 1998;99(3):243–250. doi:10.1016/S0032-5910(98)00116-8.
21. Ouchiyaama N, Tanaka T. The probability of Coalescence in Granulation Kinetics. *Ind Eng Chem, Process Des Dev.* 1975;14(3):286–289.
22. Ennis BJ, Tardos G, Pfeffer R. A microlevel-based characterization of granulation phenomena. *Powder Technol.* 1991;65(1-3):257–272. doi:10.1016/0032-5910(91)80189-P.
23. Liu LX, Litster JD, Iveson SM, Ennis BJ. Coalescence of Deformable Granules in Wet Granulation Processes. *AIChE J.* 2000;46(3):529–539.
24. Sochon RPJ, Dorvlo SK, Rudd a. I, Hayati I, Hounslow MJ, Salman a. D. Granulation of Zinc Oxide. *Chem Eng Res Des.* 2005;83(11):1325–1330. doi:10.1205/cherd.05099.
25. Schaefer T, Mathiesen C. Melt pelletization in a high shearmixer. IX. Effects of binder particle size. *Int J Pharm.* 1996;139((1-2)):139–148.
26. Pitt K, Smith R, Houslow MJ, Litster JD. KINETICS OF IMMERSION NUCLEATION DRIVEN BY SURFACE TENSION. In: *7th International Granulation Workshop.*; 2015.
27. Eshtiaghi N, Arhatari B, Hapgood KP. Producing hollow granules from hydrophobic powders in high-shear mixer granulators. *Adv Powder Technol.* 2009;20(6):558–566. doi:10.1016/j.apr.2009.08.006.
28. Eshtiaghi N, Liu JJS, Hapgood KP. Formation of hollow granules from liquid marbles: Small scale experiments. *Powder Technol.* 2010;197(3):184–195. doi:10.1016/j.powtec.2009.09.013.



29. Eshtiaghi N, Hapgood KP. A quantitative framework for the formation of liquid marbles and hollow granules from hydrophobic powders. *Powder Technol.* 2012;223:65–76. doi:10.1016/j.powtec.2011.05.007.
30. van den Dries K, Vromans H. Quantitative proof of liquid penetration-involved granule formation in a high shear mixer. *Powder Technol.* 2009;189(2):165–171. doi:10.1016/j.powtec.2008.04.007.
31. Rough SL, Wilson DI, York DW. Effect of solids formulation on the manufacture of high shear mixer agglomerates. *Adv Powder Technol.* 2005;16(2):145–169. doi:10.1163/1568552053621704.
32. Rough S., Wilson D., Bayly a, York D. Tapping characterisation of high shear mixer agglomerates made with ultra-high viscosity binders. *Powder Technol.* 2003;132(2-3):249–266. doi:10.1016/S0032-5910(03)00075-5.
33. Rough SL, Wilson DI, Bayly a. E, York DW. Mechanisms in high-viscosity immersion–granulation. *Chem Eng Sci.* 2005;60(14):3777–3793. doi:10.1016/j.ces.2005.02.013.
34. Shafiee Afarani M, Samimi A, Bahadori Yekta E. Synthesis of alumina granules by high shear mixer granulator: Processing and sintering. *Powder Technol.* 2013;237:32–40. doi:10.1016/j.powtec.2013.01.007.
35. Saleh K, Vialatte L, Guigon P. Wet granulation in a batch high shear mixer. *Chem Eng Sci.* 2005;60(14):3763–3775. doi:10.1016/j.ces.2005.02.006.
36. Hapgood KP, Khanmohammadi B. Granulation of hydrophobic powders. *Powder Technol.* 2009;189(2):253–262. doi:10.1016/j.powtec.2008.04.033.
37. Charles-Williams H, Wengeler R, Flore K, Feise H, Hounslow MJ, Salman AD. Granulation behaviour of increasingly hydrophobic mixtures. *Powder Technol.* 2012. doi:10.1016/j.powtec.2012.06.009.
38. Žižek K, Hraste M, Gomzi Z. High shear granulation of dolomite – I: Effect of shear regime on process kinetics. *Chem Eng Res Des.* 2012:1–17. doi:10.1016/j.cherd.2012.06.014.

39. Herd AC. Induction Periods in the Granulation kinetics of Very Fine Powders. *Powder Technol.* 1979;24:103–104.
40. Johansen a, Schaefer T. Effects of physical properties of powder particles on binder liquid requirement and agglomerate growth mechanisms in a high shear mixer. *Eur J Pharm Sci.* 2001;14(2):135–47. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/11500259>.
41. Fu J, Adams MJ, Reynolds GK, Salman a. D, Hounslow MJ. Impact deformation and rebound of wet granules. *Powder Technol.* 2004;140(3):248–257. doi:10.1016/j.powtec.2004.01.012.
42. Eckhard S, Nebelung M. Investigations of the correlation between granule structure and deformation behavior. *Powder Technol.* 2011;206(1-2):79–87. doi:10.1016/j.powtec.2010.06.002.
43. Pagnoux C, Tessier-Doyen N, Pringuet A, Cerbelaud M, Garcia-Perez P. Influence of the suspension flocculated state on the microstructure of alumina spheres elaborated by colloidal granulation. *J Eur Ceram Soc.* 2009;29(8):1379–1385. doi:10.1016/j.jeurceramsoc.2008.09.007.
44. Wang Z, Kwauk M, Li H. Fluidization of fine particles. *Chem Eng Sci.* 1998;53(3):377–395.
45. Xu C, Zhu J. Experimental and theoretical study on the agglomeration arising from fluidization of cohesive particles—effects of mechanical vibration. *Chem Eng Sci.* 2005;60(23):6529–6541. doi:10.1016/j.ces.2005.05.062.
46. Xu C, Zhu J. Parametric study of fine particle fluidization under mechanical vibration. *Powder Technol.* 2006;161(2):135–144. doi:10.1016/j.powtec.2005.10.002.
47. Zhou L, Wang H, Zhou T, Li K, Kage H, Mawatari Y. Model of estimating nano-particle agglomerate sizes in a vibro-fluidized bed. *Adv Powder Technol.* 2013;24(1):311–316. doi:10.1016/j.appt.2012.08.002.

48. Mawatari Y, Tsunekawa M, Tatemoto Y, Noda K. Favorable vibrated fluidization conditions for cohesive fine particles. *Powder Technol.* 2005;154(1):54–60. doi:10.1016/j.powtec.2005.04.026.
49. Liu Y-D, Kimura S. Fluidized-bed nitridation of fine silicon powder. *Powder Technol.* 1999;106(3):160–167. doi:10.1016/S0032-5910(99)00072-8.
50. Nakazato T, Kawashima J-I, Masagaki T, Kato K. Penetration of fine cohesive powders through a powder–particle fluidized bed. *Adv Powder Technol.* 2006;17(4):433–451. doi:10.1163/156855206777866128.
51. van Ommen JR, Valverde JM, Pfeffer R. Fluidization of nanopowders: a review. *J Nanopart Res.* 2012;14(3):737. doi:10.1007/s11051-012-0737-4.
52. Phillips D, Lannutti JJ. Measuring physical density with X-ray computed tomography. *NDT E Int.* 1997;30(6):339–350. Available at: <http://www.sciencedirect.com/science/article/pii/S0963869597000200>. Accessed March 30, 2012.
53. Ketcham R a., Carlson WD. Acquisition, optimization and interpretation of X-ray computed tomographic imagery: applications to the geosciences. *Comput Geosci.* 2001;27(4):381–400. doi:10.1016/S0098-3004(00)00116-3.
54. Barrett JF, Keat N. Artifacts in CT : Recognition and Avoidance. *Radio Graph.* 2004;24:1679–1691.
55. Yu L, Leng S. Image Reconstruction Techniques. *Image Wisely.* 2013. Available at: <http://www.imagewisely.org/imaging-modalities/computed-tomography/medical-physicists/articles/image-reconstruction-techniques>. Accessed March 15, 2013.
56. Jan ML, Chen CY, Yeh CK, Yeh TR, Wang MT. 3D image reconstruction using cone beam tomography. *NDT E Int.* 1994;27(2):83–87. doi:10.1016/0963-8695(94)90314-X.

57. Grass M, Kohler T, R. P. 3D cone-beam CT reconstructions for circular trajectories. *Phys Med Biol*. 2000;45(2):329–347. Available at: <http://www.ncbi.nlm.nih.gov/pubmed/17689223>.
58. Feldkamp LA, Davis LC, Kress JW. Practical cone-beam algorithm. *J Opt Soc Am A*. 1984;1(6):612–619.
59. Rahmanian N, Ghadiri M, Jia X, Stepanek F. Characterisation of granule structure and strength made in a high shear granulator. *Powder Technol*. 2009;192(2):184–194. doi:10.1016/j.powtec.2008.12.016.
60. Farber L, Tardos G, Michaels JN. Use of X-ray tomography to study the porosity and morphology of granules. *Powder Technol*. 2003;132(1):57–63. doi:10.1016/S0032-5910(03)00043-3.
61. Lindquist W, Venkatarangan A. Investigating 3D geometry of porous media from high resolution images. *Phys Chem Earth, Part ....* 1999;25(7):593–599. Available at: <http://www.sciencedirect.com/science/article/pii/S146418959900085X>. Accessed February 15, 2014.
62. Le PK, Avontuur P, Hounslow MJ, Salman AD. A microscopic study of granulation mechanisms and their effect on granule properties. *Powder Technol*. 2011;206(1-2):18–24. doi:10.1016/j.powtec.2010.06.014.
63. Nosrati A, Skinner W, Robinson DJ, Addai-Mensah J. Microstructure analysis of Ni laterite agglomerates for enhanced heap leaching. *Powder Technol*. 2012;232:106–112. doi:10.1016/j.powtec.2012.08.016.
64. Cottrino S, Jorand Y, Maire E, Adrien J. Characterization by X-ray tomography of granulated alumina powder during in situ die compaction. *Mater Charact*. 2013;81:111–123. doi:10.1016/j.matchar.2013.04.004.
65. Bhuiyan IU, Mouzon J, Forsberg F, Forsmo SPE, Sjödaahl M, Hedlund J. Consideration of X-ray microtomography to quantitatively determine the size distribution of bubble cavities in iron ore pellets. *Powder Technol*. 2013;233:312–318. doi:10.1016/j.powtec.2012.09.012.

66. Crean B, Parker A, Roux D Le, et al. Elucidation of the internal physical and chemical microstructure of pharmaceutical granules using X-ray micro-computed tomography, Raman microscopy and infrared spectroscopy. *Eur J Pharm Biopharm.* 2010;76(3):498–506. doi:10.1016/j.ejpb.2010.08.006.
67. Rab M a., Haling RE, Aarons SR, Hannah M, Young IM, Gibson D. Evaluation of X-ray computed tomography for quantifying macroporosity of loamy pasture soils. *Geoderma.* 2014;213:460–470. doi:10.1016/j.geoderma.2013.08.037.
68. Taud H, Martinez-Angeles R, Parrot JF, Hernandez-Escobedo L. Porosity estimation method by X-ray computed tomography. *J Pet Sci Eng.* 2005;47(3-4):209–217. doi:10.1016/j.petrol.2005.03.009.
69. Dal Ferro N, Delmas P, Duwig C, Simonetti G, Morari F. Coupling X-ray microtomography and mercury intrusion porosimetry to quantify aggregate structures of a cambisol under different fertilisation treatments. *Soil Tillage Res.* 2012;119:13–21. doi:10.1016/j.still.2011.12.001.
70. Dale SA, Dosta M, Antonyuk S, Wassgren C, Heinrich S, Litster JD. UNDERSTANDING THE INFLUENCE OF GRANULE MICROSTRUCTURE ON GRANULE COMPRESSION BREAKAGE. In: *7th International Granulation Workshop.*; 2015.
71. Plumb K. Continuous Processing in the Pharmaceutical Industry. *Chem Eng Res Des.* 2005;83(6):730–738. doi:10.1205/cherd.04359.
72. Engisch WE, Muzzio FJ. Feedrate deviations caused by hopper refill of loss-in-weight feeders. *Powder Technol.* 2015;283:389–400. doi:10.1016/j.powtec.2015.06.001.
73. Engisch WE, Muzzio FJ. Method for characterization of loss-in-weight feeder equipment. *Powder Technol.* 2012;228:395–403. doi:10.1016/j.powtec.2012.05.058.
74. Engisch WE, Muzzio FJ. Loss-in-Weight Feeding Trials Case Study: Pharmaceutical Formulation. *J Pharm Innov.* 2014;10(1):56–75. doi:10.1007/s12247-014-9206-1.

75. Cartwright JJ, Robertson J, D'Haene D, Burke MD, Hennenkamp JR. Twin screw wet granulation: Loss in weight feeding of a poorly flowing active pharmaceutical ingredient. *Powder Technol.* 2013;238:116–121. doi:10.1016/j.powtec.2012.04.034.
76. Meier R, Thommes M, Rasenack N, Moll KP, Kleinebudde P. MONO - MODAL PARTICLE SIZE DISTRIBUTIONS IN TWIN - SCREW GRANULATION - DO NOT FORGET THE FEEDING SYSTEMS. In: *7th International Granulation Workshop.*; 2015.
77. GmbH S. Particle Shape. Available at: [https://www.sympatec.com/EN/Science/Characterisation/05\\_ParticleShape.html](https://www.sympatec.com/EN/Science/Characterisation/05_ParticleShape.html). Accessed November 20, 2015.
78. Reynolds G, Fu J, Cheong Y, Hounslow M, Salman a. Breakage in granulation: A review. *Chem Eng Sci.* 2005;60(14):3969–3992. doi:10.1016/j.ces.2005.02.029.

## APPENDICES

## APPENDIX A: IMAGE ANALYSIS CODE

### Commentary

The following MATLAB code is used to process and create the results shown in this thesis. The functions “imFindlabels, imFeretDiameter, and imBoundingBox” were found on the Mathworks File Exchange. See code for copyright information of creator, David Legland. The following code sections were either created wholly by Steven Dale or have received light modification for use in this system: 9.6, 9.15, 9.16, 9.17, 9.19, 9.20, 9.22, 9.23, 9.25, 9.26, and 9.30. All other code used here was written by Nathan B. Davis or by Kelly Wang working as an undergraduate researcher.

### Run Codes

```
%user interface for image analysis of macro-voids within granules

%use label2rgb3dnew for MATLAB 2012 or later

s_repeat0 = ['Enter 1 to process raw images into binary images.\n',...
            'Enter 0 to skip.\n'];
repeat0 = input(s_repeat0);

if repeat0 == 1
    while repeat0 == 1
```



```
raw_binary_processing

s_repeat0 = ['\n\nEnter 1 to reprocess raw images.\n',...
            'Enter 0 to bypass:\n'];
repeat0 = input(s_repeat0);

end

end

s_repeat = ['\n\n\nEnter 1 to run Void_Wrapping_And_Measurements.\n'...
            'Enter 0 to skip:\n'];
repeat = input(s_repeat);

if repeat == 1
    while repeat == 1
        Void_Wrapping_And_Measurements
        s_repeat1 = ['\n\nEnter 1 to repeat non excluded analysis codes.'...
                    '\n\nEnter 0 to bypass:\n'];
        repeat = input(s_repeat1);
    end
end

end
```

```
repeat2 = 1;

while repeat2 == 1

    s_excluded = ['\n\nEnter 1 to run axial analysis code.'...

        '\n\nEnter 2 to run radial analysis code.'...

        '\n\nEnter 3 to run axial and radial analysis code.'...

        '\n\nEnter 0 to bypass:\n'];

    axial_radial = input(s_axial_radial);

    if axial_radial == 1

        Normalized_axial_radial_run_codes_auto(1)

    elseif axial_radial == 2

        Normalized_axial_radial_run_codes_auto(2)

    elseif axial_radial == 3

        Normalized_axial_radial_run_codes_auto(3)

    end

    excel_yn = input('Enter 1 to run excel exporter axial alone.\nEnter 2 to
run excel exporter axial and radial.\nEnter 0 to bypass and exit.\n');

    if excel_yn == 1

        excel_importer_axial_dist

    elseif excel_yn == 2

        excel_importer_axial_dist

        excel_importer_radial_distribution_auto

    end

end
```

```

elseif excel_yn == 3

    excel_importer_radial_distribution_auto

end

repeat2 = input('To run excluded or nonexcluded analysis code again, enter
1.\nTo continue, enter 0.\n');

end

```

Error using input

Cannot call INPUT from EVALC.

Error in run\_codes\_30micron\_limit (line 7)

```
repeat0 = input(s_repeat0);
```

Published with MATLAB® 7.14

## Raw\_Binary\_Processing

```

repeat0_error = 1; %used for error checking later

appenderror = 1;

while repeat0_error > 0 || appenderror > 0

n_ids = input('\n\nInput vector of granule IDs to be processed:\n');

read_error = 1;

while read_error > 0;

```

```

    % Test if user wants to change filename from default

s_fname = ['\nEnter 0 to keep file name as\n\n'...

'Nuclei_%d_thresholded_edited.tif, default from noise elimination
script.\n', ...

'\n Enter 1 to change file path. Use "/" and include file extensions:\n'];

fname_default = input(s_fname);

if fname_default == 0

    fname_infile = 'Nuclei_%d_thresholded_edited.tif';

else

    fname_infile = input('\nInput the file name of the edited photos:\n',
's');

end

    % Test if user wishes to change read path

s_test_path = ['\nEnter 1 to keep default path "./Edited_Granules/"',...

'\nEnter 2 if images are in same folder as MATLAB code (default from
noise elimination script).',...

'\nEnter 0 to specify path.\n'];

test_path = input(s_test_path);

if test_path == 1

    file_path = './Edited_Granules/';

elseif test_path == 2

    file_path = '';

else

    file_path = input('\nEnter folder path to image file:\n', 's');

```

```

end

    fname_plus_path = [file_path fname_infile]; % Puts image and read paths
    together

%     ensures specified path exists

    read_error = file_check(n_ids, fname_plus_path, 0);

end

%         s_fname = ['\n\nEnter image name, include folder path if image is not
%         saved to the current path,\n'...

%         'Use common slashes "/" and include the file extension:\n'];

%         s_granule_ids = ['Input vector of granule IDs to be processed:\n'];

%         fname_infile = input(s_fname, 's');

%         granule_ids = input(s_granule_ids);

    repeat0_error = file_check(n_ids, fname_infile, 0);

    fname1 = './Granule_%d_edited.tif';

    appenderror = file_check(n_ids, fname1, 1);

end

fprintf('\n\nAll files accounted for...\n\n');

for n = n_ids

```

```

    fname = sprintf(fname_infile,n);

    raw_image_noise_elimination(fname, n);

end

```

### Raw\_Image\_Noise\_Elimination

```

function [] = raw_image_noise_elimination_klw( fname,granule_number)

% Eliminates noise in raw, binary granule images.

%3-D watershed image segmentation method

[z_max, slice] = watershed_3D_raw_image_cleaner(fname,granule_number);

%Label pores

fname = sprintf('Raw_Granule_%d_connectivities.mat',granule_number);

load(fname)

[x_max,y_max,z_max] = size(L);

% % Following commented code is using too much memory for analyzing
% granules formed from 25 micron alumina. I am using a different method to
% evaluate connected objects, based on object volume, to see if that uses
% less memory.

