Charged Particle Dynamics in the Lunar Environment

Annelisa Esparza[1] and Alexander Hillegass[2]
NASA Kennedy Space Center
Physics[1] and Computer Science[2]
OSTEM Fall 2020 Session
Date: 10 11 2020

Charged Particle Dynamics in the Lunar Environment

Annelisa B. Esparza[1]
Florida Institute of Technology, Melbourne, Florida, 32901
Email: annelisaesparza@gmail.com

Alexander Hillegass[2]
University of Central Florida, Orlando, Florida, 32816
Email: alexhillegass@gmail.com

As the Artemis Program streams forward, organizations of scientists and engineers across the country have been coming together to solve the complex network of problems to once again achieve the milestone of successfully touching down on the moon. This project is no exception, and it has been an honor to work with the Electrostatics and Surface Physics Laboratory (ESPL), a lab within the Exploration Research and Technology Programs' Spaceport Technologies Office (UB-G) located at the National Aeronautics and Space Administration at Kennedy Space Center (NASA KSC). The authors and their mentor James R. Phillips III[3], in tandem with researchers from the Astrodynamics and Space Robotics Laboratory (ASRL) at the University of Central Florida (UCF), have been working on creating a state-of-the-art (SOA) granular gas dynamics model for particulate contamination prevention and mitigation purposes. In essence, the underlying objective for this project is to more accurately model the resulting electrodynamic interactions between lunar regolith grains with applications to dust mitigation and rocket engine plume surface interactions. To achieve this goal, the team has been expanding upon existing open source classical molecular dynamics code developed by Sandia National Laboratories (SNL). The following reports on the details of the problem at hand as well as the contributions that the authors have made towards resolution, including but not limited to the encoding of physical attributes and interactions for non-spherical polydisperse particle distributions within various bed geometries and preemptive data analysis implementations. Significant data analysis processes were utilized, and several original algorithms were created to perform critical evaluations, resulting in only a 0.006% error in discrepancy.

[1] Graduate NIF Intern, Electrostatics and Surface Physics Laboratory, UB-G, NASA Kennedy Space Center
[2] Undergraduate NIF Intern, Electrostatics and Surface Physical Laboratory, UB-G, NASA Kennedy Space Center
[3] Research Physicist, Electrostatics and Surface Physical Laboratory, UB-G, NASA Kennedy Space Center

Table of Contents

I. Introduction

Kennedy Space Center's *Swamp Works* is the agglomeration of several teams whose goals are to provide exploration mission solutions through lean development processes and a hands-on approach.[1] In the laboratory environment, this manifests in rapid iterations of prototyping, testing, and redesigning. The Electrostatics and Surface Physics Laboratory (ESPL) primarily focuses on discovering techniques to protect flight hardware and launch equipment from electrostatic discharges by studying the physics behind the regolith particles' dynamic electrical states.[2] This project aims to offer a predictive model of lunar regolith's charge transfer and behavior when agitated by dynamical systems causing motion via sliding, vibration, or fluidization within the moon's low-gravity vacuum environment. While it is understood that the charged regolith grains experience dispersion, the trajectories of these particulates are not well known for polydisperse non-spherical cases.[3] By working with the knowledge of previously configured models[4-7] and implementations made over the past few months,[8-10] the team has moved closer towards their goal of extending the modeling capabilities of three-dimensional complex polygons with inherent properties to include their charge behaviors with respect to electrostatic environmental interactions. Once the resultant models have been successfully compared to experimental data, the program can be used to accurately project the precautions required to protect equipment and infrastructure required for a sustained human presence in the lunar environment from the charged dust grains.

II. Details

The details of this project have been broken down into three portions. The first regards the underlying pre-existing code that the project utilizes. The second chronicles the revisions and additions to the open source code. Finally, the third illustrates the data analysis techniques employed upon example projects and features the mathematical reasoning behind the particular approaches. The results of their invocation will be presented later on in Section III, under the subsection *Data Analysis and Visualizations*.

A. Source Codes and Packages

The current modeling is rooted in the classical molecular dynamics code developed by Sandia National Laboratories (SNL) known as Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS).[11] An open source extension of this package by the name of LAMMPS Improved for General Granular and Granular Heat Transfer Simulations (LIGGGHTS)[12] was used as the primary starting point for this project. However, the open source version of LIGGGHTS had not been modified to extend past simulations for monodisperse spherical distributions, so the alterations are described in *Coding Renditions* below. For visualization purposes, the team employed the open source packages of ParaView[13] in conjunction with The Visualization Toolkit (VTK).[14] Communication between all of the open source packages was performed with scripts written by the team in Python 3.[15]

B. Coding Renditions

Simulation accuracy has been improved greatly of the course of the project. Improvements to the LIGGGHTS software package have enabled realistic simulations through the calculation of new forces *(Section 1)*. Creation of a LIGGGHTS application programming interface (API) or *wrapper* has simplified the use of these simulations *(Section 2)*. Multi-sphere approximations of non-spherical particles have facilitated simulation of regolith shaped particles *(Section 3)*. Automated creation of mesh objects enables simulations of the lab environment *(Section 4)*. These improvements have happened both inside LIGGGHTS base files and scripts that facilitate with the usage of LIGGGHTS.

1. Updates to LIGGGHTS

The end goal of a simulation that accurately represents particle interaction in lab experiments cannot be achieved using LIGGGHTS as it is. New forces must be added to LIGGGHTS. Changes to the LIGGGHTS software package itself enable calculation of new forces as the project moves towards realistic simulations. Forces such as electrostatic force, particle to particle gravity, tribocharging, and much more have been added to LIGGGHTS over the course of the project. These forces do not describe every interaction in a lab test but should provide a basis from which to build on. Once testing can proceed, the coefficients for particle properties can be measured and guide the team towards realistic simulations.

