GMA
Granular Microstructure Analyzer

Version 1.0

Nathaniel Ching

Granular Science Laboratory
New Jersey Institute of Technology

August 31, 2011

Abstract

Granular Microstructure Analyzer is a MATLAB program with a Graphical User Interface (GUI) for analyzing the microstructure of arrangements of particles. GMA can find the Voronoi volumes, solids fraction, free volume, voids fraction, and coordination number of each particle. GMA can only do calculations on systems of spheres with a uniform radius. Periodic and non-periodic boundaries are supported.

Table of Contents

Overview

GMA (Granular Microstructure Analyzer) is a MATLAB Graphical User Interface (GUI) for analyzing the microstructure of arrangements of particles. The program eases calculations of several particle-by-particle statistics. When given several groups of particles, GMA will do an average over all of the groups. The program can process both 3D and 2D arrangements of uniform spheres.

Some simulations of particles use periodic boundaries. Figure 1 illustrates a periodic boundary. The particle passing through the boundary on the left side "wraps around" and arrives on the right side. GMA will simulate the periodic boundaries when doing calculations, as shown in the Algorithms section of this report.

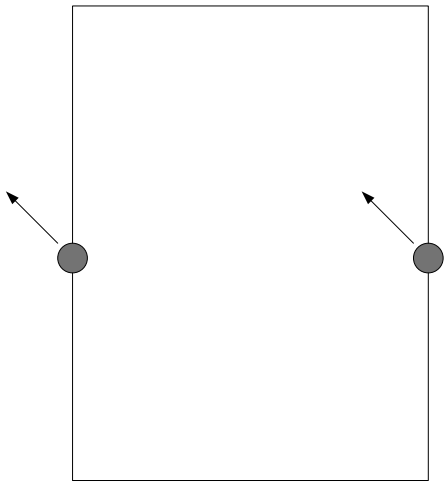

Figure 1 – Illustration of Periodic Boundaries in the Horizontal Direction

In order to avoid unnecessary wordiness, the references to shape and geometry in this report will be in 3D. Thus, the space enclosed by the geometry will be called the "volume" and the particles will be "spheres." However, the mathematics for 2D problems is done with circles and areas, so the results are valid for both 2D and 3D.

The "Theoretical Background" section introduces the key theories behind GMA. The "User Interface" section gives an overview of the user interface. The "Algorithm" section details the algorithm for finding the Voronoi cell volumes and the coordination number. The "Features and Specifications" section deals with system requirements, special features, and the file formats. "Future Work" covers features that could be added to GMA. The appendix contains a short history of the code and a copy of the license.

Theoretical Background

Voronoi Cells

The Voronoi cell of a point P in a group of points S is the space closer to point P than to any other point in S. The boundary of the Voronoi cell of P is equidistant to at least two points in S. The following picture shows the Voronoi cells for a 2D disk arrangement. The top is open, so the edges of the Voronoi cells extend to infinity.

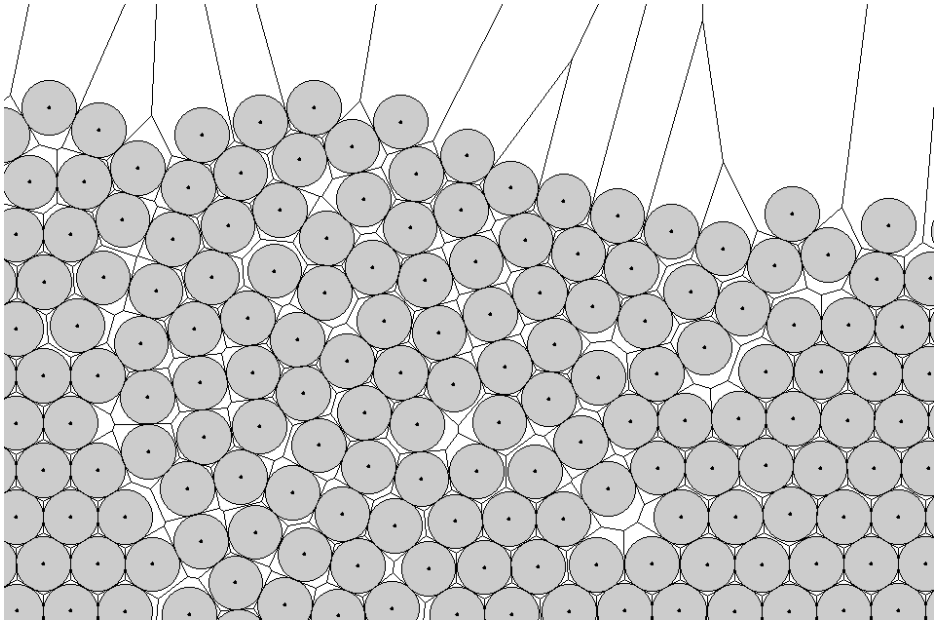

Figure 2 – Sample of 2D Arrangement with Voronoi Cells Drawn

The Voronoi cell is useful in the study of granular materials because it gives a precise definition of how much space a single particle takes. This allows a local definition of volume and density in an arrangement, down to the single-particle level. Also, particles that share a common Voronoi surface can be thought of as neighbors. This allows an exact definition of particle neighbors without requiring the particles to touch.