```

```

% volume = x_max*y_max*z_max;

% L_1D = double(reshape(L, volume,1)); % Turn L into 1D vector, easier to
% %     since coordinate information is unnecessary in this case

% num_connections = max(L_1D);

%
%
% bins = num_connections + 1; %Must include zeros in air/particle counts
% connection_sizes = hist(L_1D,bins);

%
% mod_sizes = connection_sizes(2:end); %eliminate air count from vector
%
% [granule_size, granule_label] = max(mod_sizes);

% L = eq(L, granule_label);

% if isempty(slice) == 0

%     fprintf('The following slices were eliminated from image processing since
%     they contained all 0 values:\n')

%     disp(slice)

%     if length(slice) > 1

%         fprintf('\nWARNING: More than one slice eliminated. Check to make
%         sure image was properly extracted.\n')

%     end

%     z_max_new = slice(1) - 1;

% else

%     z_max_new = z_max;

% end

```

```
num_pores = max(L(:));

[x_max,y_max,z_max] = size(L);

num_pores = max(L(:));

pore_vol = zeros(num_pores,1);

weighted_sum_x = zeros(num_pores,1);

weighted_sum_y = zeros(num_pores,1);

weighted_sum_z = zeros(num_pores,1);

pore_COM = zeros(num_pores,3);

z_matrix = zeros(num_pores,z_max);

z_positions = cell(num_pores,1);

for z = 1:z_max

    for y = 1:y_max

        for x = 1:x_max

            if L(x,y,z) > 0

                pore_num = L(x,y,z);

                pore_vol(pore_num) = pore_vol(pore_num)+1;

                weighted_sum_x(pore_num) = weighted_sum_x(pore_num)+x;

                weighted_sum_y(pore_num) = weighted_sum_y(pore_num)+y;

                weighted_sum_z(pore_num) = weighted_sum_z(pore_num)+z;

                z_matrix(pore_num,z) = z;

            end

        end

    end

end
```



```

    end

    disp(z)
end

%Calculate COM and total volume in micron cubed from data

for i = 1:num_pores

    pore_COM(i,1) = weighted_sum_x(i)/pore_vol(i);

    pore_COM(i,2) = weighted_sum_y(i)/pore_vol(i);

    pore_COM(i,3) = weighted_sum_z(i)/pore_vol(i);

    z_positions{i} = unique(z_matrix(i,:));

    z_positions{i} = z_positions{i}(z_positions{i}>0);

    disp(i)
end

pore_vol = pore_vol.*6*6*6;

vol_equiv_sphere_dia = (pore_vol.*6./pi).^(1/3);

clear pore_vol z_matrix

clear pore_COM

clear weighted_sum_x

clear weighted_sum_y

clear weighted_sum_z

%thresh_dia sets equivalent volume sphere diameter for "exclusion" so that

%they won't be counted later. This is necessary because the ferets diameter
code

%otherwise takes too long to run. excluding items we know will already be

```

```

%removed later

thresh_dia = max(vol_equiv_sphere_dia);

false_labels = find(vol_equiv_sphere_dia < thresh_dia);

num_changes = length(false_labels);

for k=1:num_changes

    num_to_change=false_labels(k);

    z_max = max(z_positions{num_to_change});

    z_min = min(z_positions{num_to_change});

    for z = z_min:z_max

        A = L(:, :, z) ;

        positions = find(A == num_to_change);

        num_of_values = length(positions);

        for j=1:num_of_values

            A(positions(j))=0;

        end

        L(:, :, z) = A;

    end

end

for z = 1:z_max

    A = uint8(L(:, :, z));

    C = label2rgb3d_singleregion(A, 'gray', [1 1 1], 'noshuffle');

```

```

%     fname = sprintf('G:/MATLAB/Ferets Diameter
Testing/Nuclei_1/Excluded_RGB/rgb%d.tif',k);

    fname3 = sprintf('./Granule_%d_edited.tif',granule_number);

    BWIMAGE = im2uint8(im2bw(C));

    imwrite(BWIMAGE,fname3, 'tiff', 'writemode', 'append',
'compression','none');

    fprintf('writing BW_image %d/%d \n',z,z_max)

end

clear

end

```

### Watershed\_3D\_Raw\_Image\_Cleaner

```

function [z_max, slice] = watershed_3D_raw_image_cleaner( fname,granule_number
)

%3-D watershed image segmentation method

% Name tiff stack file

info = imfinfo(fname); %Retrieve image info

z_max = numel(info);%Name variable for number of images

z_max_check = z_max;

% Import image intensity values into matrix A and

% layer into 3D matrix im_3Dmatrix

A = imread (fname,1);

[x_max,y_max]= size(A);

```

```
im_3Dmatrix = uint32(zeros(x_max,y_max,z_max)); %preallocate space for matrix

xy_area = x_max*y_max;

num_slice = 0

for k = 1:z_max

    A = imread(fname,k);

    B = A == 0;

    %    B = im2bw(B);

    %    B = imcomplement(B);

    im_3Dmatrix (:,:,k)=B;

    disp(k)

    clear A

    im_2D = double(reshape(B, xy_area,1));

    air_vs_granule = hist(im_2D, 2);

    black_space = air_vs_granule(2);

    slice = find(black_space == xy_area);

    if length(slice) > 0

        z_max_check = z_max_check - 1;

        num_slice = num_slice+1

    end

    clear B

    clear im_2D

    clear air_vs_granule

    clear black_space
```

```

clear slice

end

% %      Following code is a high memory form of the code which does not work
with large files

% %      runs out of memory at im_2D (immediately below this), has been

% % transferred and reformatted within loop starting at line 16 to use less

% % memory

%      im_2D = double(reshape(im_3Dmatrix, xy_area, z_max));

%

%      % Counts white/black spaces per slice

%      air_vs_granule = hist(im_2D, 2);

%      black_space = air_vs_granule(2,:);

%      % Finds slices that are completely black, will prevent proper

%      % connectivity labeling in the granule

%      slice = find(black_space == xy_area);

%      if length(slice) > 0

%          z_max = slice(1) - 1;

%      else

%          z_max = z_max;

%      end

slice = num_slice;

im_3Dmatrix = im_3Dmatrix(:,:, 1:z_max);

cc = bwconncomp(im_3Dmatrix);

clear A

```

```

clear im_3Dmatrix

L = labelmatrix(cc);

clear cc

save(sprintf('Raw_Granule_%d_connectivities',granule_number),'L','-v7.3')

clear L

end

```

### Stack\_convex\_hulls\_3D

```

function [] = stack_convex_hulls_3D(fname,granule_number)

%UNTITLED4 Summary of this function goes here

% Detailed explanation goes here

info = iminfo(fname); %Retrieve image info

z_max = numel(info);%Name variable for number of images

% Import image intensity values into matrix A and invert black particles
% into white particles

x_max = info(1,1).Height;

y_max = info(1,1).Width;

im_3Dmatrix = uint8(zeros(x_max,y_max,z_max));

air_matrix = uint8(zeros(x_max,y_max,z_max));

%Change into a black and white image

for k = 1:z_max

    A = imread(fname,k);

    A = 255-A;

    A_1 = A>54;

    im_3Dmatrix(:,:,k) = A_1;

```

```

    fprintf('loading image %d/%d \n',k,z_max)

end

clear A_1

%xy plane

for k = 1:z_max

    A_2 = im_3Dmatrix(:, :, k);

    outline = bwconvhull(A_2, 'union', 4);

    air_matrix(:, :, k) = outline;

    fprintf('xy convex hull image %d/%d \n', k, z_max)

end

clear A_2

%yz plane

for k = 1:x_max

    A_3(:, :) = im_3Dmatrix(k, :, :);

    outline = bwconvhull(A_3, 'union', 4);

    outline = uint8(outline);

    a_1(:, :) = air_matrix(k, :, :);

    a_1 = a_1 + outline;

    air_matrix(k, :, :) = a_1;

    fprintf('yz convex hull image %d/%d \n', k, x_max)

end

clear A_3 a_1

%xz plane

for k = 1:y_max

    A_4(:, :) = im_3Dmatrix(:, k, :);

```

```

        outline = bwconvhull(A_4, 'union', 4);

        outline = uint8(outline);

        a_2(:, :) = air_matrix(:, k, :);

        a_2 = a_2 + outline;

        air_matrix(:, k, :) = a_2;

        fprintf('xz convex hull image %d/%d \n', k, y_max)

    end

clear A_4 a_2

air_matrix = (air_matrix == 3);

air_matrix = uint8(air_matrix);

air_matrix = air_matrix - im_3Dmatrix;

air_matrix = air_matrix.*55;

for k = 1:z_max

    A = imread(fname, k);

    A = 255-A;

    B = air_matrix(:, :, k);

    C = A + B;

    fname2 = sprintf('./Granule %d segmented.tif', granule_number);

    imwrite(C, fname2, 'writemode', 'append', 'compression', 'none');

    fprintf('writing image %d/%d \n', k, z_max)

end

end

```



## Void\_Wrapping\_And\_Measurements

### Contents

- [Inputs](#)
- [Run wrapped particle, only use after edited images are tested](#)

### Inputs

```
n_ids = input('\n\nInput vector of granule IDs to be processed:\n');

thresh_diameter = input('\n\nInput desired threshold diameter for Ferets
Diameter Measurements (value in voxels):\n');

read_error = 1;

while read_error > 0;

    % Test if user wants to change filename from default

    s_fname = ['\n\nEnter 0 to keep file name as\n\n'...
              'Granule_%d_edited.tif, default from noise elimination script.\n', ...
              '\n\n Enter 1 to change file path. Use "/" and include file extensions:\n'];

    fname_default = input(s_fname);

    if fname_default == 0

        fname_infile = 'Granule_%d_edited.tif';

    else

        fname_infile = input('\n\nInput the file name of the edited photos:\n',
's');
```

```
end

% Test if user wishes to change read path

s_test_path = ['\nEnter 1 to keep default path "./Edited_Granules/"',...
              '\nEnter 2 if images are in same folder as MATLAB code (default from
noise elimination script).',...
              '\nEnter 0 to specify path.\n'];

test_path = input(s_test_path);

if test_path == 1
    file_path = './Edited_Granules/';

elseif test_path == 2
    file_path = '';

else
    file_path = input('\nEnter folder path to image file:\n', 's');

end

fname_plus_path = [file_path fname_infile]; % Puts image and read paths
together

% ensures specified path exists

read_error = file_check(n_ids, fname_plus_path, 0);

end

save_error = 1;
```

```
while save_error > 0

    s_save_path_test = ['\nIf Nuclei_%d folder is in path, enter 1.\n',...
        'To specify path to Nuclei_%d folder, enter 0:\n'];

    save_path_test = input(s_save_path_test);

    if save_path_test == 1

        save_path = ['./Nuclei_%d/'];

    else

        s_savepartial = ['\nInput file path to Nuclei_%d folder, excluding
Nuclei_%d:\n',...
            'Use "/" and end with a slash.\n'];

        save_path_partial = input(s_savepartial, 's');

        save_path = [save_path_partial 'Nuclei_%d/'];

    end

    save_error = file_check(n_ids, save_path, 0);

end

test_yn = input('\nEnter 0 to run the wrapping algorithm alone.\nEnter 1 to see
more options.\n');

repeat = 1;

while repeat == 1

    if test_yn == 0

        for nuclei_number = n_ids
```

```

                                % INPUT 1, file # range (granule ID)

fname = sprintf(fname_plus_path ,nuclei_number);

                                %INPUT 2, file name as string

stack_convex_hulls_3D(fname,nuclei_number)

                                % Call function, output 'Granule %d
segmented.tif'

                                % Also prints num particles wrapped onscreen

clearvars -except fname_plus_path save_path n_ids thresh_diameter

end

repeat = input('Enter 1 to run wrapping algorithm again.\nEnter 0 to
continue.\n');

else

repeat = 0;

end

end

test_yn2 = input('Enter 0 to run the entire code (including wrapping).\nEnter 1
to run code without wrapping.\nEnter 2 to Ferets_Diameter_Measurement and
Image_Combiner only.\nEnter 3 to run void measurements, sphericity, and macro-
void image printing.\n');

% Wrapping

if test_yn2 == 0

for nuclei_number = n_ids

                                % INPUT 1, file # range (granule ID)

fname = sprintf(fname_plus_path ,nuclei_number);

                                %INPUT 2, file name as string

stack_convex_hulls_3D(fname,nuclei_number)

```

```

                                % Call function, output 'Granule %d segmented.tif'

                                % Also prints num particles wrapped onscreen

clearvars -except fname_plus_path save_path n_ids thresh_diameter
test_yn2

end

end

%Counts voxels/pores

if test_yn2 == 1 || test_yn2 == 0 || test_yn2 == 2

```

Run wrapped particle, only use after edited images are tested

#### Run Feret's Code

```

for nuclei_number = n_ids

    fname = sprintf('Granule %d segmented.tif',nuclei_number);

    Ferets_Diameter_Analysis_User_Input_Size(fname,thresh_diameter,nuclei_number)

    clearvars -except fname_plus_path save_path n_ids thresh_diameter
test_yn2

end

clearvars -except fname_plus_path save_path n_ids thresh_diameter test_yn2

for nuclei_number = n_ids

    fname = sprintf('Granule %d
min_Ferets_XYZ_Exclusion_BW.tif',nuclei_number);

    image_combiner(fname,nuclei_number)

    clearvars -except fname_plus_path save_path n_ids thresh_diameter
test_yn2

```

```
end

%

% clearvars -except fname_plus_path save_path n_ids thresh_diameter
test_yn2

end

%Calculates volume and center of mass for every particle in the
system

%Size of voxels are not accounted for so COM values correspond to pixels

if test_yn2 == 0 || test_yn2 == 1 || test_yn2 == 3

for nuclei_number = n_ids

fname = sprintf('Nuclei_%d_air_labels.tif',nuclei_number);

void_size_position_measurement(fname,nuclei_number)

clearvars -except fname_plus_path save_path n_ids thresh_diameter
test_yn2

end

clearvars -except fname_plus_path save_path n_ids thresh_diameter test_yn2

fprintf('Running_Surface_Area_Measurement_Code')
```

```

for nuclei_number = n_ids

    fname = sprintf('Nuclei_%d_air_labels.mat',nuclei_number);

    real_surface_area_measurement(fname,nuclei_number)

clearvars -except fname_plus_path save_path n_ids thresh_diameter test_yn2

end

clearvars -except fname_plus_path save_path n_ids thresh_diameter test_yn2

for nuclei_number = n_ids

    fname = sprintf('Nuclei_%d_real_surface_area.mat',nuclei_number);

    Equivalent_Surface_Area(fname,nuclei_number)

clearvars -except fname_plus_path save_path n_ids thresh_diameter test_yn2

end

clearvars -except fname_plus_path save_path n_ids thresh_diameter test_yn2

for nuclei_number = n_ids

    fname = sprintf('Nuclei_%d_air_labels.mat',nuclei_number);

    exclusion_to_RGB_images(fname,nuclei_number)

clearvars -except fname_plus_path save_path n_ids thresh_diameter
test_yn2

end

clearvars -except fname_plus_path save_path n_ids thresh_diameter test_yn2

for nuclei_number = n_ids

    fname = sprintf('Nuclei_%d_air_labels.mat',nuclei_number);

    voids_RGB_images(fname,nuclei_number)

```

```

        clearvars -except fname_plus_path save_path n_ids thresh_diameter
        test_yn2

    end

    clearvars -except fname_plus_path save_path n_ids thresh_diameter test_yn2

    for nuclei_number = n_ids

        fname =
        sprintf('Granule_%d_Combined_Image_Rewritten.tif',nuclei_number);

        phase_voxel_counts(fname,nuclei_number)

        clearvars -except fname_plus_path save_path n_ids thresh_diameter
        test_yn2

    end

    clearvars -except fname_plus_path save_path n_ids thresh_diameter test_yn2

%     clearvars -except fname_plus_path save_path n_ids thresh_diameter
test_yn2

%     end

%

end

```

### Ferets\_Diameter\_Analysis\_User\_Input\_Size

```

function [] = Ferets_Diameter_Analysis_User_Input_Size(
fname,thresh_diameter,granule_number )

%2-D Feret's Diameter segmentation code. Performs segmentation based on
%user input "thresh_diameter" to separate out macro-void objects within the
%granule of interest. a pre-cleaning of 3-D objects smaller than the
%threshold diameter is included (objects 1 voxel smaller than threshold

```



```
%diameter value).

% Feret's Diameter measurement is performed in XY plane and saved as
% intermediate image. It is rewritten into the YZ plane, measured, saved,
% then into the XZ plane, measured and saved. The image after processing
% in XZ Plane is rewritten into the original XY orientation for further
% processing.

%2-D Processing is done one image slice at a time to reduce memory load and
%increase Feret's diameter Run Time. Reading multiple slices for 2-D
%processing into Feret's calculation significantly increases run time.

% thresh_diameter = # of voxels, so can be used easily with various
% resolutions

% Name tiff stack file

fprintf('loading_Granule_%d',granule_number)

info = imfinfo(fname); %Retrieve image info

z_max = numel(info);%Name variable for number of images

% Import image intensity values into matrix A and
% layer into 3D matrix im_3Dmatrix

A = imread (fname,1);

[x_max,y_max]= size(A);

im_3Dmatrix = uint32(zeros(x_max,y_max,z_max)); %preallocate space for matrix
```



```
[x_max,y_max,z_max] = size(L);

num_pores = max(L(:));

pore_vol = zeros(num_pores,1);

weighted_sum_x = zeros(num_pores,1);

weighted_sum_y = zeros(num_pores,1);

weighted_sum_z = zeros(num_pores,1);

pore_COM = zeros(num_pores,3);

z_matrix = zeros(num_pores,z_max);

z_positions = cell(num_pores,1);

for z = 1:z_max

    for y = 1:y_max

        for x = 1:x_max

            if L(x,y,z) > 0

                pore_num = L(x,y,z);

                pore_vol(pore_num)= pore_vol(pore_num)+1;

                weighted_sum_x(pore_num)= weighted_sum_x(pore_num)+x;

                weighted_sum_y(pore_num)= weighted_sum_y(pore_num)+y;

                weighted_sum_z(pore_num)= weighted_sum_z(pore_num)+z;

                z_matrix(pore_num,z) = z;

            end

        end

    end

    disp(z)
```

```

end

%Calculate COM and total volume in micron cubed from data

for i = 1:num_pores

    pore_COM(i,1) = weighted_sum_x(i)/pore_vol(i);

    pore_COM(i,2) = weighted_sum_y(i)/pore_vol(i);

    pore_COM(i,3) = weighted_sum_z(i)/pore_vol(i);

    z_positions{i} = unique(z_matrix(i,:));

    z_positions{i} = z_positions{i}(z_positions{i}>0);

    disp(i)

end

pore_vol = pore_vol.*6*6*6;

vol_equiv_sphere_dia = (pore_vol.*6./pi).^(1/3);

clear vol_equiv_sphere_dia

clear z_matrix

clear pore_COM

clear weighted_sum_x

clear weighted_sum_y

clear weighted_sum_z

%thresh_diameter sets equivalent volume sphere diameter for "exclusion" so that

%they won't be counted later. This is necessary because the ferets diameter

code

%otherwise takes too long to run. excluding items we know will already be

%removed later