2. Improve LIGGGHTS User Experience

The LIGGGHTS particle simulator is a complex software. There are new features found every day that may help the process or make it more complicated. One aim of this project was to create a simplified user experience. The

wrapper created accomplishes this by abstracting away many of the details involved in a simulation and providing one-line access to common features. The API simplifies use of the program, saves time, and reduces the amount of knowledge necessary to set up a simulation. In addition to simplification, it enables entirely new features to supplement the capabilities of LIGGGHTS. These features include ability to create custom mesh objects for each simulation; easy alteration of location, velocity, charge, and size of specific particles; automatic enclosure of simulation area in rigid box; saving and loading of data for multi-stage simulations; easier editability of characteristics of particles; implementation of action queue for movement of rigid mesh objects. Many of these features are accessible through one-line python commands and significantly reduces the amount of writing required to run a simulation. This abstraction also removes the strict rules around ordering of commands. The API removes the required knowledge specific to LIGGGHTS and allows users to focus on design of the simulation environment and physics involved.

3. Non-Spherical Particles

Non-spherical particles are necessary for the project to progress from lab experiments to a fully realized model for particle interactions. The wrapper gives users the ability to import mesh objects in common 3D object formats inside the simulation as particles. These particles can closely resemble regolith samples taken from the moon or other particle shapes used in testing. The limitation of mesh particles is the loss of some forces added to LIGGGHTS for our simulations. To remedy this, a particle type called a *multisphere* can be used. This particle is made up of many spheres that are used to approximate the shape of a mesh. The multisphere particles can keep the shape of mesh particles used but maintain the ability to interact through forces added to LIGGGHTS. Multisphere particles can be created from the same 3D object files used as a mesh body and maintain many of the physical properties of the object modeled.[16]

4. Rigid Mesh Objects

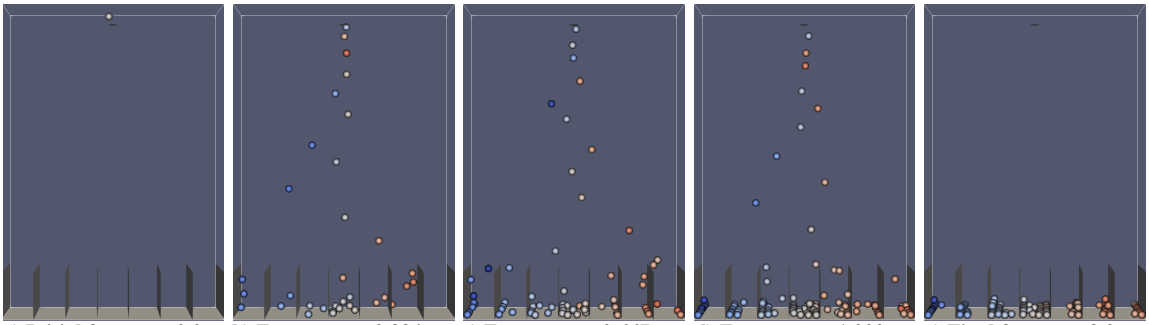
Accurate simulation of a lab test requires all variables to be as consistent as possible. Significant work has been done to ensure the physics behind the simulation are correct, but the environments must also closely match each other. Verification of the model will take place using data collected in 2007 by Hogue[4] at *Swamp Works*. This simulation involved tipping a tray of particles over onto an inclined plane where they then rolled into another tray for collection. Each of these objects must be created and brought to the LIGGGHTS simulation because there is no native method for creation of 3D objects for use in its simulations. To create rigid 3D mesh objects ParaView[13] can be used for custom sizing of each object. Through ParaView, the API is able to programmatically create mesh objects that fit the LIGGGHTS simulation. LIGGGHTS can then import meshes and use the API action queue to manipulate them as needed.

C. Analysis Applications

In addition to these integral applications, the extra time allotted for significant gains on data analysis and visualization techniques to verify accurate simulations prior to their required execution. For the analysis aspect we were able to implement group and regional integration for masses, charges, and count over time *(Section 5)*; enhance the general understanding of the position, movement, forces, and charges of particles through the use of histogram and scatter plots *(Section 6)*; determine particle location based on size dependence over time *(Section 7)*; calculate and visualize conduction rates between particles as a function of conductivity values, charges, particle radii, and separation *(Section 8)*; and develop algorithms to resolve angles of surface indentation and compare these values to the imposing object's structure for accuracy *(Section 9)*.

5. Mass, Charge, and Count Integration

This feature allows the developer to declare a set of particles initialized either within a *region* of defined three-dimensional space or a *group* of particles that have a defined property or range of properties in common to be tracked over time. The term *integrate* means to take the sum of a particular quantity, e.g. *integrate* to solve for the instantaneous total mass of a *group* of particles that initially have negative charge values or *integrate* to find the instantaneous total charge of a collection of particles that were initially in the first-quadrant *region* but perhaps moved out of that space over time. To further the complexity, this *integration* can be applied only upon particles that are within a particular *region* of space at that timestep or upon a *group* of particles that match a particular description. To expand upon the first example, one could *integrate* the *group* of particles that initially have negative charge values, but only include the masses of those that are currently in the first-quadrant *region* for the summation. The ability to perform these *integrations* is useful to see the evolution of charges and masses over time, while knowing the count allows for statistical calculations like standard deviations, averages, and per-particle attributes.



a) Initial frame $t = 0.0$ s. b) Frame at $t = 0.334$ s. c) Frame at $t = 0.667$ s. d) Frame at $t = 1.000$ s. e) Final frame $t = 2.0$ s.