Kepler Conjecture

The Kepler conjecture states that the densest possible arrangement of 3D spheres has a density of

$$\frac{Sphere\,Volume}{Dense\,Cell\,Volume} = \frac{\pi}{\sqrt{18}} \approx 0.74048 \tag{1}$$

The proof of the Kepler conjecture has not been formally accepted nor disproven [1]. The two known crystals that give this density are Hexagonal Close Pack (HCP) and Face Centered Cubic (FCC).

Voronoi Volume at Dense Crystal Packing

Using the Kepler conjecture (1), the Voronoi cell volume of the densest possible 3D arrangement is

$$\text{3D Crystal Voronoi Cell Volume} = \frac{\sqrt{18}}{\pi} V_{sphere} = \frac{\sqrt{18}}{\pi} \cdot \frac{4}{3} \pi r^3 = 4r^3 \sqrt{2}$$

In a 2D arrangement, the densest possible crystal is a hexagonal crystal with hexagonal Voronoi cells. The minimum diameter of the hexagon is the diameter of the particle. The hexagon is made of 6 equilateral triangles with a height of r. Therefore, the hexagon has an area of

$$\text{2D Crystal Voronoi Cell Area} = 6\left[r\left(\frac{2r}{\sqrt{3}} \right) \right] = 2r^2 \sqrt{3}$$

Solids Fraction

The solids fraction is simply the ratio of the solids volume to total volume. The solids fraction is synonymous to the density of the packing.

$$\text{Solids Fraction} = \frac{\text{Sphere Volume}}{\text{Voronoi Cell Volume}}$$

Free Volume

The free volume is the amount of volume that could be freed if the arrangement of particles formed the densest possible crystal. This is the definition used by Kumar [2].

$$\text{Free Volume} = \text{Voronoi Cell Volume} - \text{Crystal Voronoi Cell Volume}$$

Voids Fraction

The voids fraction is the ratio of the free space, or voids, to the total space. The formula for the voids fraction is

$$\text{Voids Fraction} = \frac{\text{Voronoi Cell Volume} - \text{Sphere Volume}}{\text{Voronoi Cell Volume}}$$

Coordination Number

The coordination number is the number of particles touching the current particle. For a 2D system, the maximum coordination number is 6, which happens in a hexagonal crystal. For a 3D system, the maximum coordination number is conjectured to be 12 [1], which happens in both a FCC and a HCP crystal.

Gamma Distribution

The gamma distribution is an exponential distribution for continuous, non-negative variables. Before the gamma distribution can be defined, the gamma function must be defined. The gamma function is a generalized factorial.

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt$$

The gamma Probability Density Function (PDF) has two parameters in addition to the variable. The parameter k is often called the shape and θ is often called the scale.

$$f(x|k,\theta) = x^{k-1} \frac{e^{-x/\theta}}{\theta^k \Gamma(k)} \qquad ,x \in [0,\infty)$$

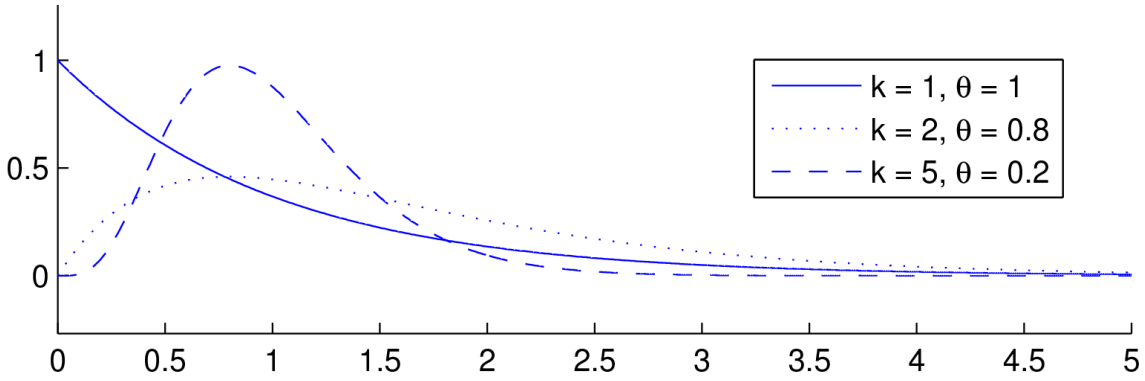

Figure 3 – Gamma PDF with various parameters

User Inteface

The user interface is designed to make using GMA simple. GMA is a single-window application, and multiple instances of GMA can be run at the same time. However, multiple instances of GMA may not calculate in parallel due to limitations imposed by MATLAB Figure 4 is a screenshot of GMA at startup.