```

```
thresh_dia_pre_clean = thresh_diameter-1;

thresh_volume = thresh_dia_pre_clean.*6*6*6;

false_labels = find(pore_vol < thresh_volume);

num_changes = length(false_labels);

for k=1:num_changes

    num_to_change=false_labels(k);

    z_max = max(z_positions{num_to_change});

    z_min = min(z_positions{num_to_change});

    for z = z_min:z_max

        A = L(:, :, z) ;

        positions = find(A == num_to_change);

        num_of_values = length(positions);

        for j=1:num_of_values

            A(positions(j))=0;

        end

        L(:, :, z) = A;

    end

end

num_images = size(L,3);

% writes intermediate image directly into folder containing processing
% code. Intermediate images are necessary for memory purposes as Feret's
```

```

% Diameter codes takes too long to run if entire image stack (500+ slices)
is being

```

```

% held in memory.

% fname commented is a method for writing image to specific location

% Macro-Void = black = 0 in saved image

for k = k1:k2

    A = double(L(:,:,k));

    C = label2rgb3d_singleregion(A, 'gray', [1 1 1], 'noshuffle');

%     fname = sprintf('G:/MATLAB/Ferets Diameter
Testing/Nuclei_1/Excluded_RGB/rgb%d.tif',k);

    fname3 = sprintf('Granule %d excluded_2_BW.tif',granule_number);

    BWIMAGE = im2bw(C);

    imwrite(BWIMAGE, fname3, 'writemode', 'append', 'compression', 'none');

    fprintf('writing BW_image %d/%d \n', k, z_max)

    disp(k);

end

k1 = k1+k2_original;

    k2 = k2+k2_original;

clear L

end

clearvars -except granule_number thresh_diameter

%This section performs Feret's Diameter measurement and void removal in the
%XY plane, then saves as an intermediate image to a subfolder. Image is

```

```
%processed one image slice at a time to reduce memory hold. Feret's
%measurements are also saved in a subfolder.

fname = sprintf('Granule %d excluded_2_BW.tif',granule_number);

info = imfinfo(fname); %Retrieve image info

z_max = numel(info);%Name variable for number of images

% Import image intensity values into matrix A and
% layer into 3D matrix im_3Dmatrix

A = imread (fname,1);

[x_max,y_max]= size(A);

im_3Dmatrix = uint32(zeros(x_max,y_max,z_max)); %preallocate space for matrix

k_max=z_max

k1 = 1;

k2 = 10;

k2_original = k2;

track = k_max./k2

track_max = floor(track)

x = track_max.*k2_original

k_test = k_max-x

for i = 1:k_max

    disp (i)
```

```

A = imread (fname,i);

B = A == 0;

%   B = im2bw(B);

%   B = imcomplement(B);

im_3Dmatrix (:,:,i)=B;

clear B

clear A

cc = bwconncomp(im_3Dmatrix,8);

clear im_3Dmatrix

L = labelmatrix(cc);

clear cc

num_pores = max(L(:));

thetal = linspace(0,180,7);

FD = imFeretDiameter(L(:,:,i),thetal);

min_FD = min(FD,[],2);

save
(sprintf('./Nuclei_%d/Ferets_Values/Granule_%d_min_Ferets_XY.mat',granule_number,i),'min_FD');

clear FD

%obtaining total number of labeled segments

nLabels = max(L);

n = max(nLabels);

[x_max,y_max,z_max] = size(L);

num_pores = max(L(:));

pore_vol = zeros(num_pores,1);

```



```

weighted_sum_x = zeros(num_pores,1);

weighted_sum_y = zeros(num_pores,1);

weighted_sum_z = zeros(num_pores,1);

pore_COM = zeros(num_pores,3);

z_matrix = zeros(num_pores,z_max);

z_positions = cell(num_pores,1);

for z = 1:z_max

    for y = 1:y_max

        for x = 1:x_max

            if L(x,y,z) > 0

                pore_num = L(x,y,z);

                pore_vol(pore_num) = pore_vol(pore_num)+1;

                weighted_sum_x(pore_num) = weighted_sum_x(pore_num)+x;

                weighted_sum_y(pore_num) = weighted_sum_y(pore_num)+y;

                weighted_sum_z(pore_num) = weighted_sum_z(pore_num)+z;

                z_matrix(pore_num,z) = z;

            end

        end

    end

    disp(z)

end

%Calculate COM and total volume in micron cubed from data

for k = 1:num_pores

```

```

pore_COM(k,1) = weighted_sum_x(k)/pore_vol(k);

pore_COM(k,2) = weighted_sum_y(k)/pore_vol(k);

pore_COM(k,3) = weighted_sum_z(k)/pore_vol(k);

z_positions{k} = unique(z_matrix(k,:));

z_positions{k} = z_positions{k}(z_positions{k}>0);

disp(k)

end

pore_vol = pore_vol.*6*6*6;

vol_equiv_sphere_dia = (pore_vol.*6./pi).^(1/3);

clear pore_COM

clear pore_vol

%thresh_diameter = 5; can be used to hard_code thresholding limit

false_labels = find(min_FD < thresh_diameter);

num_changes = length(false_labels);

for k=1:num_changes

    num_to_change=false_labels(k);

    z_max = max(z_positions{num_to_change});

    z_min = min(z_positions{num_to_change});

    for z = z_min:z_max

        A = L(:, :, z) ;

        positions = find(A == num_to_change);

```

```

        num_of_values = length(positions);

        for j=1:num_of_values

            A(positions(j))=0;

        end

        L(:,:,z) = A;

    end

end

num_images = size(L,3);

A = double(L(:,:,i));

C = label2rgb3d_singleregion(A, 'gray', [1 1 1], 'noshuffle');

fname3 = sprintf('./Nuclei_%d/Intermediate_Ferets_Images/Granule %d
min_Ferets_XY_Exclusion_BW.tif', granule_number, granule_number);

BWIMAGE = im2bw(C);

imwrite(BWIMAGE, fname3, 'writemode', 'append', 'compression', 'none');

fprintf('writing BW_image %d/%d \n', i, z_max)

clear min_FD

end

clearvars -except granule_number thresh_diameter

fprintf('rewriting_into_YZ_Plane_for_Granule_%d', granule_number)

% This section rewrites the image into the YZ plane and performs Feret's

% Diameter measurement and void exclusion

fname = sprintf('./Nuclei_%d/Intermediate_Ferets_Images/Granule %d
min_Ferets_XY_Exclusion_BW.tif', granule_number, granule_number);

```

```

info = imfinfo(fname); %Retrieve image info

z_max = numel(info); %Name variable for number of images

% Import image intensity values into matrix A and
% layer into 3D matrix im_3Dmatrix

A = imread (fname,1);

[x_max,y_max]= size(A);

im_3Dmatrix = uint32(zeros(x_max,y_max,z_max)); %preallocate space for matrix

F = uint32(zeros(x_max,y_max,z_max));

k_max=z_max;

k1 = 1;

k2 = 10;

k2_original = k2;

k_final = x_max;

track = x_max./k2;

track_max = floor(track);

ref = track_max.*k2_original;

x1 = ref+1;

n_test = x_max-ref;

for k = 1:k_max

    A = imread (fname,k);

    B = A == 0;

    %    B = im2bw(B);

    %    B = imcomplement(B);

    im_3Dmatrix (:,:,k)=B;

```

```

end

clear B

clear A

%converts im_3Dmatrix into a YZ plane 2D set for labeling and Feret's
%diameter. It does exclusion of voids inside of it and then rewrites
%values L values for that (X,[:, :]) set into im_3Dmatrix. At the end of the
entire function,
%im_3Dmatrix is labeled and printed without having left its original
%orientation

for Loop = 1:k_final

    yz_rewrite(:, :) = im_3Dmatrix(Loop, :, :);

    yz_3Dmatrix(:, :) = yz_rewrite(:, :);

    clear yz_rewrite

    cc = bwconncomp(yz_3Dmatrix, 8);

    clear yz_3Dmatrix

    L = labelmatrix(cc);

    clear cc

    theta1 = linspace(0, 180, 7);

    FD = imFeretDiameter(L(:, :), theta1);

min_FD = min(FD, [], 2);

save
(sprintf('./Nuclei_%d/Ferets_Values/Granule_%d_min_Ferets_YZ.mat', granule_number, Loop), 'min_FD');

disp (Loop)

```

```
clear FD

nLabels = max(L);

n = max(nLabels);

[x_max,y_max,z_max] = size(L);

num_pores = max(L(:));

pore_vol = zeros(num_pores,1);

weighted_sum_x = zeros(num_pores,1);

weighted_sum_y = zeros(num_pores,1);

weighted_sum_z = zeros(num_pores,1);

pore_COM = zeros(num_pores,3);

z_matrix = zeros(num_pores,z_max);

z_positions = cell(num_pores,1);

for z = 1:z_max

    for y = 1:y_max

        for x = 1:x_max

            if L(x,y,z) > 0

                pore_num = L(x,y,z);

                pore_vol(pore_num)= pore_vol(pore_num)+1;

                weighted_sum_x(pore_num)= weighted_sum_x(pore_num)+x;

                weighted_sum_y(pore_num)= weighted_sum_y(pore_num)+y;
```

```

        weighted_sum_z(pore_num) = weighted_sum_z(pore_num) + z;

        z_matrix(pore_num, z) = z;

    end

end

end

% disp(z)

end

%Calculate COM and total volume in micron cubed from data

for k = 1:num_pores

    pore_COM(k,1) = weighted_sum_x(k)/pore_vol(k);

    pore_COM(k,2) = weighted_sum_y(k)/pore_vol(k);

    pore_COM(k,3) = weighted_sum_z(k)/pore_vol(k);

    z_positions{k} = unique(z_matrix(k,:));

    z_positions{k} = z_positions{k}(z_positions{k}>0);

    %disp(k)

end

pore_vol = pore_vol.*6*6*6;

vol_equiv_sphere_dia = (pore_vol.*6./pi).^(1/3);

%thresh_diameter = 5; can be used to hard code thresholding limit

false_labels = find(min_FD < thresh_diameter);

num_changes = length(false_labels);

```

```

for k=1:num_changes

    num_to_change=false_labels(k);

    z_max = max(z_positions{num_to_change});

    z_min = min(z_positions{num_to_change});

    for z = z_min:z_max

        A = L(:, :, z) ;

        positions = find(A == num_to_change);

            num_of_values = length(positions);

            for j=1:num_of_values

                A(positions(j))=0;

            end

            L(:, :, z) = A;

        end

    end

end

A = double(L(:, :));

C = label2rgb3d_singleregion(A, 'gray', [1 1 1], 'noshuffle');

fname3 = sprintf('./Nuclei_%d/Intermediate_Ferets_Images/Granule %d
min_Ferets_YZ_Exclusion_BW.tif', granule_number, granule_number);

BWIMAGE = im2bw(C);

imwrite(BWIMAGE, fname3, 'writemode', 'append', 'compression', 'none');

%fprintf('writing BW_image %d', n)

end

```



```
clearvars -except granule_number thresh_diameter

fprintf('rewriting_into_XZ_Plane_for_Granule_%d',granule_number)

% rewrites YZ image into ZX (or XZ) image and performs Feret's Diameter

% measurement and exclusion

fname = sprintf('./Nuclei %d/Intermediate_Ferets_Images/Granule %d
min_Ferets_YZ_Exclusion_BW.tif',granule_number,granule_number);

info = imfinfo(fname); %Retrieve image info

z_max = numel(info);%Name variable for number of images

% Import image intensity values into matrix A and

% layer into 3D matrix im_3Dmatrix

A = imread (fname,1);

[x_max,y_max]= size(A);

im_3Dmatrix = uint32(zeros(x_max,y_max,z_max)); %preallocate space for matrix

k_max=z_max;

k1 = 1;

k2 = 10;

k2_original = k2;

k_final = x_max;

track = x_max./k2;

track_max = floor(track);

ref = track_max.*k2_original;

x1 = ref+1;

n_test = x_max-ref;
```

```

for k = 1:k_max

    A = imread (fname,k);

    B = A == 0;

    %    B = im2bw(B);

    %    B = imcomplement(B);

    im_3Dmatrix (:,:,k)=B;

end

clear B

clear A

%converts im_3Dmatrix into a XZ plane 2D set for labeling and Feret's
%diameter from the earlier rewrite that changed it from (X,Y,Z) into (Y,Z,X).
It should be back into X,Y,Z then into X,Z,Y. It does exclusion of voids
inside of it and then rewrites

%values L values for that (X,::) set into im_3Dmatrix. At the end,

%im_3Dmatrix is labeled and printed without having left its original
%orientation

for Loop = 1:k_final

    xz_rewrite(:, :) = im_3Dmatrix(Loop, :, :);

    xz_3Dmatrix(:, :) = xz_rewrite(:, :);

    clear xz_rewrite

    cc = bwconncomp(xz_3Dmatrix,8);

    clear xz_3Dmatrix

    L = labelmatrix(cc);

```

```

clear cc

theta1 = linspace(0,180,7);

FD = imFeretDiameter(L(:,:,:),theta1);

min_FD = min(FD,[],2);

save
(sprintf('./Nuclei_%d/Ferets_Values/Granule_%d_min_Ferets_XZ.mat',granule_number,Loop),'min_FD');

disp (Loop)

clear FD

nLabels = max(L);

n = max(nLabels);

[x_max,y_max,z_max] = size(L);

num_pores = max(L(:));

pore_vol = zeros(num_pores,1);

weighted_sum_x = zeros(num_pores,1);

weighted_sum_y = zeros(num_pores,1);

weighted_sum_z = zeros(num_pores,1);

pore_COM = zeros(num_pores,3);

z_matrix = zeros(num_pores,z_max);

z_positions = cell(num_pores,1);

for z = 1:z_max

```

```

for y = 1:y_max
    for x = 1:x_max
        if L(x,y,z) > 0
            pore_num = L(x,y,z);
            pore_vol(pore_num) = pore_vol(pore_num)+1;
            weighted_sum_x(pore_num) = weighted_sum_x(pore_num)+x;
            weighted_sum_y(pore_num) = weighted_sum_y(pore_num)+y;
            weighted_sum_z(pore_num) = weighted_sum_z(pore_num)+z;
            z_matrix(pore_num,z) = z;
        end
    end
end

% disp(z)
end

%Calculate COM and total volume in micron cubed from data
for k = 1:num_pores
    pore_COM(k,1) = weighted_sum_x(k)/pore_vol(k);
    pore_COM(k,2) = weighted_sum_y(k)/pore_vol(k);
    pore_COM(k,3) = weighted_sum_z(k)/pore_vol(k);
    z_positions{k} = unique(z_matrix(k,:));
    z_positions{k} = z_positions{k}(z_positions{k}>0);
    %disp(k)
end

```

```
pore_vol = pore_vol.*6*6*6;

vol_equiv_sphere_dia = (pore_vol.*6./pi).^(1/3);

%thresh_diameter = 5; can be used to hard code thresholding diameter

false_labels = find(min_FD < thresh_diameter);

num_changes = length(false_labels);

for k=1:num_changes

    num_to_change=false_labels(k);

    z_max = max(z_positions{num_to_change});

    z_min = min(z_positions{num_to_change});

    for z = z_min:z_max

        A = L(:, :, z) ;

        positions = find(A == num_to_change);

            num_of_values = length(positions);

            for j=1:num_of_values

                A(positions(j))=0;

            end

            L(:, :, z) = A;

        end

    end

end

A = double(L(:, :));
```

```

C = label2rgb3d_singleregion(A, 'gray', [1 1 1], 'noshuffle');

fname3 = sprintf('./Nuclei %d/Intermediate_Ferets_Images/Granule %d
min_Ferets_XZ_Exclusion_BW.tif', granule_number, granule_number);

BWIMAGE = im2bw(C);

imwrite(BWIMAGE, fname3, 'writemode', 'append', 'compression', 'none');

fprintf('writing BW_image %d', n)

end

clearvars -except granule_number thresh_diameter

fprintf('rewriting_into_XY_Plane_for_Granule_%d', granule_number)

fname = sprintf('./Nuclei %d/Intermediate_Ferets_Images/Granule %d
min_Ferets_XZ_Exclusion_BW.tif', granule_number, granule_number);

info = imfinfo(fname); %Retrieve image info

z_max = numel(info); %Name variable for number of images

% Import image intensity values into matrix A and
% layer into 3D matrix im_3Dmatrix

A = imread (fname, 1);

[x_max, y_max] = size(A);

im_3Dmatrix = uint32(zeros(x_max, y_max, z_max)); %preallocate space for matrix

k_max = z_max;

```

```

k1 = 1;

k2 = 10;

k2_original = k2;

k_final = x_max;

track = x_max./k2;

track_max = floor(track);

ref = track_max.*k2_original;

x1 = ref+1;

n_test = x_max-ref;

for k = 1:k_max

    A = imread (fname,k);

    B = A == 0;

    %    B = im2bw(B);

    %    B = imcomplement(B);

    im_3Dmatrix (:,:,k)=B;

end

clear B

clear A

%converts im_3Dmatrix into a XY plane 2D set after all Feret's analysis

for Loop = 1:k_final

    xz_rewrite(:, :) = im_3Dmatrix(Loop, :, :);

    xz_3Dmatrix(:, :) = xz_rewrite(:, :);

    clear xz_rewrite

```

```

cc = bwconncomp(xz_3Dmatrix,8);

clear xz_3Dmatrix

L = labelmatrix(cc);

clear cc

disp(Loop)

A = double(L(:,:,));

C = label2rgb3d_singleregion(A,'gray',[1 1 1],'noshuffle');

fname3 = sprintf('Granule %d
min_Ferets_XYZ_Exclusion_BW.tif',granule_number);

BWIMAGE = im2bw(C);

imwrite(BWIMAGE,fname3,'writemode','append','compression','none');

fprintf('writing BW_image %d',n)

end

end

```

### Image Combiner

```

function [] = image_combiner (fname,granule_number)

% Combines Macro-void output image with wrapped granule image (Granule %d
% segmented) to create a 3-phase image of macro-voids, pores, and particle
% matrix.

% The final image is "Granule_%d_Combined_Image_Rewritten.tif" so that the
% voxel values will be read properly by the axial/radial distributions

% Also creates the Macro-Void Label Matrix of connected objects. File is