Figure 1. Separation by charge. *Frames demonstrating the charges that are attracted to regions at the bottom based on their initial charge.*[†]

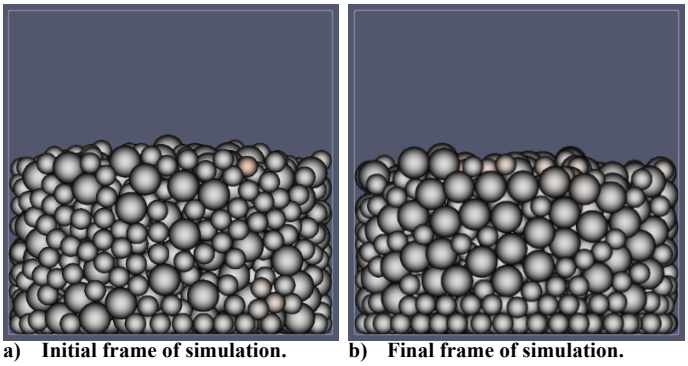
To build off of this application, notice how in the figure above the particles *separate themselves*[*] based upon their charge values[†] as time *t* increases. This simulation was set with the conditions of Gaussian charge distribution, so in theory, the mass within each region should match a Gaussian distribution, wherein the central containment would have the largest mass. In Section 13, we will see two examples of the integration function being utilized: mass per region[‡] in Figure 7b and charge per region[‡] in Figure 7c.

6. Histograms & Scatter Plots

While the groundwork for a histogram script was already in existence by the time we joined the project, the additions and adjustments made since have been considerable. Initially, the script would only display histograms for the particles' positions, i.e. *x-position, y-position,* and *z-position*, and requesting an individual coordinate or alternate characteristic would result in a run-time error. Those run-time errors have been resolved to allow for individual position vectors to be plotted as histograms, as well as any combination of *x, y,* and *z velocity* vectors, *x, y,* and *z force* vectors, scalar *charges,* and scalar *radii*. In addition, a *radial* position vector was defined to allow for cylindrical or spherical coordinate analyses instead of being limited to Cartesian. The framework to plot the particles on a scatter plot in their coordinate system of choice was also constructed. All of these are simple flags at the top of the script for the user to alter to their desired preferences. Once the settings are in place, there is also the ability to create a video compilation of the histograms and their evolution over time, or to simply plot the first and final data set overlapped – but nonetheless distinct – as a single image. An example of this "before-and-after" style can be seen in Figure 8 from Section 14. As an added feature, all of the following sections of data analysis methods have been implemented together as part of the histogram script, allowing multiple techniques and visualizations to be performed at once. Stylistically, this is again performed with simple yes-no flags at the top of the script, occasionally followed by a set of defined global variables for any relevant parameters.

7. Brazil Nut Effect

Building upon the implementations described in the sections prior, we were able to explore the phenomena that larger particles move *against* gravity under vibrational circumstances, also known as the *Brazil Nut Effect*.[17] By applying a vibrational scenario to a spherical polydisperse set where a portion of the particles have a radius of 2 mm and the complementary portion have a radius of 3 mm and allowing it to run for 5,000,000 microsecond timesteps, we were able to observe



a) Initial frame of simulation. b) Final frame of simulation.

Figure 2. ParaView results for Brazil Nut Effect. *Frames demonstrating the Brazil Nut Effect for a collection of mixed particles with 2 mm and 3 mm radii.*

[*] The panels dividing the regions are positively and negatively charged, respectively, leading to the particles being attracted to a region of equal and opposite charge.

[†] Red coloring indicates positive charge while blue coloring indicates negative charge. Greater vibrance in color corresponds to a higher magnitude of charge.

[‡] The regions are explicitly defined in Figure 7a later on in Section 13.

the difference between the layering of particle sizes visually from the ParaView display frames (Figure 2b) as well as in the position histograms (Figure 8 in Section 14) and chart that displays the average height of the particles as a function of cycle (Figure 9 in Section 15).

8. Conduction Rates

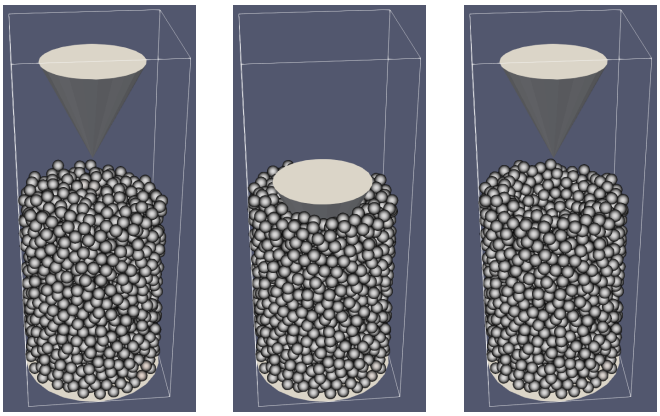
Determining *conduction rates*, or the amount of time it takes for charge to transfer between particles until they reach an equilibrium rate, gives us important insight on the way particles interact as a ratio of mass or surface area to charge. They also provide a convenient way to check if the simulation's interactions are being accurately portrayed. However, the period of conduction happens on the scale of microseconds, even for low conductivity values, which presents a unique problem. To address it, an algorithm to discern equilibrium states for particles post-interaction was created. Since the process is subject to error, three levels of checks were implemented: visualizations, plots, and data structures composed of particle radii, conductivity values, initial and final charge values, timestep values for when conduction begins and ends, and classification of whether or not the conduction has completed during the simulation period are sent to an Excel file. This allows the user to determine whether the results match their expectations over large ranges of initial conditions in one place. The visual and plot aspects are seen in Section 16, Figure 10.

9. Angle of Repose

Determining the angle of repose (AoR) from discrete element method (DEM) simulations is not necessarily a *new* calculation of interest,[18] but the justification behind its determination is nonetheless of importance. On the largest of scales, the ability to determine the slope of an indentation caused by surface interactions from rocket plumes can allow for recalibration of a landing equipment's angle of approach, preventing component degradation from shear forces.