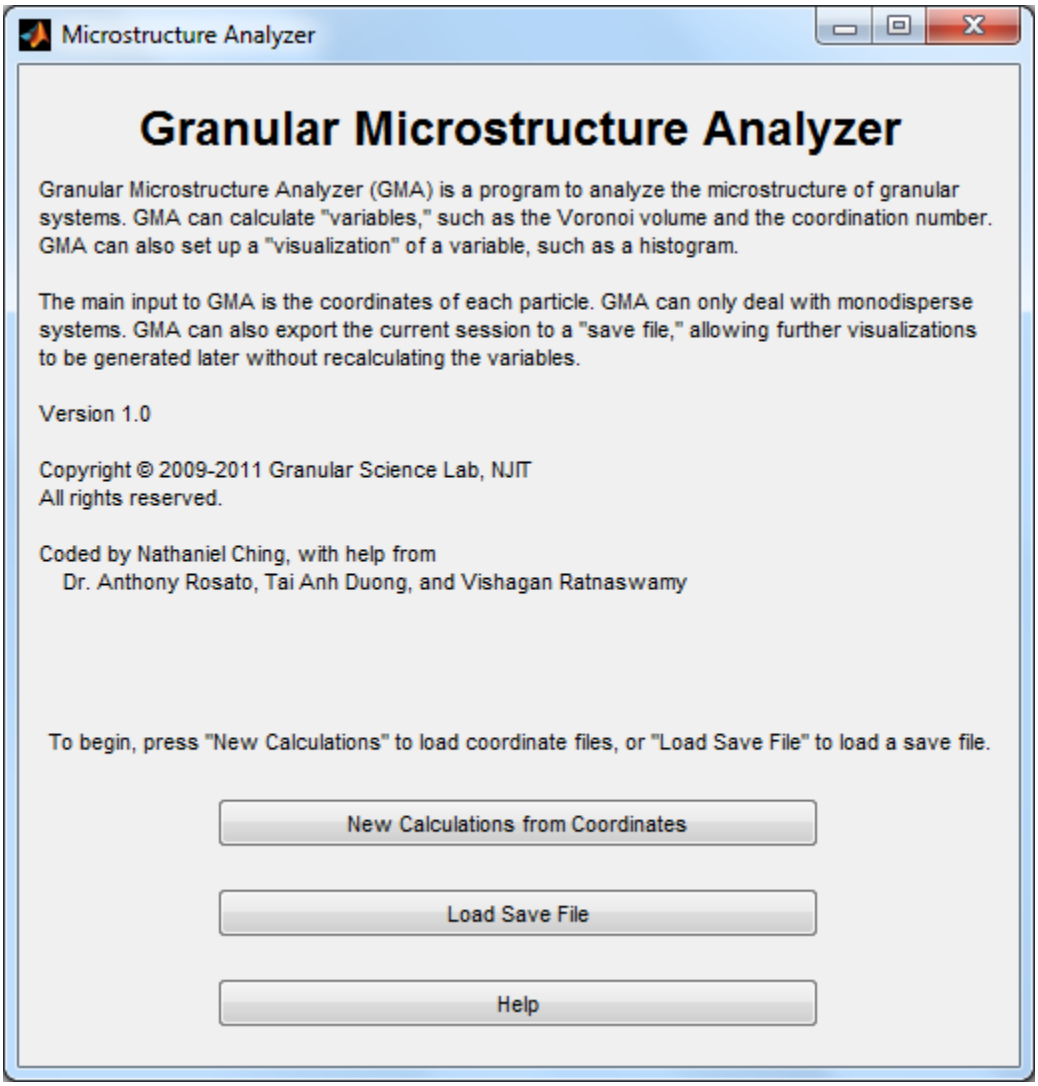

Figure 4 – Startup Screen for GMA

GMA first asks the user what is wanted. Then, GMA finds a way to display what the user wants, asking for additional information if necessary. Figure 5 shows a flowchart of the user interface for GMA. “Necessary parameters” includes the particle radius and the location of the boundaries.