```



```
% titled "Nuclei_%d_air_labels"

%fname=sprintf('Granule %d min_Ferets_XYZ_Exclusion_BW.tif',granule_number);

fname2 = sprintf('Granule %d segmented.tif',granule_number);

info = imfinfo(fname); %Retrieve image info

z_max = numel(info);%Name variable for number of images

disp(granule_number)

for k=1:z_max

    disp(k)

    A = imread(fname,k);

    bwimage = A == 0;

    Add = uint8(bwimage);

    B = Add.*255;

    C = imcomplement(B);

    D = C -230;

    G = imread(fname2,k);

    H = imcomplement(G);

    final = H - D;

    fname3 = sprintf('Granule_%d_Combined_image.tif',granule_number);

    imwrite(final,fname3,'writemode','append','compression','none');

end

clearvars -except granule_number
```

```
fname = sprintf('Granule_%d_Combined_image.tif',granule_number);

info = imfinfo(fname); %Retrieve image info

z_max = numel(info);%Name variable for number of images

disp(granule_number)

for k=1:z_max

    disp(k)

L = imread(fname,k);

positions_particle = find(L == 0);

num_changes = length(positions_particle);

for N = 1:num_changes

    L(positions_particle(N)) = 255;

end

clear positions_particle num_changes N

positions_macro_voids = find(L == 200);

num_changes = length(positions_macro_voids);

for N = 1:num_changes

    L(positions_macro_voids(N)) = 55;

end

positions_micro_voids = find(L == 175);
```

```

num_changes = length(positions_micro_voids);

for N = 1:num_changes

    L(positions_micro_voids(N)) = 140;

end

positions_external_spaces = find(L == 230);

num_changes = length(positions_external_spaces);

for N = 1:num_changes

    L(positions_external_spaces(N)) = 0;

end

fname3 = sprintf('Granule_%d_Combined_Image_Rewritten.tif',granule_number);

imwrite(L,fname3,'writemode','append','compression','none');

end

clearvars -except granule_number

%3-D watershed image segmentation method

% Name tiff stack file

fname4 =sprintf('Granule_%d_Combined_Image_Rewritten.tif',granule_number);

info = imfinfo(fname4); %Retrieve image info

z_max = numel(info);%Name variable for number of images

% Import image intensity values into matrix A and

% layer into 3D matrix im_3Dmatrix

A = imread (fname4,1);

```

```
[x_max,y_max]= size(A);

im_3Dmatrix = uint32(zeros(x_max,y_max,z_max)); %preallocate space for matrix
```

```
for k = 1:z_max

    A = imread (fname4,k);

    B = A == 55;

    %    B = im2bw(B);

    %    B = imcomplement(B);

    im_3Dmatrix (:,:,k)=B;

end

cc = bwconncomp(im_3Dmatrix);

clear A

clear im_3Dmatrix

L = labelmatrix(cc);

num_pores = max(L(:));

save(sprintf('Nuclei_%d_air_labels',granule_number),'L','-v7.3')

save(sprintf('Nuclei_%d_pore_numbers',granule_number),'num_pores')

clearvars except granule_number

end
```

## Void\_Size\_Position\_Measurement

```
function [] = void_size_position_measurement(fname,granule_number)

% Measures the volume and calculates the 3-D Center of Mass (COM) in X,Y,Z

% coordinates of the labeled macro-void objects

% Saves object volume, volume equivalent sphere diameter, COM, and z
% positions in separate files.

    %fname = sprintf('Nuclei_%d_air_labels.mat',granule_number);

    load (fname)

    [x_max,y_max,z_max] = size(L);

    num_pores = max(L(:));

    pore_vol = zeros(num_pores,1);

    weighted_sum_x = zeros(num_pores,1);

    weighted_sum_y = zeros(num_pores,1);

    weighted_sum_z = zeros(num_pores,1);

    pore_COM = zeros(num_pores,3);

    z_matrix = zeros(num_pores,z_max);

    z_positions = cell(num_pores,1);
```

```

for z = 1:z_max

    for y = 1:y_max

        for x = 1:x_max

            if L(x,y,z) > 0

                pore_num = L(x,y,z);

                pore_vol(pore_num)= pore_vol(pore_num)+1;

                weighted_sum_x(pore_num)= weighted_sum_x(pore_num)+x;

                weighted_sum_y(pore_num)= weighted_sum_y(pore_num)+y;

                weighted_sum_z(pore_num)= weighted_sum_z(pore_num)+z;

                z_matrix(pore_num,z) = z;

            end

        end

    end

    disp(z)

end

%Calculate COM and total volume in micron cubed from data

for k = 1:num_pores

    pore_COM(k,1) = weighted_sum_x(k)/pore_vol(k);

    pore_COM(k,2) = weighted_sum_y(k)/pore_vol(k);

    pore_COM(k,3) = weighted_sum_z(k)/pore_vol(k);

    z_positions{k} = unique(z_matrix(k,:));

    z_positions{k} = z_positions{k}(z_positions{k}>0);

    disp(k)

end

```

```

pore_vol = pore_vol.*6*6*6;

vol_equiv_sphere_dia = (pore_vol.*6./pi).^(1/3);

%Store variables in MATLAB cmd

save (sprintf('Nuclei_%d_num_pores-1',granule_number), 'num_pores')

save (sprintf('Nuclei_%d_pore_COM',granule_number), 'pore_COM')

save (sprintf('Nuclei_%d_pore_vol',granule_number), 'pore_vol')

save (sprintf('Nuclei_%d_vol_equiv_sphere_dia',granule_number),
'vol_equiv_sphere_dia')

save (sprintf('Nuclei_%d_z_positions',granule_number), 'z_positions')

clearvars -except fname_plus_path save_path n_ids test_yn2

end

```

### Real\_Surface\_Area\_Measurement

```

function [] = real_surface_area_measurement(fname,granule_number)

% Calculates surface area of labeled macro-void objects. Measurement is
% done by a brute force counting of exposed faces to voxels in the
% connected object. Only the appropriate z range is considered.

load (fname)

```

```
fname2 = sprintf('Nuclei_%d_vol_equiv_sphere_dia.mat',granule_number);

load (fname2)

fname3 = sprintf('Nuclei_%d_z_positions.mat',granule_number);

load (fname3)

low_thresh_dia = 10.5;

fname4 = sprintf('Nuclei_%d_num_pores-1.mat',granule_number);

load (fname4)

high_thresh_dia = 20000;

false_labels = find(vol_equiv_sphere_dia < high_thresh_dia);

num_changes = length(false_labels);

[x_max,y_max,z_max] = size(L);

disp(z_max)

void_surface_area = 0;

    for k=1:num_changes

        disp(k)

        surface_area = 0;

        non_exposed_faces = 0;

        exposed_surface_area = 0;

        num_to_change=false_labels(k);

        z_high = max(z_positions{num_to_change});

        z_low = min(z_positions{num_to_change});

        for z = z_low:z_high

            zabove = z+1;

            zbelow = z-1;

            A = L(:, :, z) ;
```



```
    if zbelow<1
        B = zeros(x_max,y_max,1);
    else
        B = L(:,zbelow);
    end

    if zabove>z_max
        C = zeros(x_max,y_max,1);
    else
        C = L(:,zabove);
    end

for x=1:x_max

    xbefore = x-1;
    xafter = x+1;

    Amid = A(x,:) == k;

    if x == 1
        Afirst = zeros(1,y_max);
    else
        Afirst = A(xbefore,:) == k;
    end

    if x == x_max
        Alast = zeros(1,y_max);
    else
        Alast = A(xafter,:) == k;
    end

    Bmid = B(x,:) == k;
```

```

    Cmid = C(x,:) == k;

    for y = 1:y_max

        ybefore = y-1;

        yafter = y+1;

        if Amid(1,y) == 0

            non_exposed_faces = non_exposed_faces+1;

        elseif Amid(1,y) == 1

            Faces_check =
Amid(1,ybefore)+Amid(1,yafter)+Afirst(1,y)+Alast(1,y)+Bmid(1,y)+Cmid(1,y);

            number_of_exposed_faces = 6-Faces_check;

            exposed_surface_area = number_of_exposed_faces.*6.*6;

            surface_area=surface_area+exposed_surface_area;

        end

    end

    end

    end

    void_surface_area(k,1) = surface_area;

    end

    save (sprintf('Nuclei_%d_real_surface_area',granule_number),
'void_surface_area')

    clearvars -except granule_number fname

end

```

## Equivalent\_Surface\_Area

```

function [] = Equivalent_Surface_Area(fname,granule_number)

%Calculates the sphericity of labeled macro void objects
%and sorting the data for void sizes and distributions,
%including the surface area distributions into a single file

load (fname)

fname2 = sprintf('Nuclei_%d_pore_vol.mat',granule_number);

load (fname2)

fname3 = sprintf('Nuclei_%d_vol_equiv_sphere_dia.mat',granule_number);

load (fname3)

S = ((36.*pi.*(pore_vol.^2)).^(1/3))./void_surface_area;

fname4 = sprintf('Nuclei_%d_pore_COM.mat',granule_number);

load (fname4)

sphericity(:,1) = S;

sphericity(:,2) = pore_COM(:,1);

sphericity(:,3) = pore_COM(:,2);

sphericity(:,4) = pore_COM(:,3);

sphericity(:,5) = pore_vol(:,1);

sphericity(:,6) = void_surface_area(:,1);

sphericity(:,7) = vol_equiv_sphere_dia(:,1);

save(sprintf('Nuclei_%d_Sphericity_Values',granule_number),'sphericity');

clear

end

```

## Exclusion\_To\_RGB\_Images

```

function [] = exclusion_to_RGB_images(fname,granule_number)

%Removes labeled void objects with sphericity values less than 0.1, which
%have been identified as non-macro-void objects. Afterwards, the code will
%generate a series of RGB tiff images.

% The exclusion limit can be easily changed to remove objects based on
% other parameters or values as desired (change low_thresh_dia and
% false_labels definitions)

load(fname)

fname2 = sprintf('Nuclei_%d_Sphericity_Values.mat',granule_number);

load (fname2)

fname3 = sprintf('Nuclei_%d_z_positions.mat',granule_number);

load (fname3)

low_thresh_dia = 0.1;

false_labels = find(sphericity(:,1) < low_thresh_dia);

num_changes = length(false_labels);

for k=1:num_changes

    num_to_change=false_labels(k);

    z_max = max(z_positions{num_to_change});

    z_min = min(z_positions{num_to_change});

    for z = z_min:z_max

        A = L(:, :, z) ;

```

```

        positions = find(A == num_to_change);

        num_of_values = length(positions);

        for j=1:num_of_values

            A(positions(j))=0;

        end

        L(:,:,z) = A;

    end

end

save(sprintf('Nuclei_%d_voids_excluded_air_labels-1',granule_number),'L','-v7.3')

clearvars -except granule_number

%Creating RGB images based on the excluded created voids

    fname = sprintf('Nuclei_%d_voids_excluded_air_labels-1.mat',granule_number);

    load (fname)

    num_images = size(L,3);

for k = 1:num_images

    A = double(L(:,:,k));

    B = label2rgb3dnew(A,'jet',[1 1 1],'shuffle');

    fname = sprintf('./Nuclei_%d/Excluded_RGB/rgb%d.tif',granule_number,k);

    imwrite(B,fname,'tif')

    disp(k);

end

clear

end

```

## Voids\_RGB\_Images

```

function [] = voids_RGB_images(fname,granule_number)

%converts labeled, non-excluded, void labeling into RGB images.

load (fname)

num_images = size(L,3);

for k = 1:num_images

    A = double(L(:,:,k));

    B = label2rgb3dnew(A, 'jet', [1 1 1], 'shuffle');

    fname1 = sprintf('./Nuclei_%d/RGB/rgb%d.tif',granule_number,k);

    imwrite(B,fname1, 'tif')

    disp(k);

end

clear

end

```

## Phase\_Voxel\_Counts

```

function [ ] = phase_voxel_counts(fname,granule_number, particle_density)

%Calculates and saves total number of voxels in the 3 phases, microvoid or
%"pore" macro-void, and particle.

%Calculation is also performed at the beginning of the Radial Distribution
%code, but it is performed separately here for analysis purposes

```

```
microvoid_label = 140;

macrovoid_lower = 55;

macro_void_upper = 56;

particle_label = 255;

microvoid_density = 0.001;

particle_density = 3.9;

macrovoid_density = 0.001;

info = imfinfo(fname);

% Define maximum for x, y, and z

z_max = numel(info); %Name variable for number of images

A = imread (fname,1);

[x_max,y_max]= size(A);

%Preallocate space for matrices that change in size

part_count = zeros(1,z_max);

microvoid_count = zeros(1,z_max);

macrovoid_count = zeros(1,z_max);

tot_b = zeros(1,z_max);

weighted_A_x = zeros(x_max,y_max);

weighted_A_y = zeros(x_max,y_max);
```

```
x_b = zeros(1,z_max);

y_b = zeros(1,z_max);

z_b = zeros(1,z_max);

% Make position matrices

x_position = (0:x_max-1).*6;

y_position = (0:y_max-1).*6;

z_position = (0:z_max-1).*6;

% Import images one at a time

for z = 1:z_max

    A = imread (fname,z);

    A = double(A);

    %Replace grey value intensity with density values and count number of
    %voxels for each phase

    for k = 1:(x_max*y_max)

        if A(k) == particle_label

            A(k) = particle_density;

            part_count(z) = part_count(z)+1 ;

        elseif A(k) == microvoid_label

            A(k) = microvoid_density;

            microvoid_count(z) = microvoid_count(z)+1;

        elseif A(k) == macrovoid_lower

            A(k) = macrovoid_density;

            macrovoid_count(z) = macrovoid_count(z)+1;
```



```

else A(k) = 0;

end

end

% Find COM of particle

tot_b(z) = sum(A(:));

z_b(z) = sum(sum(z_position(z).* A));

for x = 1:x_max

    weighted_A_x(x,:) = x_position(x) .* A(x,:);

end

x_b(z) = sum(weighted_A_x(:));

for y = 1:y_max

    weighted_A_y(:,y) = y_position(y) .* A(:,y);

end

y_b(z)=sum(weighted_A_y(:));

end

tot_macrovoid_vox = sum(macrovoid_count);

tot_part_vox = sum(part_count);

tot_microvoid_vox = sum(microvoid_count);

tot_den = sum(tot_b);

cum_x = sum(x_b);

cum_y = sum(y_b);

cum_z = sum(z_b);

```

```

x_COM = cum_x / tot_den;

y_COM = cum_y / tot_den;

z_COM = cum_z / tot_den;

COM = [x_COM, y_COM, z_COM];

save(sprintf('Granule_%d_tot_microvoid_vox_norm',granule_number),'tot_microvoid
_vox')

save(sprintf('Granule_%d_tot_macrovoid_vox_norm',granule_number),'tot_macrovoid
_vox')

save(sprintf('Granule_%d_tot_part_vox_norm',granule_number),'tot_part_vox')

end

```

### Label2rgb3dnew

```

function RGB = label2rgb3dnew(varargin)

%LABEL2RGB Convert label matrix to RGB image.

%   RGB = LABEL2RGB(L) converts a label matrix L, such as returned by
%   LABELMATRIX, BWLABEL, BWLABELN, or WATERSHED, into a color RGB image
%   for the purpose of visualizing the labeled regions.
%
%   RGB = LABEL2RGB(L, MAP) defines the colormap to be used in the RGB
%   image.  MAP can either be an n x 3 colormap matrix, a string containing
%   the name of a colormap function (such as 'jet' or 'gray'), or a
%   function handle of a colormap function (such as @jet or @gray).
%   LABEL2RGB evaluates MAP so that there is a different color for each
%   region in L.  If MAP is not specified, 'jet' is used as the default.

```

```

%
% RGB = LABEL2RGB(L, MAP, ZEROCOLOR) defines the RGB color of the
% elements labeled 0 in the input label matrix L. ZEROCOLOR can either
% be an RGB triple, or one of the following: 'y' (yellow), 'm',
% (magenta), 'c' (cyan), 'r'(red), 'g' (green), 'b' (blue), 'w' (white),
% or 'k' (black). If ZEROCOLOR is not specified, c[1 1 1] is used as the
% default.
%
% RGB = LABEL2RGB(L, MAP, ZEROCOLOR, ORDER), controls how colormap colors
% are assigned to regions in the label matrix. If ORDER is 'noshuffle'
% (the default), then colormap colors are assigned to the label matrix
% regions in numerical order. If ORDER is 'shuffle', then colormap
% colors are pseudorandomly shuffled.
%
% Class Support
% -----
% The input label matrix L can have any numeric class. It must contain
% finite nonnegative integers. RGB is uint8.
%
% Example 1
% -----
% %Use label2rgb to customize display of label matrix.
%
%     I = imread('rice.png');
%
%     figure, imshow(I)

```

```
%     BW = im2bw(I, graythresh(I));

%     CC = bwconncomp(BW);

%     L = labelmatrix(CC);

%     RGB = label2rgb(L);

%     RGB2 = label2rgb(L, 'spring', 'c', 'shuffle');

%     figure, imshow(RGB), figure, imshow(RGB2)

%

%     See also BWCONNCOMP,BWLABEL,COLORMAP,ISMEMBER,LABELMATRIX,WATERSHED.

[label,map,zerocolor,order,fcnflag] = parse_inputs(varargin{:});

% Determine the number of regions in the label matrix.

numregion = 15000;

% If MAP is a function, evaluate it. Make sure that the evaluated function
% returns a valid colormap.

if fcnflag == 1

    if numregion == 0

        cmap = [];

    else

        cmap = feval(map, numregion);

        if ~isreal(cmap) || any(cmap(:) > 1) || any(cmap(:) < 0) || ...
```

```

        ~isequal(size(cmap,2),3) || size(cmap,1) < 1

        error(message('images:label2rgb:functionReturnsInvalidColormap'));

    end

end

else

    cmap = map;

end

% If ORDER is set to 'shuffle', create a private stream with a fixed seed,
% which creates the same "random" permutation every time it is called.

if isequal(order,'shuffle')

    stream = RandStream('swb2712','seed',0);

    index = randperm(stream,numregion);

    cmap = cmap(index,:,:);

end

% Issue a warning if the zerocolor (boundary color) matches the color of one
% of the regions.

for i=1:numregion

    if isequal(zerocolor,cmap(i,:))

        warning(message('images:label2rgb:zerocolorSameAsRegionColor', i));

    end

end

cmap = [zerocolor;cmap];