For this obstacle, several functions that call upon each other were added to the histogram script. To verify accuracy, one function reads in the data from the stereolithography (STL) files of the object used to perform the indentation and outputs the dimensions of the object. This allows for the theoretical angle of indentation to be calculated and then compared with the algorithmic results. The critical function that outputs the AoR requests inputs of the dimensions from the STL file, the central coordinate of the object's imposition, the particles' position data, an arbitrary number of bins, and two threshold values that are used when calling an additional function that removes outliers.[§]

To start, the STL read-in function is called to initialize the dimensions of the imposing object and calculates the theoretical AoR. Afterwards, the obvious outliers are removed from the data set, and then the particles located at a height maximum within each bin range are set aside. The outliers from this secondary data set are removed, and then a one-degree polynomial curve fitting is performed. From the slope of this linear result, the angle of the indentation with respect to the ground is calculated.[**]



a) Frame at t = 0.01 s **b) Frame at t = 0.17 s** **c) Frame at t = 0.5 s**

Figure 3. Penetrating cone. *Frames demonstrating the (a) before, (b) during, and (c) after-effects of a cone with a 20 mm radius and height of 40 mm imposing an indentation on a collection of particles.*

The sample calculation for this process will be based on a cone with a radius r of 20 mm and height h of 40 mm penetrating a collection of particles contained in a cylindrical region (Figure 3). Finding the theoretical angle of the indentation θ_{theo} was found using Eq. (1) below:

$$\theta_{theo} = tan^{-1}\left(\frac{h}{r}\right) \quad (1)$$

To establish the preliminary accuracy of the algorithm, the simulation was initialized with a high value of particle *cohesion*, which is a property that causes particles composed of the same substance to stick together. This prevented any particles from slipping due to gravity after the indentation was made, ensuring a clear slope that should match the theoretical angle θ_{theo} from the cone, seen in Section 17, Figure 11.

[§] The outlier removal function performs a combination of logical statements based on relative positions and a statistical comparison based on standard deviation.

[**] Since the slope is equivalent to the angle's tangential component, finding the angle with respect to the "ground" is as simple as taking the inverse tangent of the slope, similar to Eq. (1). See Figure 11 for a clearer depiction.

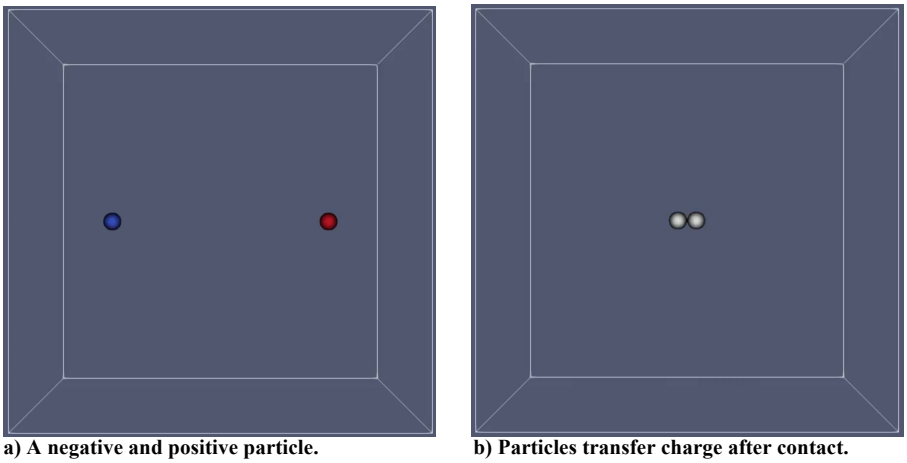
III. Summary of Outcomes

Following a similar format of Section II, the summary of outcomes will be broken down into only two subsections, given that the source code remains relatively unchanged, just extended. The first describes the resulting features from the coding changes mentioned in *Coding Renditions*, while the second includes resulting calculations and data visualizations when the referenced techniques in *Analysis Applications* are applied.

D. Rendition Results

10. Updates to LIGGGHTS

Improvements to the LIGGGHTS simulation package are crucial to the success of the project. Multiple forces and properties have been added to the simulation such as Coulomb force, electrical conductivity, particle-to-particle gravity, and tribocharging. The resulting simulations mechanically resemble what we expect lab results to show but measurements of test particle properties must happen to accurately model the test environment.

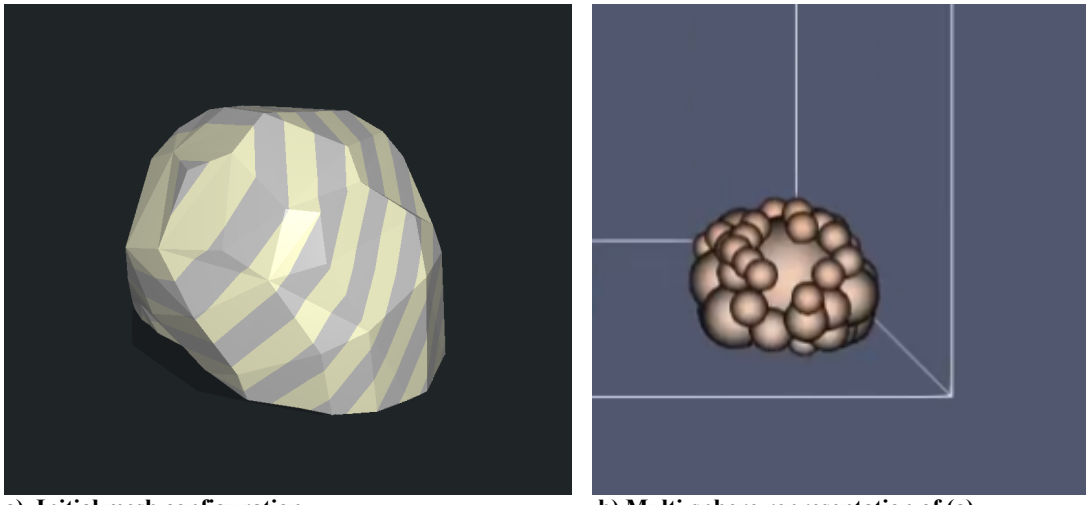


a) A negative and positive particle. **b) Particles transfer charge after contact.**