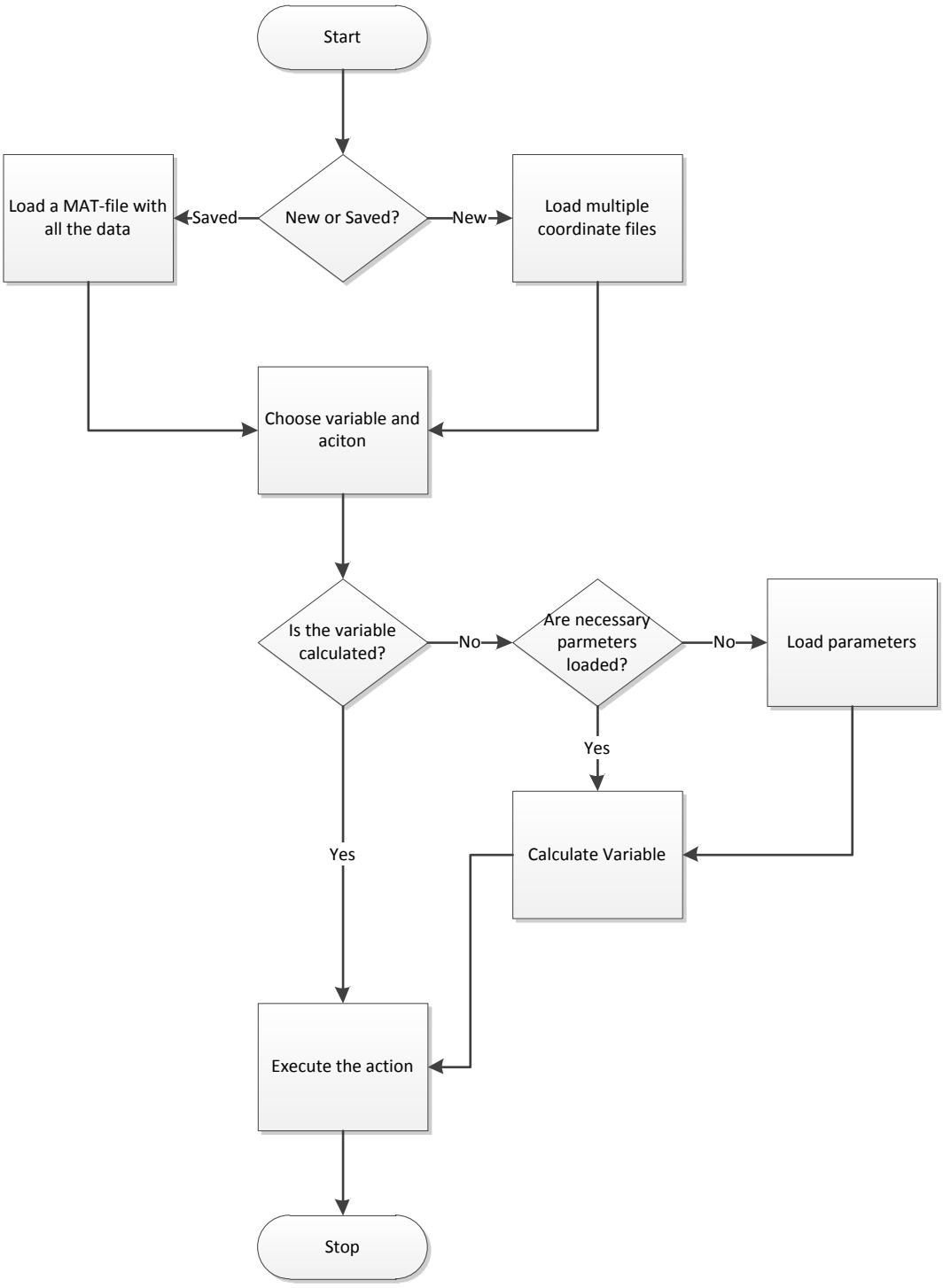

Figure 5 – Flowchart for the User Interface of GMA

Algorithm

Periodic Boundaries

To find the Voronoi cell volume and the coordination number of particles near

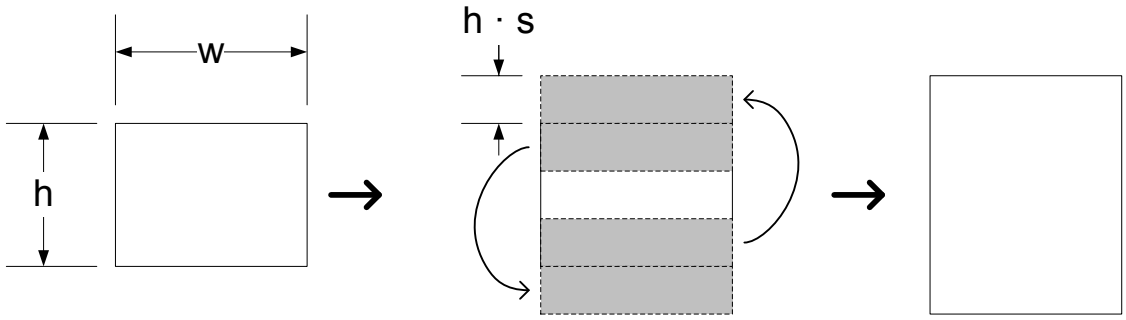

Figure 6 – Illustration of Copying Particles Around Periodic Boundaries

When finding the Voronoi volumes, the ratio added is 1,so a whole copy of the original particles to be copied above and below the boundaries. The result is that the number of particles is multiplied by 3 for each direction that the arrangement is periodic.

$$s = 1$$

When calculating the coordination number, particles within one particle diameter of the boundary are copied.

$$h \cdot s = \text{particle diameter}$$

Voronoi Cell Volume

The Delaunay triangulation is found for the entire arrangement of particle centers, including the duplicated particles. The Delaunay triangulation is found using the Computational Geometry Algorithms Library (CGAL) [3] included in MATLAB. The vertices of the Voronoi cells are found from the Delaunay triangulation using another CGAL function. Each of the Voronoi cell vertices is checked to see if it is inside the convex hull of particles or outside. The results of the search are stored in an array.