```

```

if isa(label,'uint8') || isa(label,'uint16') || isa(label,'uint32')

    RGB = ind2rgb8(label, cmap);

else

    % Using label + 1 for two reasons: 1) IND2RGB and IND2RGB8 do not like
    % double arrays containing zero values, and 2) for non-double, IND2RGB would
    % cast to a double and do this.

    RGB = ind2rgb8(double(label)+1,cmap);

end

% Function: parse_inputs
% -----

function [L, Map, Zerocolor, Order, Fcnflag] = parse_inputs(varargin)

% L          label matrix: matrix containing non-negative values.

% Map       colormap: name of standard colormap, user-defined map, function
%           handle.

% Zerocolor  RGB triple or Colorspec

% Order     keyword if specified: 'shuffle' or 'noshuffle'

% Fcnflag   flag to indicating that Map is a function

narginchk(1,4);

% set defaults

L = varargin{1};

Map = 'jet';

```

```
Zerocolor = [1 1 1];

Order = 'noshuffle';

Fcnflag = 0;

% parse inputs

if nargin > 1

    Map = varargin{2};

end

if nargin > 2

    Zerocolor = varargin{3};

end

if nargin > 3

    Order = varargin{4};

end

% error checking for L

validateattributes(L,{'numeric','logical'}, ...

    {'real' '2d' 'nonsparse' 'finite' 'nonnegative' 'integer'}, ...

    mfilename,'L',1);

% error checking for Map

[fcn, fcnchk_msg] = fcnchk(Map);

if isempty(fcnchk_msg)

    Map = fcn;

    Fcnflag = 1;

else

    if isnumeric(Map)

        if ~isreal(Map) || any(Map(:) > 1) || any(Map(:) < 0) || ...
```

```

        ~isequal(size(Map,2), 3) || size(Map,1) < 1

        error(message('images:label2rgb:invalidColormap'));

    end

else

    error(fcnchk_msg);

end

end

% error checking for Zerocolor

if ~ischar(Zerocolor)

    % check if Zerocolor is a RGB triple

    if ~isreal(Zerocolor) || ~isequal(size(Zerocolor),[1 3]) || ...

        any(Zerocolor> 1) || any(Zerocolor < 0)

        error(message('images:label2rgb:invalidZerocolor'));

    end

else

    [cspec, msg] = cspecchk(Zerocolor);

    if ~isempty(msg)

        %message is translated at source.

        error(message('images:label2rgb:notInColorspec', msg))

    else

        Zerocolor = cspec;

    end

end

end

% error checking for Order

valid_order = {'shuffle', 'noshuffle'};

```



```

idx = strncmpi(Order, valid_order,length(Order));

if ~any(idx)

    error(message('images:label2rgb:invalidEntryForOrder'))

elseif nnz(idx) > 1

    error(message('images:label2rgb:ambiguousEntryForOrder', Order))

else

    Order = valid_order(idx);

end

```

---

Copyright 1993-2011 The MathWorks, Inc.  
Published with MATLAB® 7.14

### Jet\_Color\_Key

```

clear,clc

a = zeros(640,100);

c = zeros(640,100);

percent = 0:0.0818:5.1534;

for k = 1:2:63;

    a((k*10-9):(k*10+10),:) = k;

end

d = horzcat(c,a);

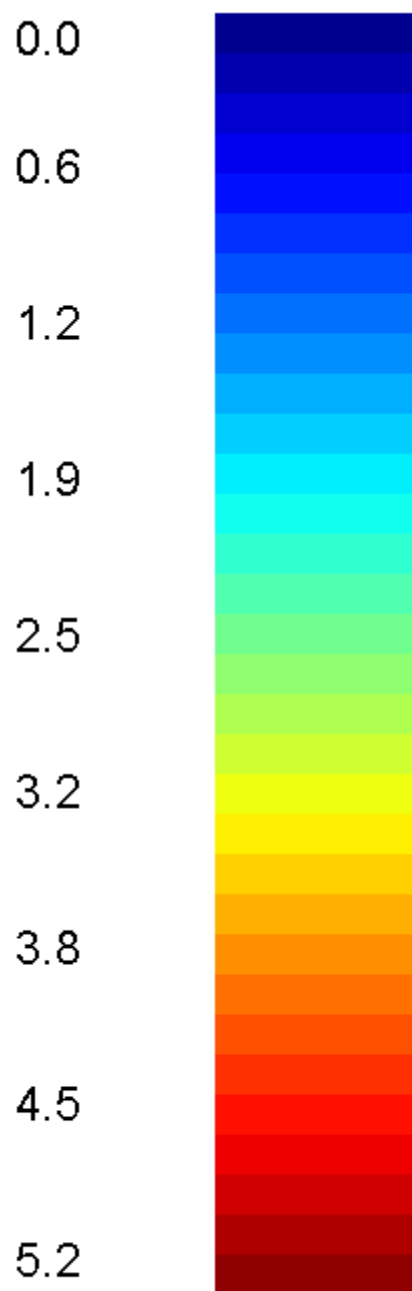
b = label2rgb(d,'jet','w','noshuffle');

label_1 = sprintf('%.1f',percent(1));

label_2 = sprintf('%.1f',percent(8));

```

```
label_3 = sprintf('%.1f',percent(16));  
label_4 = sprintf('%.1f',percent(24));  
label_5 = sprintf('%.1f',percent(32));  
label_6 = sprintf('%.1f',percent(40));  
label_7 = sprintf('%.1f',percent(48));  
label_8 = sprintf('%.1f',percent(56));  
label_9 = sprintf('%.1f',percent(64));  
  
imshow(b)  
  
hold on  
  
text(1,1+13,label_1,'FontSize',18);  
text(1,78,label_2,'FontSize',18);  
text(1,156,label_3,'FontSize',18);  
text(1,234,label_4,'FontSize',18);  
text(1,312,label_5,'FontSize',18);  
text(1,390,label_6,'FontSize',18);  
text(1,468,label_7,'FontSize',18);  
text(1,546,label_8,'FontSize',18);  
text(1,624,label_9,'FontSize',18);  
  
hold off  
  
e = getframe(gcf);  
  
imwrite(e.cdata,'Granule_9_edge_dist_jet_color_key.tif','compression','none')
```



## File\_Check

### Contents

- [Checks how many numbers need to be printed to file name, assume %d used](#)
- [Ensure file exists, prevents appending to preexisting file](#)

```
function error = file_check(IDs, fname0, read_or_append)

% Checks if file exists to ensure code can run without breaking

% read_or_append = 0 to see if MATLAB can find the file

% append_yn = 1 tp warn users that a file already exists
```

#### Checks how many numbers need to be printed to file name, assume %d used

specify the presence of a changing number, and assuming that number represents the granule number

```
num_IDs = length(find(fname0 == '%'));
printIDs = zeros(num_IDs, 1);
```

Error using file\_check (line 12)

Not enough input arguments.

#### Ensure file exists, prevents appending to preexisting file

```
if read_or_append == 0

    error = 0;

    for n = IDs

        printIDs(:,1) = n;

        fname = sprintf(fname0, printIDs);

        if exist(fname, 'file') == 0

            if error == 0
```

```

        fileerror = ['ERROR:',...
                    '\nThe following file(s) are entered incorrectly or
cannot be found:\n',...
                    '\n%s\n'];

    else

        fileerror = '%s\n';

    end

    fprintf(fileerror, fname);

    error = 1 + error;

end

end

elseif read_or_append == 1

    error = 0;

    num_IDs = length(find(fname0 == '%'));

    printIDs = zeros(num_IDs, 1);

    for n = IDs

        printIDs(:,1) = n;

        fname = sprintf(fname0, printIDs);

        fname_append = sprintf(fname, n);

        if exist(fname_append, 'file') ~= 0

            if error == 0

                s_appendererror = ['\n\nWARNING: The following file(s)
already exist in the directory\n'...

```

```

                                'and the following script is written to append to
existing files:\n',...

```

```

                                '%s\n'];

                                error = 1;

                                else

                                s_appenderror = '%s\n';

                                error = error + 1;

                                end

                                fprintf(s_appenderror, fname_append);

                                end

                                end

end

if error == 0

    fprintf('\nFile checking complete. No errors were found.\n');

end

end

```

### Excel\_importer\_radial\_distribution\_auto

```

% Radial Distribution Excel results

%Used in line with user interface.

% For current code design, Select 0 for user input. excluded data uses a
% different processing path and different file names

n_ids = input('\n\nInput vector of granule IDs to be processed:\n');

```

```

file_name = input('\n\nInput file name for radial distribution export:\n');

test = ['\nEnter 0 to import nonexcluded data.\n'...

        '\nEnter 1 to import excluded data.\n'];

axial_radial = input(test);

if axial_radial == 0

    s_ex = 'Granule';

else

    s_ex = 'Nuclei';

end

for gran_num = n_ids

    i = find(n_ids == gran_num);

    fprintf('Granule %d \n',gran_num)

    fname_bind = sprintf('%s_%d_rad_macro_counts_norm.mat',s_ex, gran_num);

    fname_part = sprintf('%s_%d_rad_part_counts_norm.mat',s_ex, gran_num);

    fname_air = sprintf('%s_%d_rad_micro_counts_norm.mat',s_ex, gran_num);

    %fname_tot_bind = sprintf('%s_%d_tot_bind_vox_50_bin_size.mat',s_ex,
gran_num);

    %fname_tot_air = sprintf('%s_%d_tot_air_vox_50_bin_size.mat',s_ex,
gran_num);

    %fname_tot_part =
sprintf('%s_%d_tot_part_vox_50_bin_size.mat',s_ex,gran_num);

    %fname_ellipsoid_radii = sprintf('%s_%d_ellipsoid_radii.mat',s_ex,
gran_num);

    load(fname_bind)

    load(fname_part)

```

```

load(fname_air)

%load(fname_tot_bind)

%load(fname_tot_air)

%load(fname_tot_part)

%load(fname_ellipsoid_radii)

%maj_dia = max(ellipsoid_radii(:))*6*2;

%min_dia = min(ellipsoid_radii(:))*6*2;

xlrage_bind = sprintf('C%d', ((i)*3));

xlrage_air = sprintf('C%d', ((i)*3+2));

xlrage_part = sprintf('C%d', ((i)*3+1));

xlrage_tot_bind = strcat(char('A'+(i)), '3');

xlrage_tot_part = strcat(char('A'+(i)), '4');

xlrage_tot_air = strcat(char('A'+(i)), '5');

%xlrage_ell_maj_dia = strcat(char('A'+(i)), '28');

%xlrage_ell_min_dia = strcat(char('A'+(i)), '27');

xlswrite(file_name, count_micro_norm_bin, 4, xlrage_bind)

xlswrite(file_name, count_macro_norm_bin, 4, xlrage_air)

xlswrite(file_name, count_part_norm_bin, 4, xlrage_part)

%   %xlswrite(file_name, tot_bind_vox, 2, xlrage_tot_bind)

%   xlswrite(file_name, tot_part_vox, 2, xlrage_tot_part)

%   xlswrite(file_name, tot_air_vox, 2, xlrage_tot_air)

%   xlswrite(file_name, maj_dia, 2, xlrage_ell_maj_dia)

%   xlswrite(file_name, min_dia, 2, xlrage_ell_min_dia)

end

```



## Excel\_Importer\_Axial\_Dist

```

% Axial Distribution Excel results

n_ids = input('Input vector of granule IDs to be processed:\n');

file_name_partial = input('Input granuel information, file will save as
Normalized_Imaging_Results_submic_[user input].xlsx\n','s');

test = ['\nEnter 0 to import nonexcluded data.\n'...
        '\nEnter 1 to import excluded data.\n'];

axial_radial = input(test);

if axial_radial == 0
    s_ex = 'Granule';
else
    s_ex = 'Nuclei';
end

for gran_num = n_ids
    i = find(n_ids == gran_num);

    file_name = ['./Normalized_Imaging_Results_submic_',file_name_partial];

    fprintf('Granule %d \n',gran_num)

    fname_bind = sprintf('%s_%d_ax_bind_counts_norm.mat',s_ex ,gran_num);

    fname_part = sprintf('%s_%d_ax_part_counts_norm.mat',s_ex ,gran_num);

    fname_air = sprintf('%s_%d_ax_air_counts_norm.mat',s_ex ,gran_num);

    %fname_tot_bind = sprintf('%s_%d_tot_bind_vox.mat',s_ex ,gran_num);

    %fname_tot_air = sprintf('%s_%d_tot_air_vox_50_bin_size.mat',s_ex
,gran_num);

```

```

    %fname_tot_part = sprintf('%s_%d_tot_part_vox_50_bin_size.mat',s_ex
,gran_num);

    fname_ellipsoid_radii = sprintf('%s_%d_ellipsoid_radii.mat',s_ex
,gran_num);

    load(fname_bind)

    load(fname_part)

    load(fname_air)

    %load(fname_tot_bind)

    %load(fname_tot_air)

    %load(fname_tot_part)

    load(fname_ellipsoid_radii)

    maj_dia = max(ellipsoid_radii(:))*6*2;
    min_dia = min(ellipsoid_radii(:))*6*2;
    xlrange_bind = sprintf('C%d',((i)*3));

%    error source?

    xlrange_air = sprintf('C%d',((i)*3+2));

%    error source?

    xlrange_part = sprintf('C%d',((i)*3+1));

    xlrange_tot_bind = strcat(char('A'+ i), '3');

    xlrange_tot_part = strcat(char('A'+(i)), '4');

    xlrange_tot_air = strcat(char('A'+(i)), '5');

    xlrange_ell_maj_dia = strcat(char('A'+(i)), '28');

    xlrange_ell_min_dia = strcat(char('A'+(i)), '27');

    xlswrite(file_name,norm_binder_count,3,xlrange_bind)

```

```
% %  
  
% % %      ??? Error using ==> xlswrite at 211  
  
% % % Excel returned: Error: Object returned error code: 0x800A03EC.  
  
% % %  
  
% % % Error in ==> excel_importer_axial_dist at 31  
  
% % %      xlswrite(file_name,norm_binder_count,3,xlrange_bind)  
  
% % %  
  
% % % Error in ==> run at 57  
  
% % %      evalin('caller', [s ';'']);  
  
% don't need binder?  
  
      xlswrite(file_name,norm_air_count,3,xlrange_air)  
  
%      same error,  
  
% xlrange_air =  
  
%  
  
% C-22  
  
      xlswrite(file_name,norm_particle_count,3,xlrange_part)  
  
      %xlswrite(file_name,tot_bind_vox,2,xlrange_tot_bind)  
  
      %xlswrite(file_name,tot_part_vox,2,xlrange_tot_part)  
  
      %xlswrite(file_name,tot_air_vox,2,xlrange_tot_air)  
  
      %xlswrite(file_name,maj_dia,2,xlrange_ell_maj_dia)  
  
      %xlswrite(file_name,min_dia,2,xlrange_ell_min_dia)  
  
end
```

## Excel\_Importer\_Sphericity\_Values

```

% set granule_number for desired file set for export.  set appropriate
% subtraction value such that gran_reference = 1:10. i.e. granule_number-x
% = 1:10 or appropriate for experimental set

for granule_number =21:25

    gran_reference = granule_number-20;

    file_name = 'Sphericity_Results_25mic_7PK90_5min_drum.xlsx';

    fprintf('Granule %d \n',granule_number)

    fname = sprintf('Nuclei_%d_Sphericity_Values.mat',granule_number);

    load (fname)

    i = gran_reference;

    fname2 = sprintf('Granule_%d_tot_macrovoid_vox_norm.mat',granule_number);

    fname3 = sprintf('Granule_%d_tot_microvoid_vox_norm.mat',granule_number);

    fname4 = sprintf('Granule_%d_tot_part_vox_norm.mat',granule_number);

    load (fname2)

    load (fname3)

    load (fname4)

    character_string = ['B','C','D','E','F','G','H','I','J','K'];

    character_string2 = ['B','D','F','H','J','L','N','P','R','T'];

    character_string3 = ['C','E','G','I','K','M','O','Q','S','U'];

        xrange_interest =
sprintf('%c2',character_string(gran_reference));

        xrange_interest2 = sprintf('%c2',character_string2(gran_reference));

```

```
xlrange_interest3 = sprintf('%c2',character_string3(gran_reference));  
xlrange_micro = sprintf('AC%d',((i)*3-1));  
xlrange_part = sprintf('AC%d',((i)*3));  
xlrange_macro = sprintf('AC%d',((i)*3+1));  
  
xlswrite(file_name,sphericity(:,1),1,xlrange_interest);  
xlswrite(file_name,sphericity(:,2),2,xlrange_interest);  
xlswrite(file_name,sphericity(:,3),3,xlrange_interest);  
xlswrite(file_name,sphericity(:,4),4,xlrange_interest);  
xlswrite(file_name,sphericity(:,5),5,xlrange_interest);  
xlswrite(file_name,sphericity(:,6),6,xlrange_interest);  
xlswrite(file_name,sphericity(:,7),7,xlrange_interest);  
xlswrite(file_name,sphericity(:,1),8,xlrange_interest2);  
xlswrite(file_name,sphericity(:,5),8,xlrange_interest3);  
xlswrite(file_name,sphericity(:,1),9,xlrange_interest2);  
xlswrite(file_name,sphericity(:,7),9,xlrange_interest3);  
xlswrite(file_name,tot_microvoid_vox,5,xlrange_micro);  
xlswrite(file_name,tot_part_vox,5,xlrange_part);  
xlswrite(file_name,tot_macrovoid_vox,5,xlrange_macro);  
  
clear  
  
end
```

## Cspecchk3d

```
function [cspec, msg] = cspecchk3d(varargin)

%
% CSPECCHK(VARARGIN) returns an RGB triple if VARARGIN is part of the
% ColorSpec or a valid RGB triple
%
% CSPECCHK is a helper function for LABEL2RGB and any other function that
% is creating a color image.
%
% [CSPEC, MSG] = CSPECCHK(varargin) returns an empty string in MSG if
% VARARGIN is part of the ColorSpec. Otherwise, CSPECCHK returns an error
% message string in MSG.
%
%
%
% error checking for nargin and setting defaults

error(nargchk(1, 1, nargin, 'struct'));

cspec = varargin{1};

msg = '';
```

```

% assigning colors to RGB triples.

yellow = [1 1 0];

magenta = [1 0 1];

cyan = [0 1 1];

red = [1 0 0];

green = [0 1 0];

blue = [0 0 1];

white = [1 1 1];

black = [0 0 0];

% making a table of cspec elements

cspec_el = {'yellow', yellow; 'magenta', magenta; 'cyan', cyan; 'red', ...
           red; 'green', green; 'blue', blue; 'white', white; 'k', black; ...
           'black', black};

if ~ischar(cspec)

    % check if cspec is a RGB triple

    if ~isreal(cspec) || ~isequal(size(cspec), [1 3]) || any(cspec > 1) || ...
        any(cspec < 0)

        msg = 'Invalid RGB triple entry for the ColorSpec.';

    end

else

    % check if cspec is part of cspec_el that defines the ColorSpec

    idx = strmatch(lower(cspec), cspec_el(:, 1));

    if isempty(idx)

```

```
    msg = sprintf('Entry is not part of the ColorSpec: %s.', cspec);  
  
elseif length(idx) > 1  
  
    % check if cspec equals 'b'. If yes then the cspec is blue.  
  
    % Otherwise, cspec is ambiguous.  
  
    if isequal(cspec, 'b')  
  
        cspec = blue;  
  
    else  
  
        msg = sprintf('Ambiguous entry for the ColorSpec: %s.', cspec);  
  
    end  
  
else  
  
    cspec = cspec_el{idx, 2};  
  
end  
  
end
```

Error using cspecchk3d (line 19)

Not enough input arguments.