Figure 4. Example of Coulomb force and conduction. *A negative and a positively charge conductive particle are separated by a distance. Their opposite charge means the particles will attract each other. After the two conductive particles collide the charges cancel each other out.*

11. Non-Spherical Particles

Simulation of lunar regolith particles presents the problem of using non-spherical particle inside LIGGGHTS. There are two types of non-spherical particles supported. Mesh particle shapes can be directly imported to a LIGGGHTS simulation but do not support the custom forces and properties necessary. The other option is a multi-sphere particle that has a similar shape but is composed of spheres. This retains the ability to utilize custom forces because it relies on the LIGGGHTS primitive type of spheres. Multi-sphere particles can be generated from the same files used to import mesh objects but require processing before use; see Figure 5 below as an example.[16]

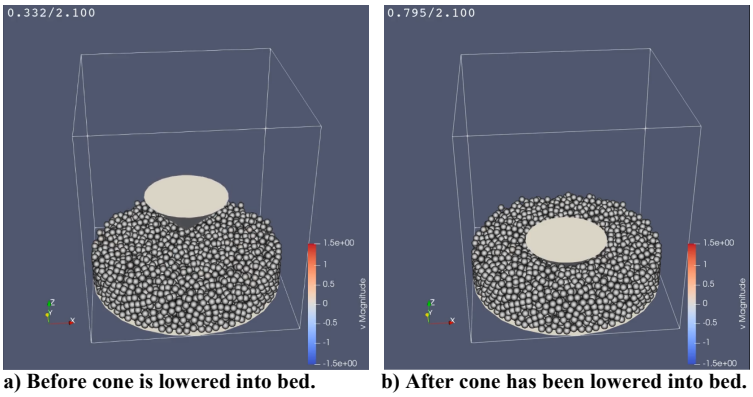


a) Initial mesh configuration. **b) Multi-sphere representation of (a).**

Figure 5. The original mesh and the multi-sphere approximation of that mesh. *The quality of the recreations is determined by the resolution specified in the algorithm and the maximum number of spheres to represent the particle. Increasing either of these values increases the execution time of the algorithm.*

12. Rigid Mesh Objects

Usage of meshes objects is necessary for accurate recreation of lab experiments. The LIGGGHTS API enables the creation and use of meshes in many simulations we run. The current implementation allows for automatic creation of trays, disks, cones, and rectangles which can all be moved around the simulation and interact with each other. Implementation of an action queue allows meshes to move at timed intervals. Figure 6 is a shot of a cratering test involving a cone and a disk. The cone gets lowered into the bed of particles then slowly removed. After removal the disk underneath the particles vibrates to reset particles and get rid of the crater.



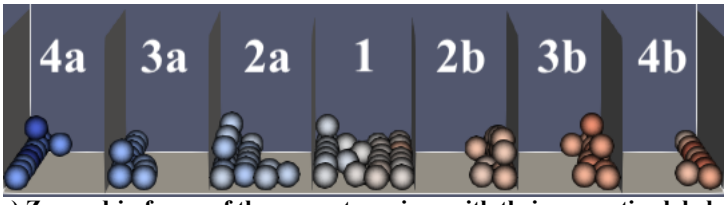
a) Before cone is lowered into bed. **b) After cone has been lowered into bed.**

Figure 6. Action queue being used with mesh. *Before and after images of the cone shaped mesh being lowered into a bed of spherical particles.*

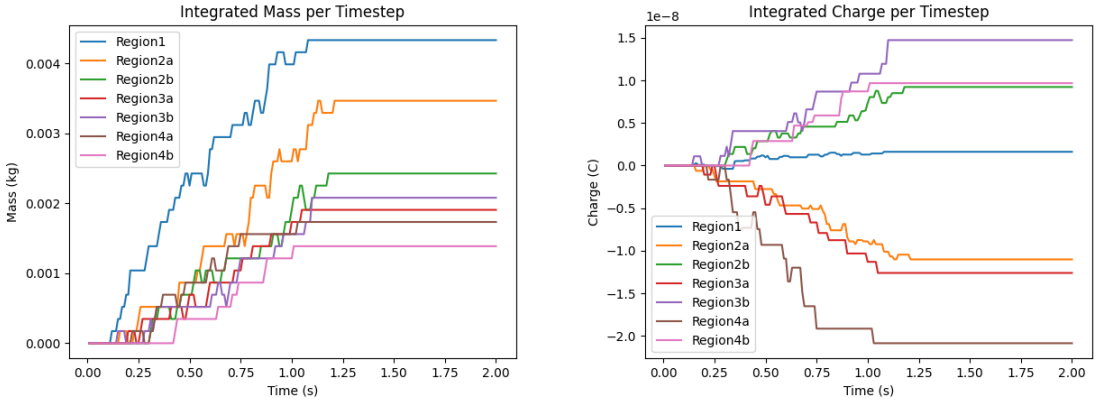
E. Data Analysis and Visualizations

13. Mass, Charge, and Count Integration

While the component to compute regional and group integration regarding particle masses, charges, and overall count is implicit within the LIGGGHTS available commands,[19] the ability to pervasively declare desired regions or groups and properties to analyze was an inclusion of our own design. The wrapper now contains functions that declare the dimensions of regions and characteristics to group particles for initial and instantaneous purposes, flags for which factor(s) of integration to perform, and commands to output and save the results on a per-timestep basis for future analysis. A Python function to read-in data from the JavaScript-based *TypeScript* file was added into the histogram script, as well as several other functions to analyze the resulting timestep integration data.



a) Zoomed-in frame of the separate regions with their respective labels.