Next, the algorithm loops through each particle. If there are periodic boundaries, the loop only goes through the original particles. Using the search results array, the program can quickly determine if any of the Voronoi vertices for this particle is outside the convex hull. If none of the Voronoi cell vertices are outside the convex hull, then the volume of the Voronoi cell is determined and added to the output.

The calculations are done in double precision with the results being recorded to 10 significant figures. The operation of finding the Voronoi vertices is the operation that takes the most time, but it cannot be optimized further in the m-files.

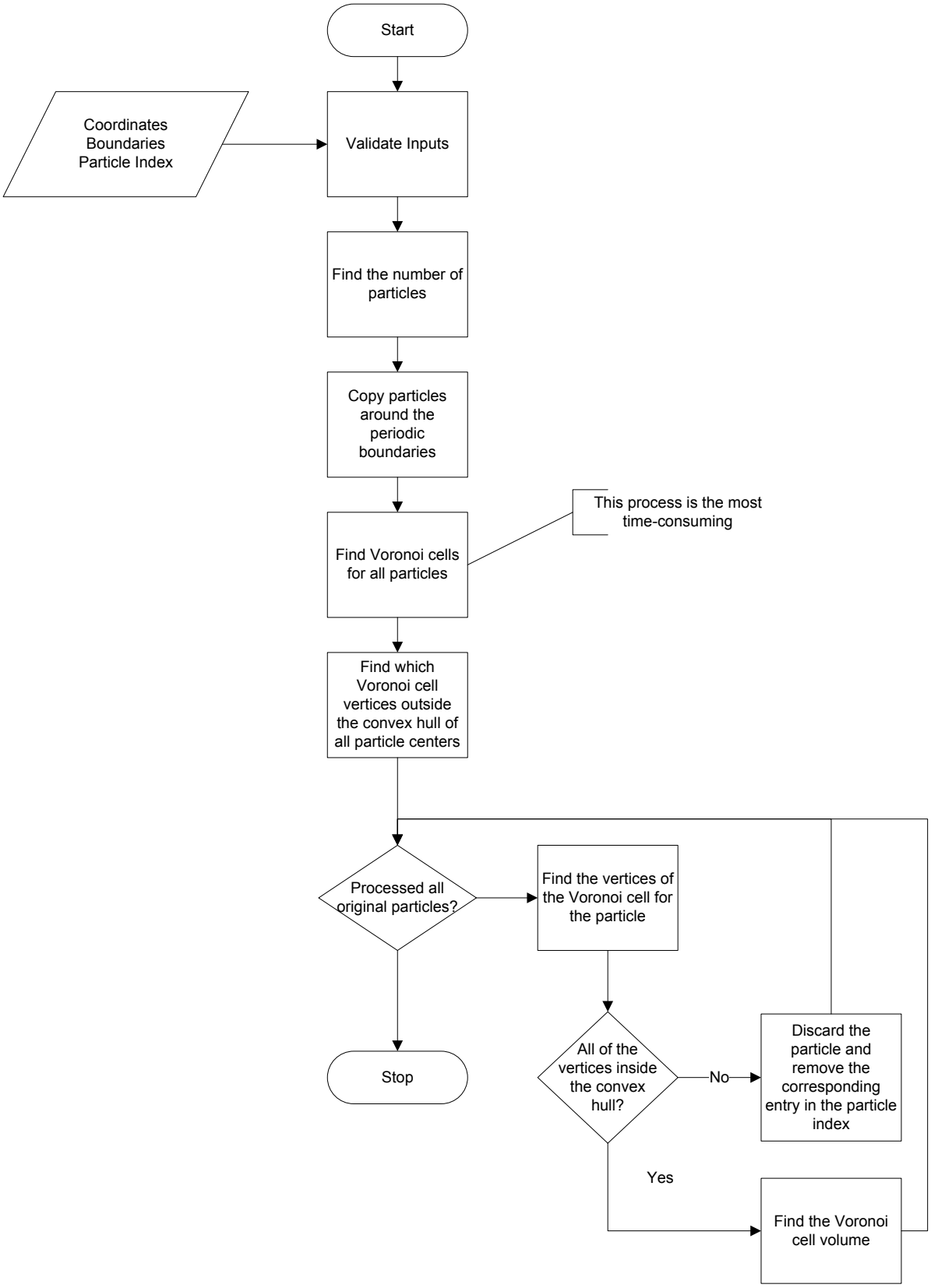

Figure 7 – Flowchart for Voronoi Volume Algorithm

Coordination Number

Before copying the particles around the periodic boundaries, the algorithm adds a particle index to the coordinates.