## Ellipsoid\_fit

```

function [ center, radii, evecs, v ] = ellipsoid_fit( X, flag, equals )

%

% Fit an ellipsoid/sphere to a set of xyz data points:

%

% [center, radii, evecs, pars ] = ellipsoid_fit( X )
% [center, radii, evecs, pars ] = ellipsoid_fit( [x y z] );
% [center, radii, evecs, pars ] = ellipsoid_fit( X, 1 );
% [center, radii, evecs, pars ] = ellipsoid_fit( X, 2, 'xz' );
% [center, radii, evecs, pars ] = ellipsoid_fit( X, 3 );

%

% Parameters:

% * X, [x y z] - Cartesian data, n x 3 matrix or three n x 1 vectors
% * flag      - 0 fits an arbitrary ellipsoid (default),
%              - 1 fits an ellipsoid with its axes along [x y z] axes
%              - 2 followed by, say, 'xy' fits as 1 but also x_rad = y_rad
%              - 3 fits a sphere
%

% Output:

% * center    - ellipsoid center coordinates [xc; yc; zc]
% * ax        - ellipsoid radii [a; b; c]
% * evecs     - ellipsoid radii directions as columns of the 3x3 matrix
% * v         - the 9 parameters describing the ellipsoid algebraically:
%              Ax^2 + By^2 + Cz^2 + 2Dxy + 2Exz + 2Fyz + 2Gx + 2Hy + 2Iz = 1

```

```
%  
  
% Author:  
  
% Yury Petrov, Northeastern University, Boston, MA  
  
%  
  
error( nargchk( 1, 3, nargin ) ); % check input arguments  
  
if nargin == 1  
    flag = 0; % default to a free ellipsoid  
end  
  
if flag == 2 && nargin == 2  
    equals = 'xy';  
end  
  
if size( X, 2 ) ~= 3  
    error( 'Input data must have three columns!' );  
else  
    x = X( :, 1 );  
    y = X( :, 2 );  
    z = X( :, 3 );  
end  
  
% need nine or more data points  
  
if length( x ) < 9 && flag == 0  
    error( 'Must have at least 9 points to fit a unique ellipsoid' );  
end
```

```

end

if length( x ) < 6 && flag == 1

    error( 'Must have at least 6 points to fit a unique oriented ellipsoid' );

end

if length( x ) < 5 && flag == 2

    error( 'Must have at least 5 points to fit a unique oriented ellipsoid with
two axes equal' );

end

if length( x ) < 3 && flag == 3

    error( 'Must have at least 4 points to fit a unique sphere' );

end

end

if flag == 0

    % fit ellipsoid in the form Ax^2 + By^2 + Cz^2 + 2Dxy + 2Exz + 2Fyz + 2Gx +
2Hy + 2Iz = 1

    D = [ x .* x, ...

        y .* y, ...

        z .* z, ...

        2 * x .* y, ...

        2 * x .* z, ...

        2 * y .* z, ...

        2 * x, ...

        2 * y, ...

        2 * z ]; % ndatapoints x 9 ellipsoid parameters

elseif flag == 1

    % fit ellipsoid in the form Ax^2 + By^2 + Cz^2 + 2Gx + 2Hy + 2Iz = 1

```

```

D = [ x .* x, ...
      y .* y, ...
      z .* z, ...
      2 * x, ...
      2 * y, ...
      2 * z ]; % ndatapoints x 6 ellipsoid parameters

elseif flag == 2

% fit ellipsoid in the form Ax^2 + By^2 + Cz^2 + 2Gx + 2Hy + 2Iz = 1,
% where A = B or B = C or A = C

if strcmp( equals, 'yz' ) || strcmp( equals, 'zy' )

    D = [ y .* y + z .* z, ...
          x .* x, ...
          2 * x, ...
          2 * y, ...
          2 * z ];

elseif strcmp( equals, 'xz' ) || strcmp( equals, 'zx' )

    D = [ x .* x + z .* z, ...
          y .* y, ...
          2 * x, ...
          2 * y, ...
          2 * z ];

else

    D = [ x .* x + y .* y, ...
          z .* z, ...
          2 * x, ...

```

```

        2 * y, ...

        2 * z ];

end

else

    % fit sphere in the form A(x^2 + y^2 + z^2) + 2Gx + 2Hy + 2Iz = 1

    D = [ x .* x + y .* y + z .* z, ...

        2 * x, ...

        2 * y, ...

        2 * z ]; % ndatapoints x 4 sphere parameters

end

% solve the normal system of equations

v = ( D' * D ) \ ( D' * ones( size( x, 1 ), 1 ) );

% find the ellipsoid parameters

if flag == 0

    % form the algebraic form of the ellipsoid

    A = [ v(1) v(4) v(5) v(7); ...

        v(4) v(2) v(6) v(8); ...

        v(5) v(6) v(3) v(9); ...

        v(7) v(8) v(9) -1 ];

    % find the center of the ellipsoid

    center = -A( 1:3, 1:3 ) \ [ v(7); v(8); v(9) ];

    % form the corresponding translation matrix

    T = eye( 4 );

    T( 4, 1:3 ) = center';

```

```

% translate to the center

R = T * A * T';

% solve the eigenproblem

[ vecs evals ] = eig( R( 1:3, 1:3 ) / -R( 4, 4 ) );

radii = sqrt( 1 ./ diag( evals ) );

else

if flag == 1

    v = [ v(1) v(2) v(3) 0 0 0 v(4) v(5) v(6) ];

elseif flag == 2

    if strcmp( equals, 'xz' ) || strcmp( equals, 'zx' )

        v = [ v(1) v(2) v(1) 0 0 0 v(3) v(4) v(5) ];

    elseif strcmp( equals, 'yz' ) || strcmp( equals, 'zy' )

        v = [ v(2) v(1) v(1) 0 0 0 v(3) v(4) v(5) ];

    else % xy

        v = [ v(1) v(1) v(2) 0 0 0 v(3) v(4) v(5) ];

    end

else

    v = [ v(1) v(1) v(1) 0 0 0 v(2) v(3) v(4) ];

end

center = ( -v( 7:9 ) ./ v( 1:3 ) )';

gam = 1 + ( v(7)^2 / v(1) + v(8)^2 / v(2) + v(9)^2 / v(3) );

radii = ( sqrt( gam ./ v( 1:3 ) ) )';

vecs = eye( 3 );

end

```

## Normalized\_axial\_radial\_run\_codes

```
function [] = Normalized_axial_radial_run_codes_auto(axial_radial_test)

% Performs axial and radial analysis based on inputs. Axial analysis will
% not run in current format without running radial analysis.

n_ids = input('\n\nInput vector of granule IDs to be processed:\n');

s_density_test = ['\nEnter 0 to keep default particle density of 3.9.'...
    '\nEnter 1 to change particle density:\n'];

density_test = input(s_density_test);

if density_test == 0
    density = 3.9;
else
    density = input('\nEnter particle density: \n');
end

if axial_radial_test == 1
    % radial_distribution_normalized (and axial...) are used for non-excluded
    % granule images

    for nuclei_number = n_ids
```

```

        sprintf('Calculating Axial Distribution for Nuclei_%d',nuclei_number);

        fname =
sprintf('Granule_%d_combined_image_rewritten.tif',nuclei_number);

        axial_distribution_normalized(fname,nuclei_number)

    end

elseif axial_radial_test == 2

    for nuclei_number = n_ids

        sprintf('Calculating Radial Distribution for
Nuclei_%d',nuclei_number);

        fname =
sprintf('Granule_%d_combined_image_rewritten.tif',nuclei_number);

        radial_distribution_normalized(fname,nuclei_number, density)

    end

elseif axial_radial_test == 3

    for nuclei_number = n_ids

        sprintf('Calculating Radial Distribution for
Nuclei_%d',nuclei_number);

        fname =
sprintf('Granule_%d_combined_image_rewritten.tif',nuclei_number);

        radial_distribution_normalized(fname,nuclei_number, density)

    end

    for nuclei_number = n_ids

        sprintf('Calculating Axial Distribution for Nuclei_%d',nuclei_number);

        fname =
sprintf('Granule_%d_combined_image_rewritten.tif',nuclei_number);

```



```

        axial_distribution_normalized(fname,nuclei_number)

    end

else

    error = 1;

end

end
end

```

Error using input

Cannot call INPUT from EVALC.

Error in Normalized\_axial\_radial\_run\_codes\_auto (line 5)

```
n_ids = input('\n\nInput vector of granule IDs to be processed:\n');
```

---

*Published with MATLAB® 7.14*

### Radial\_distribution\_normalized

```

function [ ] = radial_distribution_normalized(fname,granule_number,
particle_density)

%UNTITLED3 Summary of this function goes here

% Detailed explanation goes here

binder_label = 140;

air_lower = 55;

air_upper = 56;

particle_label = 255;

```

```
binder_density = 0.001;

particle_density = 3.9;

air_density = 0.001;

info = imfinfo(fname);

% Define maximum for x, y, and z

z_max = numel(info); %Name variable for number of images

A = imread (fname,1);

[x_max,y_max]= size(A);

%Preallocate space for matrices that change in size

part_count = zeros(1,z_max);

bind_count = zeros(1,z_max);

air_count = zeros(1,z_max);

tot_b = zeros(1,z_max);

weighted_A_x = zeros(x_max,y_max);

weighted_A_y = zeros(x_max,y_max);

x_b = zeros(1,z_max);

y_b = zeros(1,z_max);

z_b = zeros(1,z_max);

% Make position matrices

x_position = (0:x_max-1).*6;
```

```

y_position = (0:y_max-1).*6;

z_position = (0:z_max-1).*6;

% Import images one at a time

for z = 1:z_max

    A = imread (fname,z);

    A = double(A);

    %Replace grey value intensity with density values and count number of
    %voxels for each phase

    for k = 1:(x_max*y_max)

        if A(k) == particle_label

            A(k) = particle_density;

            part_count(z) = part_count(z)+1 ;

        elseif A(k) == binder_label

            A(k) = binder_density;

            bind_count(z) = bind_count(z)+1;

        elseif A(k) == air_lower

            A(k) = air_density;

            air_count(z) = air_count(z)+1;

        else A(k) = 0;

        end

    end

end

% Find COM of particle

tot_b(z) = sum(A(:));

```

```

z_b(z) = sum(sum(z_position(z).* A));

for x = 1:x_max

    weighted_A_x(x,:) = x_position(x) .* A(x,:);

end

x_b(z) = sum(weighted_A_x(:));

for y = 1:y_max

    weighted_A_y(:,y) = y_position(y) .* A(:,y);

end

y_b(z)=sum(weighted_A_y(:));

end

tot_macro_vox = sum(air_count);

tot_part_vox = sum(part_count);

tot_micro_vox = sum(bind_count);

tot_den = sum(tot_b);

cum_x = sum(x_b);

cum_y = sum(y_b);

cum_z = sum(z_b);

x_COM = cum_x / tot_den;

y_COM = cum_y / tot_den;

z_COM = cum_z / tot_den;

COM = [x_COM, y_COM, z_COM];

% Radial distribution of all points from COM

```

```

% Preallocate space for radial distances

r_air = zeros(tot_macro_vox,1);

r_bind = zeros(tot_micro_vox,1);

r_part = zeros(tot_part_vox,1);

a = 1;

b = 1;

c = 1;

for z = 1:z_max

    A = imread (fname,z);

    A = double(A);

    for y = 1:y_max

        for x = 1:x_max

            if A(x,y) == binder_label

                r_bind(b) = sqrt((((x-1).*6)-COM(1)).^2 +...

                    (((y-1).*6)-COM(2)).^2 +...

                    (((z-1).*6)-COM(3)).^2);

                b = b+1;

            elseif A(x,y) == particle_label

                r_part(c) = sqrt((((x-1).*6)-COM(1)).^2 +...

                    (((y-1).*6)-COM(2)).^2 +...

                    (((z-1).*6)-COM(3)).^2);

                c = c+1;

            elseif A(x,y) == air_lower

                r_air(a) = sqrt((((x-1).*6)-COM(1)).^2 +...

                    (((y-1).*6)-COM(2)).^2 +...

```

```

((z-1).*6)-COM(3)).^2);

        a = a+1;

    end

end

end

end

r_step = 50; %binning distance in micron

num_bin = 130;

% Load radial distribution results and test for all radial distance

% inclusion

max_test = [max(r_air(:)),max(r_part(:))];

r_max = max(max_test);

if r_max > r_step * num_bin

    A = 0;

    disp ('Choose larger bin size or larger number of bins')

else disp('All data is accounted for')

    A = 1;

end

if A == 1; % Only ran if all data is accounted for

count_macro_bin = zeros(1, num_bin);

    for k = 0:num_bin -1 %1st bin from zero to r_step

        count_macro_bin(k+1)=length(find(r_air > r_step*k & r_air
<=r_step*(k+1)));

```

```

        if count_macro_bin(k+1)>50
            r_air_max_test = r_step*(k+1);
        end
    end

count_part_bin = zeros(1, num_bin);

    for k = 0:num_bin -1 %1st bin from zero to r_step
        count_part_bin(k+1) =length(find(r_part > r_step*k & r_part
<=r_step*(k+1)));
        count_part_bin(k+1) =length(find(r_part > r_step*k &
r_part <=r_step*(k+1)));
        if count_part_bin(k+1)>50
            r_part_max_test = r_step*(k+1);
        end
    end

count_micro_bin = zeros(1, num_bin);

    for k = 0:num_bin -1 %1st bin from zero to r_step
        count_micro_bin(k+1)=length(find (r_bind > r_step*k & r_bind
<=r_step*(k+1)));
    end

m = r_part<r_part_max_test;
n = r_air<r_air_max_test;

r_max_test = [max(r_air(n)),max(r_part(m)),max(r_bind)];

r_max = max(r_max_test);

r_air_norm = r_air./r_max;

```

```

clear r_air

r_part_norm = r_part./r_max;

clear r_part

r_bind_norm = r_bind./r_max;

clear r_bind

r_norm_step = 0.05;

count_micro_norm_bin = zeros(1,20);

count_part_norm_bin = zeros(1,20);

count_macro_norm_bin = zeros(1,20);

for k = 1:20

    count_micro_norm_bin(k) = length(find(r_bind_norm > r_norm_step * (k-1)
& r_bind_norm <= r_norm_step * k));

    count_part_norm_bin(k) = length(find(r_part_norm > r_norm_step * (k-1)
& r_part_norm <= r_norm_step * k));

    count_macro_norm_bin(k) = length(find(r_air_norm > r_norm_step * (k-1)
& r_air_norm <= r_norm_step * k));

end

end

save(sprintf('Granule_%d_tot_micro_vox_norm',granule_number),'tot_micro_vox')

save(sprintf('Granule_%d_tot_macro_vox_norm',granule_number),'tot_macro_vox')

save(sprintf('Granule_%d_tot_part_vox_norm',granule_number),'tot_part_vox')

save(sprintf('Granule_%d_rad_macro_counts_%d_rstep_norm',granule_number,r_step)
,'count_macro_bin')

```



```
save(sprintf('Granule_%d_rad_micro_counts_%d_rstep_norm',granule_number,r_step)
,'count_micro_bin')
```

```
save(sprintf('Granule_%d_rad_part_counts_%d_rstep_norm',granule_number,r_step),
'count_part_bin')
```

```
save(sprintf('Granule_%d_rad_macro_counts_norm',granule_number),'count_macro_no
rm_bin')
```

```
save(sprintf('Granule_%d_rad_micro_counts_norm',granule_number),'count_micro_no
rm_bin')
```

```
save(sprintf('Granule_%d_rad_part_counts_norm',granule_number),'count_part_norm
_bin')
```

```
end
```

Error using radial\_distribution\_normalized (line 13)

Not enough input arguments.