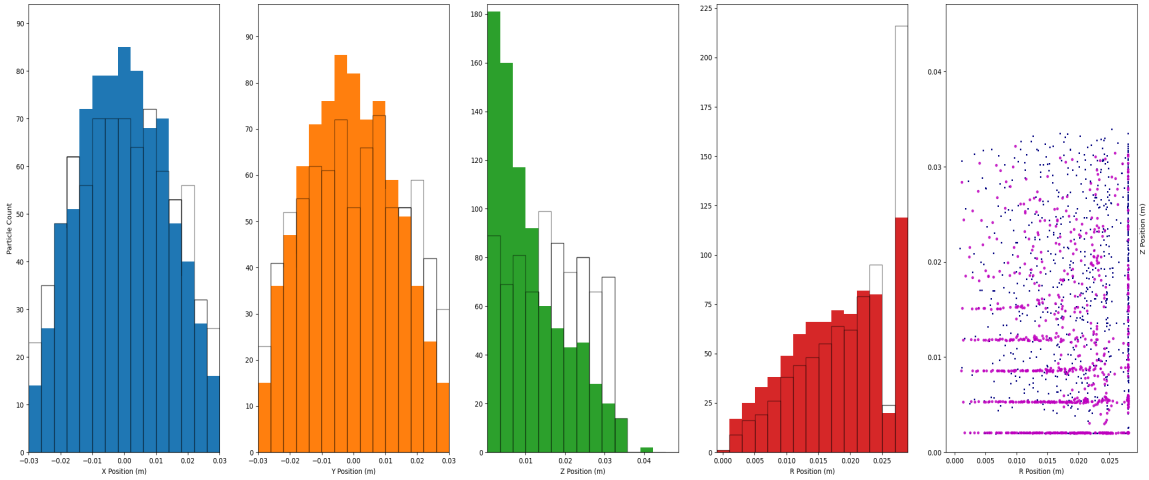
b) Integrated mass as a function of timestep. **c) Integrated charge as a function of timestep.**

Figure 7. Regional integration example. *Depicts the results of integrating with respect to certain regions defined in plot (a) for total mass in plot (b) and total charge in plot (c).*

The results from Figure 1 in Section 5 match expectations, in that with an initially Gaussian charge distribution, there will be relatively Gaussian distribution of mass relative to the central region.

14. Histograms & Scatter Plots

The following histogram and scatter plot example actually utilizes the data created by running the simulation to demonstrate the Brazil Nut Effect (see Figure 2 in Section 7 for initial and final frames).



a) X-position histogram. b) Y-position histogram. c) Z-position histogram. d) R-position histogram. e) Scatter plot.

Figure 8. Histograms for 2 mm radius particles in vibrational scenario. *Histograms and scatter plots for the final frame of particles experiencing the Brazil Nut Effect. Outlines seen in plots (a)-(d) are from the initial frame. The small blue squares in plot (e) indicate the initial positions, while the pink dots indicate the final positions.*

Notice how in Figure 8c there is a distinct shift from a relatively evenly dispersed distribution of particles at all z-positions towards a z-position of zero. This can also be seen in Figure 8e, where it appears that layers of smaller 2 mm particles have settled at the bottom when compared to their initially random positions. This follows the expectations of the phenomena and confirms the accuracy of our simulation pre-requisites.

15. Brazil Nut Effect

The figure to the right takes the data from the scenario previously described and displayed in Figure 2 from Section 7 and transforms it into a clear depiction of larger particles going *against* gravity under vibrational conditions. This was made possible by the creation of a plotting function for average particle height as a function of time evaluated for all particles as well as for particles of each distinct value of radius. For ease of use, the function is triggered simply by a flag in the histogram script. This feature can be implemented in future simulations to ensure proper particle behavior.

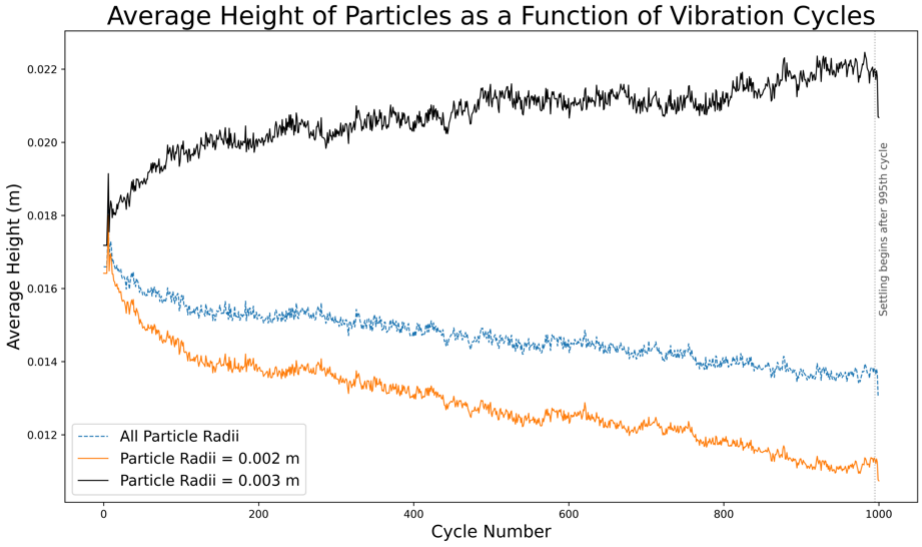
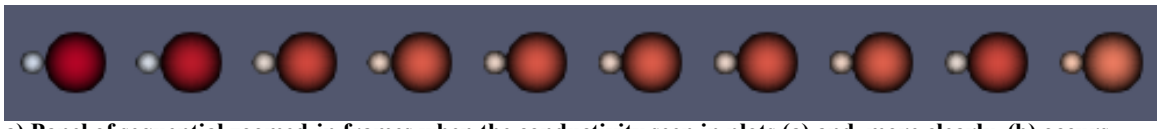