The algorithm loops through all particles, including duplicates, and finds which are touching. One loop goes through all of the particles. In the loop for each particle, another loop goes through all particles higher than the current particle. Each particle is checked with every other particle, until every particle has been checked against every other particle once. For each pair of touching particles, the particle index of each particle is added to the other particle's neighbor list. The sequential index for the duplicated particles will not match with the array index used by MATLAB.

To illustrate, particle 1 is checked against particles 2, 3, 4, …,n, where n is the number of all particles, including duplicates. Particle 2 is checked against particles 3, 4, 5, …, n. Particle 3 is checked against particles 4, 5, 6, …, n.

After finding the touching particles, the algorithm loops through the list of neighbor lists. The list of neighbor lists has as many entries as there were particles originally. For each entry, the algorithm first makes the neighbor list unique. Then, the number of particles in the neighbor list is counted and saved as the coordination number.

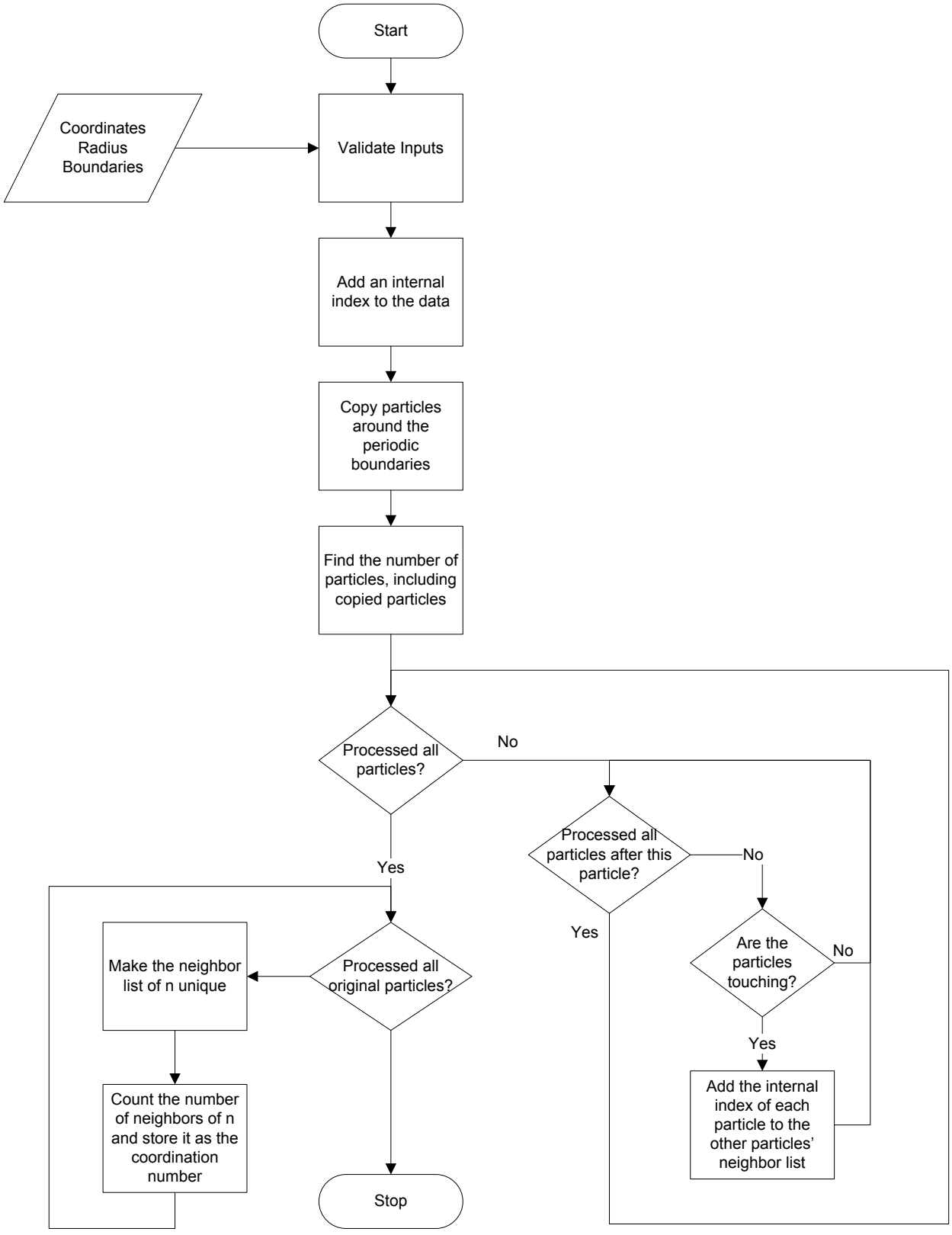

Figure 8 – Flowchart for the Coordination Number Algorithm

Histogram

The calculations for the histogram variables will preserve particle indices if found. This allows a user to input a Voronoi volume file with indices and get the selected histogram variable back with the indices.