### Axial\_distribution\_normalized

```
function [ ] = axial_distribution_normalized(fname,granule_number)

%Fits ellipsoid to edge data then finds distribution along minor ellipsoid

%axis

% Detailed explanation goes here

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Find edge voxels x,y,z positions%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Get info from compiled tiff image

info = imfinfo(fname); %Retrieve image info

% Define image sizes

z_max = numel(info);%Name variable for number of images

A = imread (fname,1);
```

```

[x_max,y_max]= size(A);

%Prallocate space for matrices that change in size

edge_count = zeros(1,z_max);

%Find total number of edge voxels by finding perimeter of segmented
%images

for z = 1:z_max

    A = imread(fname,z);

    B = A~=0;

    C = bwperim(B);

    edge_count(z) = sum(C(:));

end

total_edge_vox = sum(edge_count(:));

%Prallocate space for edge position matrices

edge_x = zeros(1,total_edge_vox);

edge_y = zeros(1,total_edge_vox);

edge_z = zeros(1,total_edge_vox);

%Find edge pixel locations

k = 1;

for z = 1:z_max

    A = imread (fname,z);

    B = A~=0;

    C = bwperim(B);

    for x = 1:x_max

        for y = 1:y_max

            if C(x,y) > 0

```

```

        edge_x(k) = x;

        edge_y(k) = y;

        edge_z(k) = z;

        k = k+1;

    end

end

end

end

edge_positions = [edge_x',edge_y',edge_z'];

clear A B C x y z edge_x edge_y edge_z edge_count

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Fit ellipse to edge positions%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[ellipsoid_center, ellipsoid_radii,ellipsoid_radii_directions,v] = ...

    ellipsoid_fit(edge_positions);

ellipsoid_center = round(ellipsoid_center);

minor_axis = find(ellipsoid_radii==min(ellipsoid_radii));

direction_vector = ellipsoid_radii_directions(:,minor_axis)';%make row
vector

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Find distance projection of phase voxels onto new axis%%%%%%%%

%Load variables that were obtained in COM calculations

load(sprintf('Granule_%d_tot_air_vox_norm.mat',granule_number))

load(sprintf('Granule_%d_tot_bind_vox_norm.mat',granule_number))

load(sprintf('Granule_%d_tot_part_vox_norm.mat',granule_number))

%Preallocate arrays

binder_proj = zeros(tot_bind_vox,1);

```

```

particle_proj = zeros(tot_part_vox,1);

air_proj = zeros(tot_air_vox,1);

%Segregate voxels based on grey value intensity, define vector_a from
%center of ellipsoid to voxel position then find scalar projection of a
%onto ellipsoid minor axis for all voxels (direction is a unit vector)

b = 1;

c = 1;

d = 1;

for z = 1:z_max

    A = imread(fname,z);

    [x_values,y_values] = ind2sub(size(A),find(A>54));

    num_values = numel(x_values);

    for k = 1:num_values

        value = A(x_values(k),y_values(k));

        vector_a = [x_values(k)-ellipsoid_center(1),...

            y_values(k)-ellipsoid_center(2),...

            z-ellipsoid_center(3)].*6;

        scalar_projection = dot(vector_a,direction_vector);

        if value == 140

            binder_proj(b) = scalar_projection;

            b = b+1;

        elseif value == 255

            particle_proj(c) = scalar_projection;

            c = c+1;

        elseif value == 55

```

```

        air_proj(d) = scalar_projection;

        d = d+1;

    end

end

    fprintf('Dot products for Granule %d image %d/%d
\n',granule_number,z,z_max)

end

max_matrix = [max(particle_proj(:)),max(air_proj(:)),max(binder_proj(:))];
max_value = max(max_matrix);

min_matrix = [min(particle_proj(:)),min(air_proj(:)),min(binder_proj(:))];
min_value = min(min_matrix);

if max_value < abs(min_value) % Then positive axis is pointing towards top

    top_of_granule = max_value; %Max value is a positive value

    particle_proj = abs(particle_proj - top_of_granule);

    binder_proj = abs(binder_proj - top_of_granule);

    air_proj = abs(air_proj - top_of_granule);

elseif max_value > abs(min_value) %positive axis towards bottom of granule

    top_of_granule = min_value;%top of granule will be negative value

    particle_proj = particle_proj - top_of_granule;

    binder_proj = binder_proj - top_of_granule;

    air_proj = air_proj - top_of_granule;

else

```

```

        fprintf('error in calculation');
    end

    %Generate values for graphs by binning data

    num_bins = 130; %From 0 to 6500 microns

    bin_size = 50;

    particle_count = zeros(1,num_bins);

    air_count = zeros(1,num_bins);

    binder_count = zeros(1,num_bins);

    x_values = zeros(1,num_bins);

    for k = 0:num_bins-1

        x_min = k*bin_size;

        x_max = (k+1)*bin_size;

        x_values(k+1) = (x_min + x_max)/2;

        particle_count(k+1) = length(find(particle_proj >= x_min &
particle_proj < x_max));

        if particle_count(k+1)>50

            r_part_max_test = bin_size*(k+1);

        end

        air_count(k+1) = length(find(air_proj >= x_min & air_proj < x_max));

        if air_count(k+1)>50

            r_air_max_test = bin_size*(k+1);

        end

        binder_count(k+1) = length(find(binder_proj >= x_min & binder_proj <
x_max));

    end

```

```
m = particle_proj < r_part_max_test;

n = air_proj < r_air_max_test;

max_proj_test = [max(particle_proj(m)), max(air_proj(n)),
max(binder_proj)];

max_proj = max(max_proj_test);

norm_particle_proj = particle_proj ./ max_proj;

clear particle_proj

norm_binder_proj = binder_proj ./ max_proj;

clear binder_proj

norm_air_proj = air_proj ./ max_proj;

clear air_proj

norm_num_bins = 20;

norm_bin_size = 0.05;

norm_particle_count = zeros(1,20);

norm_binder_count = zeros(1,20);

norm_air_count = zeros(1,20);

for k = 0: norm_num_bins - 1

    x_min = k * norm_bin_size;

    x_max = (k + 1) * norm_bin_size;

    norm_particle_count(k + 1) = length(find(norm_particle_proj >= x_min &
norm_particle_proj < x_max));

    norm_air_count(k + 1) = length(find(norm_air_proj >= x_min &
norm_air_proj < x_max));
```

```

        norm_binder_count(k+1) = length(find(norm_binder_proj >= x_min &
norm_binder_proj < x_max));

    end

save(sprintf('Granule_%d_ax_macro_counts_%d_bin_size_for_norm',granule_number,b
in_size),'air_count')

save(sprintf('Granule_%d_ax_micro_counts_%d_bin_size_for_norm',granule_number,b
in_size),'binder_count')

save(sprintf('Granule_%d_ax_part_counts_%d_bin_size_for_norm',granule_number,bi
n_size),'particle_count')

    save(sprintf('Granule_%d_ellipsoid_radii',granule_number),
'ellipsoid_radii')

save(sprintf('Granule_%d_ax_macro_counts_norm',granule_number),'norm_air_count'
)

save(sprintf('Granule_%d_ax_micro_counts_norm',granule_number),'norm_binder_cou
nt')

save(sprintf('Granule_%d_ax_part_counts_norm',granule_number),'norm_particle_co
unt')

end

```

## imFeretDiameter

### Contents

- [Extract number of orientations](#)
- [Extract spatial calibration](#)
- [Initialisations](#)

```

function [fd, labels] = imFeretDiameter(img, varargin)

%Copyright (c) 2014, David Legland

% All rights reserved.

%

```



```
% Redistribution and use in source and binary forms, with or without
% modification, are permitted provided that the following conditions are
% met:
%
% * Redistributions of source code must retain the above copyright
% notice, this list of conditions and the following disclaimer.
% * Redistributions in binary form must reproduce the above copyright
% notice, this list of conditions and the following disclaimer in
% the documentation and/or other materials provided with the distribution
%
% THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
% AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
% IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
% ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
% LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
% CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
% SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
% INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
% CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
% ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
% POSSIBILITY OF SUCH DAMAGE.
%
%IMFERETDIAMETER Feret diameter of a particle(s) for a given direction(s)
%
% FD = imFeretDiameter(IMG, THETA);
```

```
% Compute the Feret diameter for particles in image IMG (binary or
% label), for the direction THETA, given in degrees.
% The result is a N-by-1 column vector, containing the Feret diameter of
% each particle in IMG.
%
% THETA can be a set of directions. In this case, the result has as many
% columns as the number of directions, and as many rows as the number of
% particles.
%
% FD = imFeretDiameter(IMG);
% Uses a default set of directions for computing Feret diameter.
%
% FD = imFeretDiameter(..., SPACING);
% Specifies the spatial calibration of image. SPACING = [SX SY] is a
% 1-by-2 row vector that contains the size of a pixel.
% Default spacing value is [1 1].
%
% FD = imFeretDiameter(..., SPACING, ORIGIN);
% Also specifies the position of the upper left pixel, as a 1-by-2 row
% vector.
%
% FD = imFeretDiameter(..., LABELS);
% Specifies the labels for which the Feret diameter should be computed.
% LABELS is a N-by-1 column vector. This can be used to save computation
% time when only few particles / regions are of interest within the
```

```
% entire image.
%
% [FD, LABELS] = imFeretDiameter(...);
% Also returns the set of labels that were considered for measure.
%
% The maximum Feret diameter can be obtained using a max() function.
%
% Example:
%
%   % compute Feret diameter for a discrete square
%
%   img = zeros(100, 100, 'uint8');
%
%   img(21:80, 21:80) = 1;
%
%   theta = linspace(0, 180, 201);
%
%   fd = imFeretDiameter(img, theta);
%
%   figure(1); clf; set(gca, 'fontsize', 14);
%
%   plot(theta, fd); xlim([0 180]);
%
%   xlabel('Angle (in degrees)');
%
%   ylabel('Diameter (in pixels)');
%
%   title('Feret diameter of discrete square');
%
%
% % max Feret diameter:
%
%   diam = max(fd, [], 2)
%
%   ans =
%
%       84.4386
%
%
% See also
```

```

% imOrientedBox

% -----

% Author: David Legland

% e-mail: david.legland@grignon.inra.fr

% Created: 2010-03-08,    using Matlab 7.9.0.529 (R2009b)

% Copyright 2010 INRA - Cepia Software Platform.

% HISTORY

% 2011-02-06 update doc, use convex hull, use degrees instead of radians

```

### Extract number of orientations

```

theta = 180;

if ~isempty(varargin)

    var1 = varargin{1};

    if isscalar(var1)

        % Number of directions given as scalar

        theta = var1;

        varargin(1) = [];

    elseif ndims(var1) == 2 && sum(size(var1) ~= [1 2]) ~= 0 %#ok<ISMAT>

        % direction set given as vector

        theta = var1;

        varargin(1) = [];

```

```
end  
  
end
```

### Extract spatial calibration

```
% default values  
  
spacing = [1 1];  
  
origin = [1 1];  
  
calib = false;  
  
  
% extract spacing  
  
if ~isempty(varargin) && sum(size(varargin{1}) == [1 2]) == 2  
  
    spacing = varargin{1};  
  
    varargin(1) = [];  
  
    calib = true;  
  
    origin = [0 0];  
  
end  
  
  
% extract origin  
  
if ~isempty(varargin) && sum(size(varargin{1}) == [1 2]) == 2  
  
    origin = varargin{1};  
  
end
```

## Initialisations

```
nTheta = length(theta);
```

```
% check if labels are specified
```

```
labels = [];
```

```
if ~isempty(varargin) && size(varargin{1}, 2) == 1
```

```
    labels = varargin{1};
```

```
end
```

```
% extract the set of labels, without the background
```

```
if isempty(labels)
```

```
    labels = imFindLabels(img);
```

```
end
```

```
nLabels = length(labels);
```

```
% allocate memory for result
```

```
fd = zeros(nLabels, nTheta);
```

```
for i = 1:nLabels
```

```
    % extract pixel centroids
```

```
    [y, x] = find(img==labels(i));
```

```
    if isempty(x)
```

```
        continue;

    end

    % transform to physical space if needed

    if calib

        x = (x-1) * spacing(1) + origin(1);

        y = (y-1) * spacing(2) + origin(2);

    end

    % keep only points of the convex hull

    try

        inds = convhull(x, y);

        x = x(inds);

        y = y(inds);

    catch ME %#ok<NASGU>

        % an exception can occur if points are colinear.

        % in this case we transform all points

    end

    % recenter points (should be better for numerical accuracy)

    x = x - mean(x);

    y = y - mean(y);

    % iterate over orientations
```

```
for t = 1:nTheta

    % convert angle to radians, and change sign (to make transformed
    % points aligned along x-axis)

    theta2 = -theta(t) * pi / 180;

    % compute only transformed x-coordinate

    x2 = x * cos(theta2) - y * sin(theta2);

    % compute diameter for extreme coordinates

    xmin = min(x2);

    xmax = max(x2);

    % store result (add 1 pixel to consider pixel width)

    dl = spacing(1) * abs(cos(theta2)) + spacing(2) * abs(sin(theta2));

    fd(i, t) = xmax - xmin + dl;

end

end
```



## imBoundingBox

### Contents

- [Initialisations](#)
- [Process planar case](#)
- [Process 3D case](#)

```
function [boxes labels] = imBoundingBox(img)

%Copyright (c) 2014, David Legland

% All rights reserved.

%

% Redistribution and use in source and binary forms, with or without
% modification, are permitted provided that the following conditions are
% met:
%
% * Redistributions of source code must retain the above copyright
%   notice, this list of conditions and the following disclaimer.
% * Redistributions in binary form must reproduce the above copyright
%   notice, this list of conditions and the following disclaimer in
%   the documentation and/or other materials provided with the distribution
%
% THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
% AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
% IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
% ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE
% LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
% CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
```

```
% SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
% INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
% CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
% ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
% POSSIBILITY OF SUCH DAMAGE.
```

```
%IMBOUNDINGBOX Bounding box of a binary or label image
```

```
%
```

```
% BOX = imBoundingBox(IMG)
```

```
% Compute the bounding boxes of the particles in labeled image IMG. If
```

```
% the image is binary, one box, corresponding to the foreground (i.e.
```

```
% the pixels with value 1) will be computed.
```

```
%
```

```
% The result is a N-by-4 array BOX = [XMIN XMAX YMIN YMAX], containing
```

```
% coordinates of the box extent.
```

```
%
```

```
% The same result could be obtained with the regionprops function. The
```

```
% advantage of using imBoundingBox is that equivalent boxes can be
```

```
% obtained in one call.
```

```
%
```

```
% BOX = imBoundingBox(IMG3D)
```

```
% If input image is a 3D array, the result is a N-by-6 array, containing
```

```
% the maximal coordinates in the X, Y and Z directions:
```

```
% BOX = [XMIN XMAX YMIN YMAX ZMIN ZMAX].
```

```
%  
  
%  
  
% Example  
  
% % Draw a complex particle together with its bounding box  
  
% img = imread('circles.png');  
  
% imshow(img); hold on;  
  
% boxes = imBoundingBox(img);  
  
% drawBox(boxes)  
  
%  
  
% % Compute and display the bounding box of several particles  
  
% img = imread('rice.png');  
  
% img2 = img - imopen(img, ones(30, 30));  
  
% lbl = bwlabel(img2 > 50, 4);  
  
% boxes = imBoundingBox(lbl);  
  
% imshow(img); hold on;  
  
% drawBox(boxes, 'linewidth', 2, 'color', 'g');  
  
%  
  
% See also  
  
% regionprops, drawBox, imOrientedBox, imInertiaEllipse  
  
%  
  
% -----  
  
% Author: David Legland  
  
% e-mail: david.legland@grignon.inra.fr  
  
% Created: 2011-03-30, using Matlab 7.9.0.529 (R2009b)  
  
% Copyright 2011 INRA - Cepia Software Platform.
```



```

ymax = max(y) + .5;

% create the resulting bounding box

boxes(i,:) = [xmin xmax ymin ymax];

end

elseif nd == 3

```

### Process 3D case

```

dim = size(img);

for i = 1:nLabels

% extract points of the current particle

inds = find(img==labels(i));

[y x z] = ind2sub(dim, inds);

% compute extreme coordinates, and add the half-width of the pixel

xmin = min(x) - .5;

xmax = max(x) + .5;

ymin = min(y) - .5;

ymax = max(y) + .5;

zmin = min(z) - .5;

zmax = max(z) + .5;

% create the resulting bounding box

boxes(i,:) = [xmin xmax ymin ymax zmin zmax];

end

else

```

```
error('Image dimension must be 2 or 3');  
  
end
```

## imFindLabels

```
function labels = imFindLabels(img)  
  
%Copyright (c) 2014, David Legland  
  
% All rights reserved.  
  
%  
  
% Redistribution and use in source and binary forms, with or without  
% modification, are permitted provided that the following conditions are  
% met:  
  
%  
% * Redistributions of source code must retain the above copyright  
% notice, this list of conditions and the following disclaimer.  
% * Redistributions in binary form must reproduce the above copyright  
% notice, this list of conditions and the following disclaimer in  
% the documentation and/or other materials provided with the distribution  
%  
  
% THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"  
% AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE  
% IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE  
% ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE  
% LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR  
% CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF  
% SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
```

```
% INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN
% CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
% ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
% POSSIBILITY OF SUCH DAMAGE.
```

```
%IMFINDLABELS Find unique labels in a label image
```

```
%
```

```
% LABELS = imFindLabels(IMG)
```

```
% Finds the unique labels in the label image IMG. The result can be
% obtained using the unique function, but a special processing is added
% to avoid using too much memory.
```

```
%
```

```
% Example
```

```
% imFindLabels
```

```
%
```

```
% See also
```

```
%
```

```
% -----
```

```
% Author: David Legland
```

```
% e-mail: david.legland@grignon.inra.fr
```

```
% Created: 2013-07-17, using Matlab 7.9.0.529 (R2009b)
```

```
% Copyright 2013 INRA - Cepia Software Platform.
```

```
if islogical(img)
```

```
    labels = 1;
```

```
    return;
end

if isfloat(img)
    labels = unique(img(:));
    labels(labels==0) = [];
    return;
end

maxLabel = double(max(img(:)));
labels = zeros(maxLabel, 1);
nLabels = 0;
for i = 1:maxLabel
    disp(i)
    if any(img(:) == i)
        nLabels = nLabels + 1;
        labels(nLabels) = i;
    end
end

labels = labels(1:nLabels);
```



## label2rgb3d\_singleregion

```
function RGB = label2rgb3d_singleregion(varargin)

%LABEL2RGB Convert label matrix to RGB image.

%   RGB = LABEL2RGB(L) converts a label matrix L, such as returned by
%   LABELMATRIX, BWLABEL, BWLABELN, or WATERSHED, into a color RGB image
%   for the purpose of visualizing the labeled regions.
%
%
%   RGB = LABEL2RGB(L, MAP) defines the colormap to be used in the RGB
%   image.  MAP can either be an n x 3 colormap matrix, a string containing
%   the name of a colormap function (such as 'jet' or 'gray'), or a
%   function handle of a colormap function (such as @jet or @gray).
%   LABEL2RGB evaluates MAP so that there is a different color for each
%   region in L.  If MAP is not specified, 'jet' is used as the default.
%
%
%   RGB = LABEL2RGB(L, MAP, ZEROCOLOR) defines the RGB color of the
%   elements labeled 0 in the input label matrix L.  ZEROCOLOR can either
%   be an RGB triple, or one of the following: 'y' (yellow), 'm',
%   (magenta), 'c' (cyan), 'r' (red), 'g' (green), 'b' (blue), 'w' (white),
%   or 'k' (black).  If ZEROCOLOR is not specified, c[1 1 1] is used as the
%   default.
%
%
%   RGB = LABEL2RGB(L, MAP, ZEROCOLOR, ORDER), controls how colormap colors
%   are assigned to regions in the label matrix.  If ORDER is 'noshuffle'
%   (the default), then colormap colors are assigned to the label matrix
```

```
% regions in numerical order.  If ORDER is 'shuffle', then colormap
% colors are pseudorandomly shuffled.
%
% Class Support
% -----
% The input label matrix L can have any numeric class. It must contain
% finite nonnegative integers.  RGB is uint8.
%
% Example 1
% -----
% %Use label2rgb to customize display of label matrix.
%
%     I = imread('rice.png');
%
%     figure, imshow(I)
%
%     BW = im2bw(I, graythresh(I));
%
%     CC = bwconncomp(BW);
%
%     L = labelmatrix(CC);
%
%     RGB = label2rgb(L);
%
%     RGB2 = label2rgb(L, 'spring', 'c', 'shuffle');
%
%     figure, imshow(RGB), figure, imshow(RGB2)
%
% See also BWCONNCOMP, BWLABEL, COLORMAP, ISMEMBER, LABELMATRIX, WATERSHED.
```

```
[label,map,zerocolor,order,fcnflag] = parse_inputs(varargin{:});

% Determine the number of regions in the label matrix.

numregion = 1;

% If MAP is a function, evaluate it. Make sure that the evaluated function
% returns a valid colormap.

if fcnflag == 1

    if numregion == 0

        cmap = [];

    else

        cmap = feval(map, numregion);

        if ~isreal(cmap) || any(cmap(:) > 1) || any(cmap(:) < 0) || ...
            ~isequal(size(cmap,2),3) || size(cmap,1) < 1

            error(message('images:label2rgb:functionReturnsInvalidColormap'));

        end

    end

else

    cmap = map;

end

% If ORDER is set to 'shuffle', create a private stream with a fixed seed,
% which creates the same "random" permutation every time it is called.
```

```

if isequal(order,'shuffle')

    stream = RandStream('swb2712','seed',0);

    index = randperm(stream,numregion);

    cmap = cmap(index,:,:);

end

% Issue a warning if the zerocolor (boundary color) matches the color of one
% of the regions.

for i=1:numregion

    if isequal(zerocolor,cmap(i,:))

        warning(message('images:label2rgb:zerocolorSameAsRegionColor', i));

    end

end

cmap = [zerocolor;cmap];

if isa(label,'uint8') || isa(label,'uint16') || isa(label,'uint32')

    RGB = ind2rgb8(label, cmap);

else

    % Using label + 1 for two reasons: 1) IND2RGB and IND2RGB8 do not like
    % double arrays containing zero values, and 2)for non-double, IND2RGB would
    % cast to a double and do this.

    RGB = ind2rgb8(double(label)+1,cmap);

end

% Function: parse_inputs