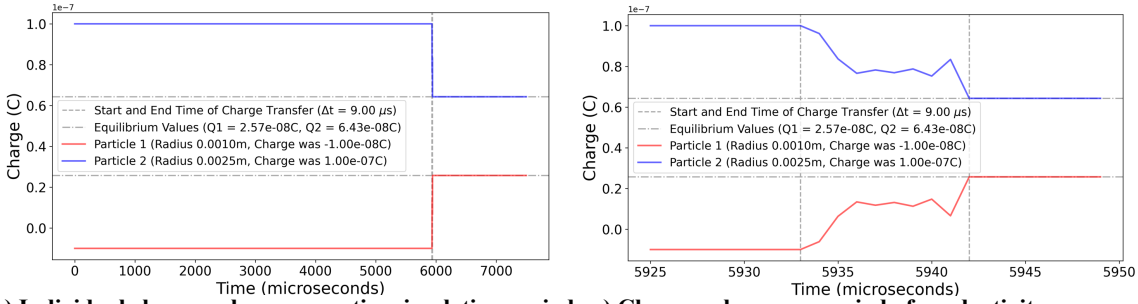
Figure 9. Plotting the Brazil Nut Effect. *Chart demonstrating the Brazil Nut Effect for a collection of mixed particles with 2 mm and 3 mm radii with the mean vertical position as a function of oscillation period with upper (black), lower (orange), and middle (blue) traces corresponding to 3 mm radius, 2 mm radius, and all size particles, respectively.*

16. Conduction Rates

Continuing from Section 8, this particular example is best understood when viewed collectively:



a) Panel of sequential zoomed-in frames when the conductivity seen in plots (a) and, more clearly, (b) occurs.



b) Individual charge values over entire simulation period. c) Charge values over period of conductivity.

Figure 10. Conduction over time. *Frames seen in (a) are from the 5933-5942 μs range and visually demonstrate the charge transfer portrayed in plots (b) and (c).*

For this scenario, two particles with conductivity values of 0.001 S/m are simulated: the smaller particle seen on the left-side of each frame in Figure 10a has a radius of 1.0 mm and initial charge of -10 nC, while the larger particle on the right-side of each frame has a radius of 2.5 mm and an initial charge of +100 nC. According to the plot, the charge conduction occurs over a period of 9 *μs*.

17. Angle of Repose

Starting from Figure 3 in Section 9, the *x-y* coordinates were converted radially, and the data set underwent the process described in Section 9. The result is the following set of images:

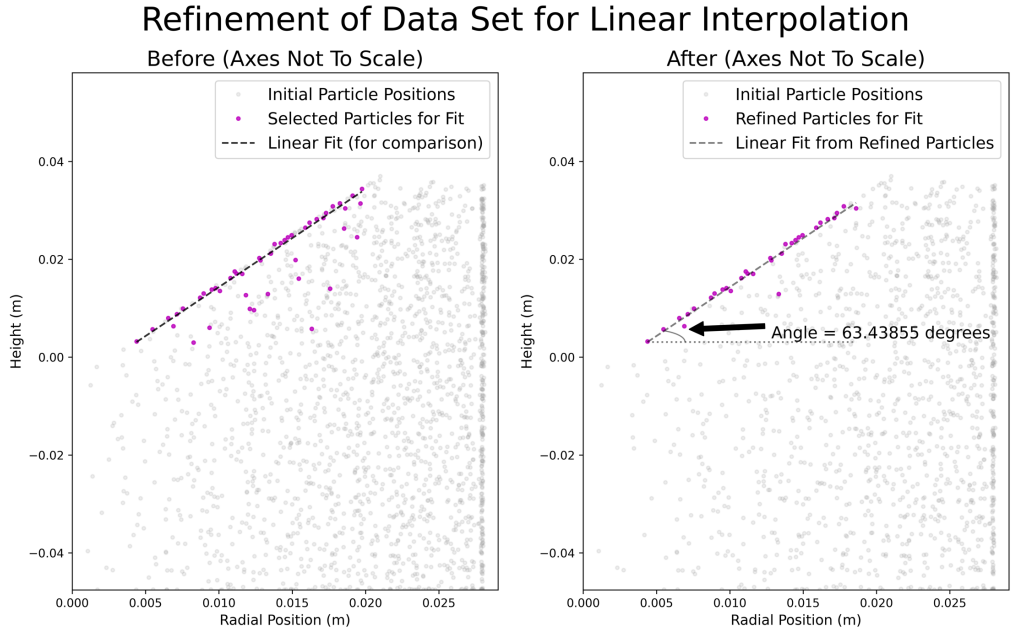


Figure 11. Determining the angle of repose (AoR). *Scatter plots displaying the particle positions where the radial position is with respect to the center of the penetrating cone. The plot on the left is after the initial outlier removal process and maximum height determination, while the plot on the right is after the second iteration of outlier removal.*

The results from Figure 11 are extraordinarily accurate, with a discrepancy of only 0.0036 degrees, which equates to a percent difference of less than 0.006%. The parameters that led to this result utilized the maximum height values for 66 bins of equal width and underwent the outlier removal function[§] twice, with threshold values of three for each round. When applying this algorithm to differently shaped penetrating objects, the parameters usually need to be changed to maintain a high degree of accuracy. However, the aim is to recursively implement chi-squared tests with altered parameter inputs until the linear fit results in a maximum R value in order to expand its applicability to determine unknown angles of repose.