The histogram of the coordination number uses a bin width of 1, because the coordination number can only be non-negative integers. The number of bins in the histogram for the other statistics is determined using

$$\text{Number of Bins} = \sqrt{\text{Number of Particles}}$$

The 2-parameter gamma fit is done using the `gamfit` function in MATLAB. The area underneath a PDF is always 1, so the total area of the histogram bars is normalized to match the height of the gamma PDF when the gamma fit is done.

When no gamma fit is done, the height of the histogram bars is normalized by the number of particles and multiplied by 100. Therefore, the resulting histogram shows the percentage of particles in a certain range. Figure 9 and Figure 10 show the histograms produced by GMA. The histogram in Figure 9 is from one arrangement of particles, so it is not smooth as would be expected from an average of many arrangements.

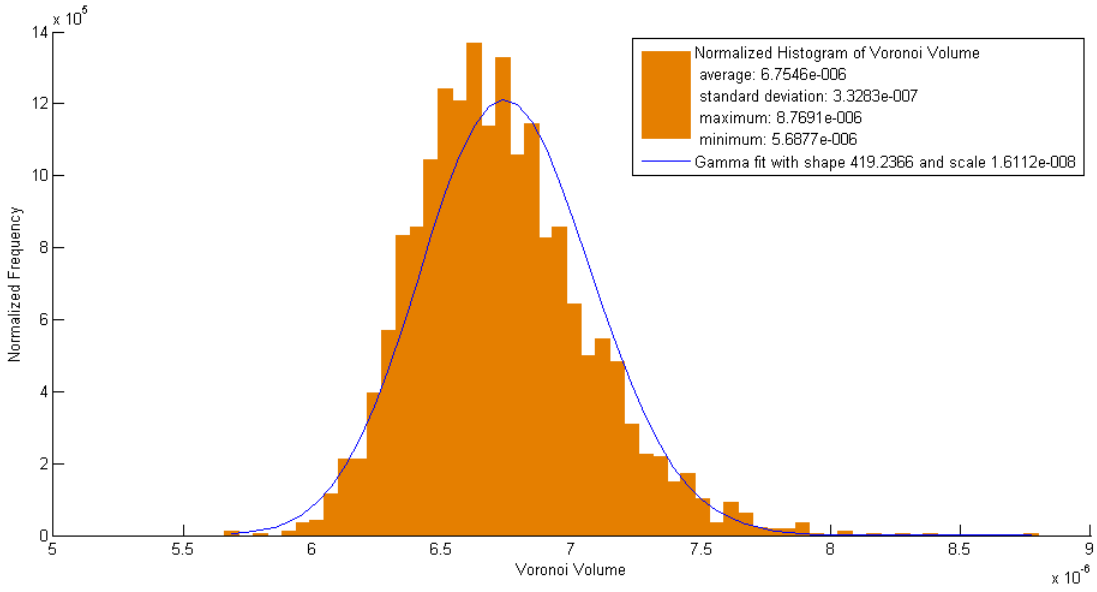

Figure 9 – Histogram of the Voronoi Volume with a Gamma Fit

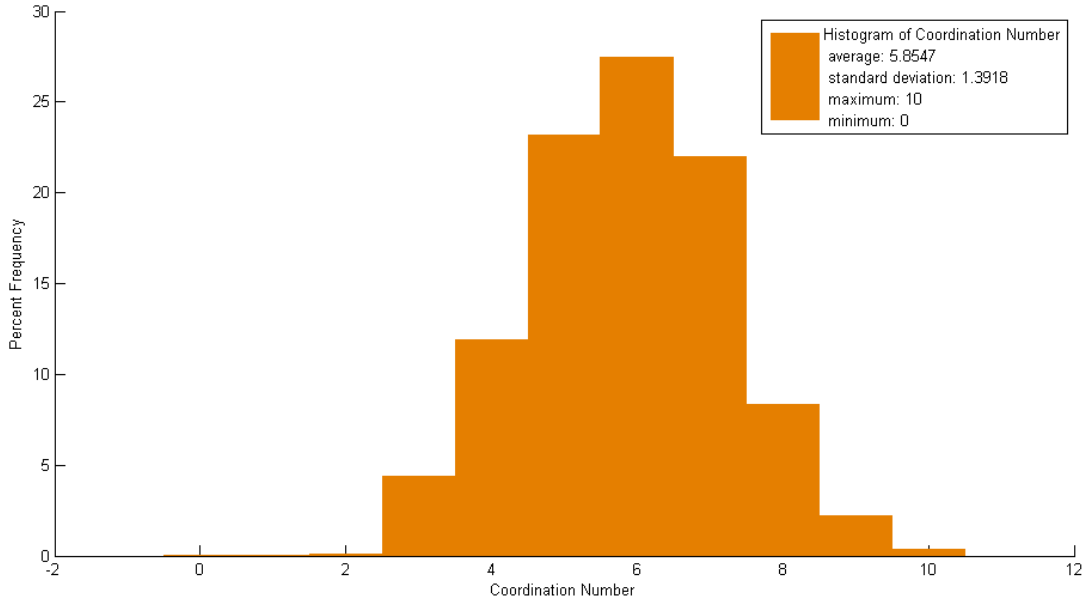

Figure 10 – Histogram of the Coordination Number

Features and Specifications

System Requirements

- MATLAB 7.8 (R2009a) and above.
  GMA uses the Computational Geometry Algorithms Library [3] that was added to MATLAB 7.8 (R2009a). Since GMA only uses m-files, it is both 32-bit and 64-bit compatible for any platform.
- 1 GB of physical memory (RAM)