```

```
% -----  
  
function [L, Map, Zerocolor, Order, Fcnflag] = parse_inputs(varargin)  
  
% L          label matrix: matrix containing non-negative values.  
  
% Map       colormap: name of standard colormap, user-defined map, function  
  
%          handle.  
  
% Zerocolor RGB triple or Colorspec  
  
% Order     keyword if specified: 'shuffle' or 'nosshuffle'  
  
% Fcnflag   flag to indicating that Map is a function  
  
  
narginchk(1,4);  
  
% set defaults  
  
L = varargin{1};  
  
Map = 'jet';  
  
Zerocolor = [1 1 1];  
  
Order = 'nosshuffle';  
  
Fcnflag = 0;  
  
% parse inputs  
  
if nargin > 1  
    Map = varargin{2};  
  
end  
  
if nargin > 2  
    Zerocolor = varargin{3};  
  
end
```

```

end

if nargin > 3
    Order = varargin{4};
end

% error checking for L
validateattributes(L,{'numeric','logical'}, ...
    {'real' '2d' 'nonsparse' 'finite' 'nonnegative' 'integer'}, ...
    mfilename,'L',1);

% error checking for Map
[fcn, fcnchk_msg] = fcnchk(Map);

if isempty(fcnchk_msg)
    Map = fcn;
    Fcnflag = 1;
else
    if isnumeric(Map)
        if ~isreal(Map) || any(Map(:) > 1) || any(Map(:) < 0) || ...
            ~isequal(size(Map,2), 3) || size(Map,1) < 1
            error(message('images:label2rgb:invalidColormap'));
        end
    else
        error(fcnchk_msg);
    end
end

% error checking for Zerocolor
if ~ischar(Zerocolor)

```

```

% check if Zerocolor is a RGB triple

if ~isreal(Zerocolor) || ~isequal(size(Zerocolor),[1 3]) || ...
    any(Zerocolor> 1) || any(Zerocolor < 0)

    error(message('images:label2rgb:invalidZerocolor'));

end

else

[cspec, msg] = cspecchk(Zerocolor);

if ~isempty(msg)

    %message is translated at source.

    error(message('images:label2rgb:notInColorspec', msg))

else

    Zerocolor = cspec;

end

end

% error checking for Order

valid_order = {'shuffle', 'noshuffle'};

idx = strncmpi(Order, valid_order,length(Order));

if ~any(idx)

    error(message('images:label2rgb:invalidEntryForOrder'))

elseif nnz(idx) > 1

    error(message('images:label2rgb:ambiguousEntryForOrder', Order))

else

    Order = valid_order{idx};

end

```

## APPENDIX B: ADDITIONAL GRANULE IMAGES

Powder A1, A2, A3 images are found in Chapter 4.

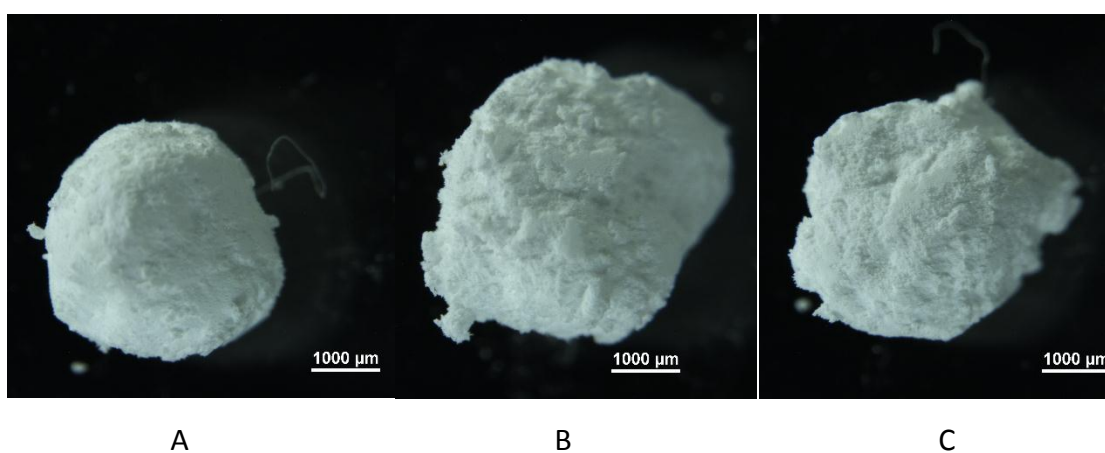


Figure 0.B.1: Optical Microscopy (A) Powder B1 (B) Powder B2 (C) Powder B3

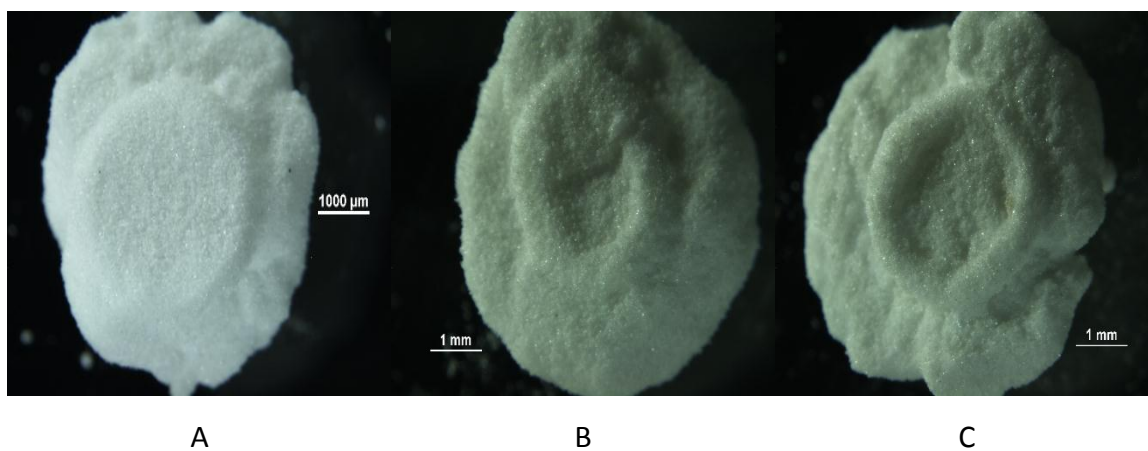
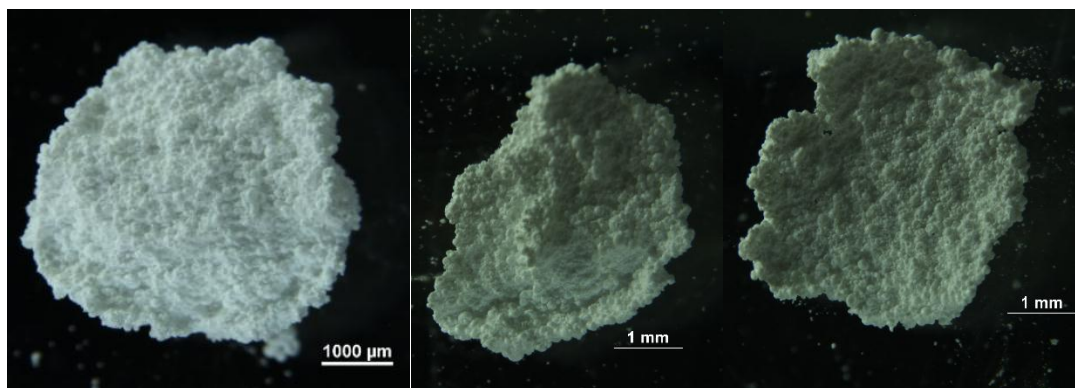


Figure 0.B.2: Optical Microscopy (A) Powder C1 (B) Powder C2 (C) Powder C3





A

B

C

Figure 0.B.3: Optical Microscopy (A) Powder D1 (B) Powder D2 (C) Powder D3

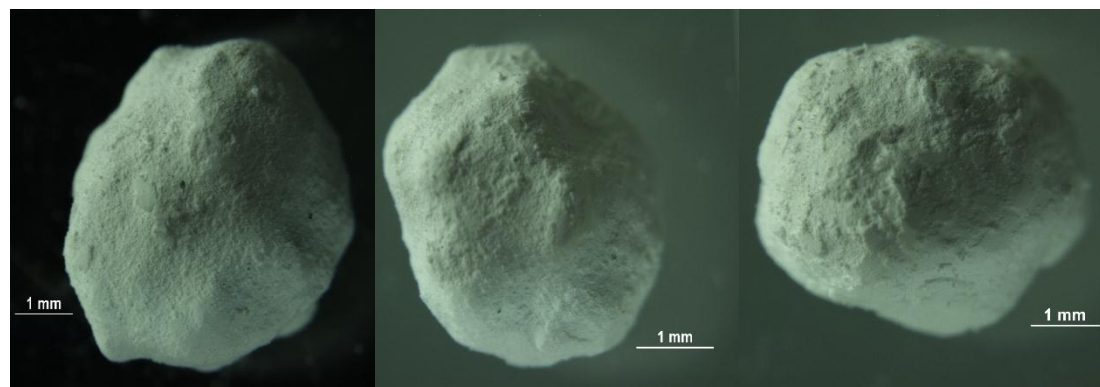


A

B

C

Figure 0.B.4: Optical Microscopy Powder A 3 seconds (A) 5.5 cP (B) 35 cP (C) 70 cP



A

B

C

Figure 0.B.5: Optical Microscopy Powder A 5 minutes (A) 5.5 cP (B) 35 cP (C) 70 cP

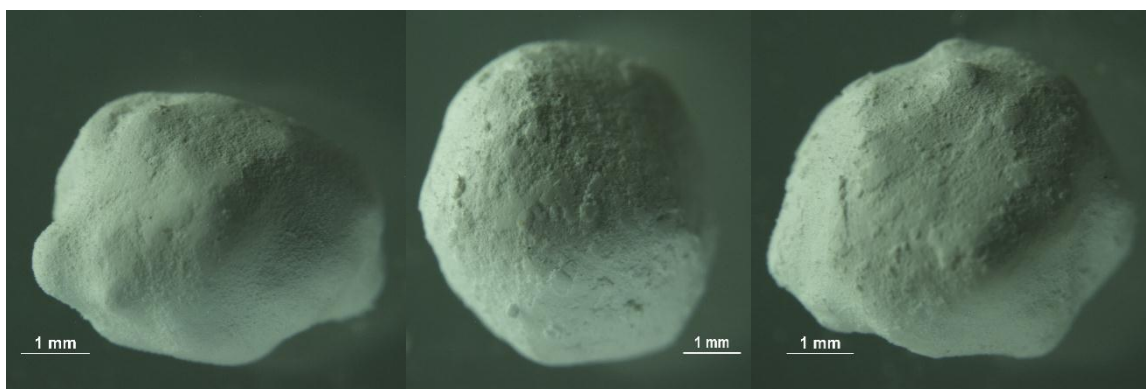


A

B

C

Figure 0.B.6: Optical Microscopy Powder A 10 minutes (A) 5.5 cP (B) 35 cP (C) 70 cP

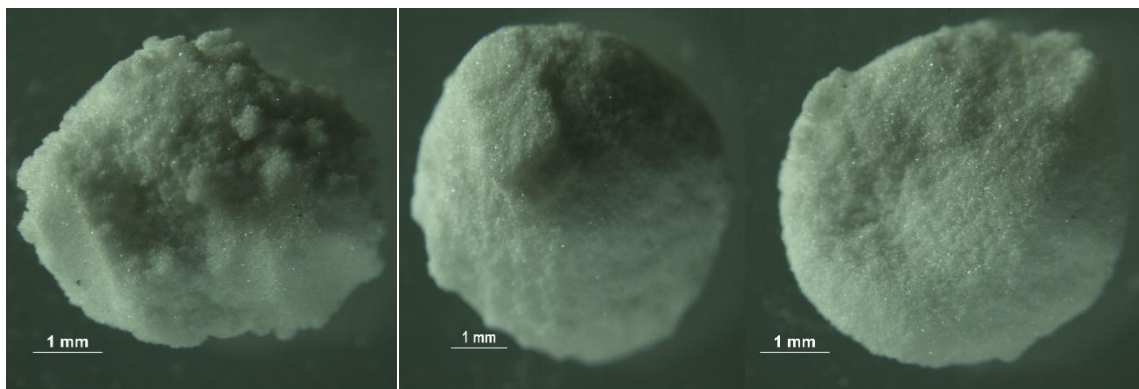


A

B

C

Figure 0.B.7: Optical Microscopy Powder A 15 minutes (A) 5.5 cP (B) 35 cP (C) 70 cP



A

B

C

Figure 0.B.8: Optical Microscopy Powder C 3 seconds (A) 5.5 cP (B) 35 cP (C) 70 cP



A

B

C

Figure 0.B.9: Optical Microscopy Powder C 10 seconds (A) 5.5 cP (B) 35 cP (C) 70 cP



A

B

C

Figure 0.B.10: Optical Microscopy Powder C 1 minute (A) 5.5 cP (B) 35 cP (C) 70 cP



A

B

C

Figure 0.B.11: Optical Microscopy Powder C 5 minute (A) 5.5 cP (B) 35 cP (C) 70 cP

## APPENDIX C: RESULTS TABLES

Table C.1: Macro-Void Size and Volume Fraction Results, including 95% CI on mean value							
Powder/Liquid Reference	Time (minutes)	Vol Fraction Largest Void	95% CI	Maximum Diameter( $\mu\text{m}$ )	95% CI	$\epsilon_{\text{void}}$	95% CI
10PK32, Submicron	0	0.266	0.011	2397.6	81.7	0.2676	0.010836
10PK32, Submicron	5	0.148	0.0164	2187.7	64.4	0.164052	0.010009
10PK32, Submicron	10	0.1234	0.0091	1944.4	58.27	0.133044	0.008406
10PK32, Submicron	15	0.131	0.016	2037.2	87.43	0.137224	0.010901
5PK90, Submicron	0	0.31408	0.014459	2716.8	49.4	0.314563	0.014401
5PK90, Submicron	5	0.179	0.015	2396	54.9	0.194	0.0124
5PK90, Submicron	10	0.157487	0.007624	2425.197	53.1	0.170426	0.008977
5PK90, Submicron	15	0.163	0.012	2360	55.81	0.17	0.010904
7PK90, Submicron	0	0.274886	0.014402	2687.156	77.9	0.275109	0.014381
7PK90, Submicron	5	0.17761	0.014674	2483.477	64.6	0.19691	0.012475
7PK90, Submicron	10	0.182251	0.008655	2326.597	52.6	0.199529	0.00885
7PK90, Submicron	15	0.186	0.0056	2428	45.9	0.193	0.005
	Time (seconds)						
10PK32, 25 micron	3	0.062311	0.015473	1535.909	150.5723	0.097487	0.022153
10PK32, 25 micron	10	0.093443	0.03132	1845.9	329.8461	0.122786	0.027772
10PK32, 25 micron	60	0.084046	0.02387	1828.704	187.1059	0.114421	0.01998
10PK32, 25 micron	300	0.101182	0.032632	1869.721	221.2334	0.13293	0.03402
5PK90, 25 micron	3	0.10467	0.01758	2002.98	123.4233	0.122573	0.01398
5PK90, 25 micron	10	0.117229	0.020429	2012.989	166.3089	0.133155	0.020291
5PK90, 25 micron	60	0.054368	0.021738	1549.579	267.987	0.085023	0.02031
5PK90, 25 micron	300	0.084152	0.026865	1716	309.1	0.097	0.028
7PK90, 25 micron	3	0.056881	0.029399	1409.03	317.7651	0.089858	0.033835
7PK90, 25 micron	10	0.061296	0.031088	1471.603	381.3435	0.076729	0.026451
7PK90, 25 micron	60	0.082067	0.017882	1740.9	147.7314	0.099377	0.017774
7PK90, 25 micron	300	0.063225	0.022951	1612.041	200.4649	0.075342	0.021347

Table 3.2: Total volume of wrapped granule, expressed as equivalent volume sphere diameter			
Powder/Liquid Reference	Time (minutes)	Granule Equivalent Sphere Volume Diameter ( $\mu\text{m}$ )	95% CI
10PK32, Submicron	0	3728	85.8
10PK32, Submicron	5	4115	131.3
10PK32, Submicron	10	3910	60.8
10PK32, Submicron	15	4020	93.3
5PK90, Submicron	0	3999	30.5
5PK90, Submicron	5	4264	60.16
5PK90, Submicron	10	4492.5	38.1
5PK90, Submicron	15	4328	80.4
7PK90, Submicron	0	4155	72
7PK90, Submicron	5	4426	60.9
7PK90, Submicron	10	4142	89.5
7PK90, Submicron	15	4255	68.7
	Time (seconds)		
10PK32, 25 micron	3	3927	96.1
10PK32, 25 micron	10	4179	98.6
10PK32, 25 micron	60	4256	118
10PK32, 25 micron	300	4167	140
5PK90, 25 micron	3	4283	79
5PK90, 25 micron	10	4149	80
5PK90, 25 micron	60	4216	65.6
5PK90, 25 micron	300	4231	58.5
7PK90, 25 micron	3	3949	105
7PK90, 25 micron	10	4059	124
7PK90, 25 micron	60	4060	62.2
7PK90, 25 micron	300	4152	54.5

#### APPENDIX D: ELECTRONIC FILES

The remaining data and outputs are contained in an electronic data repository. This includes XRCT Images, MATLAB outputs, and other raw data. Data is stored in appropriately labeled folders with .txt files providing further navigation within the individual folders as necessary.

Folder Titles are as follows:

XRCT Images

MATLAB Outputs

Axial and Radial Results

Surface Area Results

Material Properties

Droplet Size Measurement

VITA

## VITA

Nathan Bellemore Davis was born in Houston, Texas in June 1989. He graduated high school from Cypress Falls High School in Cypress, Texas in June 2007. He went on to attend Syracuse University where he received his B.S. degree in Chemical Engineering in May 2011. He joined Purdue University in August 2011 to work towards his Ph.D. degree in Chemical Engineering. His research interests lie in the broad area of powder processing and he plans to pursue an industrial career in the near future.