IV. Conclusion

While more work remains, significant progress has been made within the past few months as a result of our efforts and has resulted in a conference publication for the ASCE Earth and Space Conference in 2021.[20] The largest unresolved issue concerns verifying the expanded software's numerical results with experimental data. Although a test chamber for this project exists, COVID-19 has caused a shut-down for non-essential personnel on-center, so we have not been able to perform experiments to compare the model's simulated values. As a result, the project's future success will hinge upon our ability to accurately account for a range of interactions. In the meantime, the simulations have begun being compared to previous results (Ref. 4) for the sloped-ramp Faraday cup experimentations.

Acknowledgments

A. Esparza and A. Hillegass thank their mentor J. R. Phillips III, Research Physicist at NASA Kennedy Space Center's Electrostatics and Surface Physics Laboratory, for the guidance and opportunities he has provided, as well as the rest of the *UCF Plumes* research team from the Astrodynamics and Space Robotics Laboratory at the University of Central Florida. Individually, A. Esparza thanks Dr. C. R. Buhler, Senior Research Scientist at NASA Kennedy Space Center's Electrostatics and Surface Physics Laboratory, for the mentorship and advocation that has helped lead her to where she is today; Dr. J. Riousset, Assistant Professor at Florida Institute of Technology, for the academic and personal counseling on how to successfully complete simultaneous educational commitments; and Dr. S. Koksal, Emeritus Professor of Mathematics at Florida Institute of Technology who has been missed dearly since her passing in March of 2020, for her influential encouragement and support throughout the previous years.

References

[1]"Kennedy Space Center Swamp Works – Developing New Tools for Deep Space Exploration," *Kennedy Space Center Technology Transfer Office.* URL: https://technology-ksc.ndc.nasa.gov/featurestory/swampworks [cited 8 November 2020]

[2]Heiney, A., "Electrostatics and Surface Physics Laboratory," *Kennedy Space Center,* published online 05 Nov. 2018. URL: https://www.nasa.gov/content/electrostatics-and-surface-physics-laboratory [cited 8 November 2020]

[3]Phillips III, J. R., "Charged Particle Dynamics in the Lunar Environment," FY20 Proposal, NASA, Kennedy Space Center, FL, 2020 (unpublished).

[4]Hogue, M. D., Calle, C. I., Weitzman, P. S., Curry, D. R., "Calculating the trajectories of triboelectrically charged particles using Discrete Element Modeling (DEM)," *Journal of Electrostatics*, published online 2 Oct. 2007. doi:10.1016/j.elstat.2007.08

[5]Hogue, M. D., Calle, C. I., Weitzman, P. S., Curry, D. R., "Discrete element modeling (DEM) of triboelectrically charged particles: Revised experiments," *Journal of Electrostatics*, published online 25 Mar. 2009. doi:10.1016/j.elstat.2009.03.005

[6]Guo, Y., Curtis, J. S., "Discrete Element Method Simulations for Complex Granular Flows," *Annual Review of Fluid Mechanics*, published online 23 Jun. 2014. doi:10.1146/annurev-fluid-010814-014644

[7]Gale, M., Mehta, R., Liever, P., Buettner, K., Curtis, J., "Gas-Granular Flow Solver for Plume Surface Interaction and Cratering Simulations," *AIAA Computational Fluid Dynamics Conference*, published online 9 Jun. 2017. doi:10.2514/6.2017-4503

[8]Phillips III, J. R., "Spring 2020 Final Report," FY20 Proposal Follow-Up, NASA, Kennedy Space Center, FL, 2020 (unpublished).

[9]Esparza, A. B., "Charged Particle Dynamics in the Lunar Environment," Summer 2020 Abstract, NASA, Kennedy Space Center, FL, 2020 (unpublished).

[10]Phillips III, J. R., "Fall 2020 Midterm Report," FY21 CIF Proposal Follow-Up, NASA, Kennedy Space Center, FL, 2020 (unpublished).

[11]Plimpton, S. "Fast Parallel Algorithms for Short-Range Molecular Dynamics," *J Comp Phys,* 117, pp. 1-19, 1995.

[12]Kloss, C., Goniva, C., Hager, A., Amberger, S., Pirker, S. "Models, algorithms and validation for opensource DEM and CFD-DEM," *Progress in Computational Fluid Dynamics, An Int. J.,* Vol. 12, No. 2/3, 2012, pp. 140-152.

[13]Ayachit, U. *The ParaView Guide: A Parallel Visualization Application,* Kitware, 2015.

[14]Schroeder, W., Martin, K., Lorensen, B. *Visualization Toolkit: An Object-Oriented Approach to 3D Graphics,* 4th ed., Kitware, 2006.

[15]Van Rossum, G., Drake, F. L. *Python 3 Reference Manual.* CreateSpace, Scotts Valley, CA, 2009.

[16]Li, C., Xu, W., Meng, Q. "Multi-sphere approximation of real particles for DEM simulation base on a modified greedy heuristic algorithm," *Elsevier Powder Technology*, Vol. 286, December 2015, pp. 478-487. doi:10.1016/j.powtec.2015.08.026

[17]Matsumura, S., Richardson, D. C., Michel, P., Schwartz, S. R., Ballouz, R. L. "The Brazil nut effect and its application to asteroids," *Monthly Notices of the Royal Astronomical Society*, Vol. 443, Issue 4, 1 October 2014, pp. 3368-3380. doi:10.1093/mnras/stu1388

[18]Roessler, T., Katterfeld, A. "DEM parameter calibration of cohesive bulk materials using a simple angle of repose test," *Particuology*, 45, 2019, pp. 105-115. doi:10.1016/j.partic.2018.08.005

[19]"variable command," *DCS Computing, JKU Linz and Sandia Corporation*, Copyright 2016. URL: https://www.cfdem.com/media/DEM/docu/variable.html [cited 8 November 2020]

[20]Phillips III, J. R., Wang, H., Hillegass, A., Esparza, A., Dove, A. R., Elgohary, T.A. "Implementation of Charged Particle Behavior in Discrete Element Method (DEM) Simulations," *ASCE Earth and Space Conference,* 2021 (to be published).