Input File Format

The input files are plain text files that can be tab, space, or comma delimited. Each row of the file must have the x, y, and z coordinates of a single point, in that order. The file also can have a particle index in the first column. If only one input file is processed, then the particle index will be copied onto the result. Input files cannot have a header. In addition, input files with duplicate data points will produce errors.

Example input file:

```
1    0.9157    0.9340    0.1712
2    0.7922    0.6787    0.7060
3    0.9595    0.7577    0.0318
4    0.6557    0.7431    0.2769
5    0.0357    0.3922    0.0462
```

Output File Format

The output files are plain text, with 10 significant digits of precision for non-integer data. If only one input file was processed and that input file had an index, then that index will be put into the output file. The output files have Microsoft Windows compatible newlines and use the tab character as a delimiter.

Help

GMA comes with a help file. The help file gives the user a quick reference to the program. The button to view the help file is on the bottom of the main screen.

M-Lint

M-Lint is MATLAB's code checking tool, which checks style as well as syntax. The code contains M-Lint error suppression comments, which always start with "%#ok". All of the warnings that can be ignored are suppressed, so M-Lint shows no warnings for all of the source code. This makes new and valid warnings easier to find, because they are the only warnings in the code.

Future Work

1. GMA can currently do a 2-parameter gamma fit of the data. An additional feature would be a post processing option to find the best statistical distribution for the histogram.
2. The radial distribution can be added as a variable.
3. Multiple timesteps is partially implemented. The data structure can handle multiple timesteps, but there is no way to load multiple timesteps. With multiple timesteps, a time animation or chart can be made, showing how a variable changes over time.
4. A new visualization can be created which views the spheres using OpenGL. The spheres would be colored to represent the value of a variable on the sphere. The sphere size could also be dependent on the variable representing the sphere.
5. An installer can be created which will automatically unzip the code and make a shortcut to run the program.

References

[1] Hales, T. C., 2000, "Cannonballs and Honeycombs," Notices of the American Mathematical Society, pp. 440-449.
[2] Kumar, V. S., and Kumaran, V., 2005, "Voronoi cell volume distribution and configurational entropy of hard-spheres," The Journal of Chemical Physics, 123(11), p. 114501.
[3] "Computational Geometry Algorithms Library," http://www.cgal.org.

Appendix

Code History

GMA 1.0

- Finished August 2011
- Renamed GMA (Granular Microstructure Analyzer)
- Complete rewrite, using packages
- Uses extensions for the variables and visualizations. It is now possible to add and edit the variables without changing the core files.
- Preparation for multiple timesteps
- User can save the entire calculation set as a mat-file

Microstructure Analyzer 1.0

- Finished December 2010
- Renamed from "VoronoiGUI" to "Microstructure Analyzer" to reflect the addition of non-Voronoi based calculations to the capabilities of the program
- Complete rewrite, using a subdirectory for functions to reduce the size of the main file
- Removes the ability to pause
- Removes the ability to save the entire calculation set. The user can still save the individual results.

VoronoiGUI 1.0

- Finished July 2010
- Number of screens was reduced to 4
- Uses zipped save files that contain the input, the volumes, and the variables that create the histogram
- Does calculations inside a temporary directory so that they don't have to overwrite other calculations
- Allows the user to save
- Number of variables that could be plotted on the histogram was reduced to the four most likely
- Algorithm change for discarding cells, to discard very large but not infinite cells
- Algorithm now uses the DelaunayTri class instead of Qhull, resulting in speed gains because the Delaunay triangulation is only calculated once.

VoronoiGUI 0.5

- Finished February 2010
- Allowed users to pause calculations
- GUI was broken up into 11 screens
- Gave the user an option to save the volumes to another file
- Histogram could be calculated for several new variables
- Program name changed to VoronoiGUI to reflect that the program was intended to do more than just Voronoi volume calculations

VoronoiGUI 0.1

- Finished December 2009
- Code now done by Nathaniel Ching
- Program was named VorVolGUI for “VORonoi VOLume GUI”
- Initial GUI release, copied functionality of code by Tai Anh Duong
- GUI was all in one screen
- Added smart detection of indices, which allows for indices in some files but not in others
- Speed gains of 6 to 10 times due to elimination of useless file writes

Original

- Code done by Tai Duong
- Calculated the Voronoi volume, discarded infinite cells, and showed a histogram of the free volume and fractional volume
- MATLAB command line only

License

The Granular Microstructure Analyzer program is licensed with a BSD license, which allows modification by third parties. This preamble is not part of the license.