Status update on Beans3D, a novel numerical tool to study micro- and grain-to-grain scale charging processes

Jan Deca*(1), Hsiang-Wen Hsu(1), Xu Wang(1), Zoltan Sternovsky(1), and Mihály Horányi(1)
(1) Laboratory for Atmospheric and Space Physics, University of Colorado Boulder, Boulder, CO 80303, USA

Abstract

This paper presents a status update on Beans3D, a novel numerical tool to study micro- and grain-to-grain scale charging processes. Beans3D is a fully-parallelized recursive selfconsistent particle-particle code that is based on the Barnes-Hut algorithm. The code is especially designed to provide first-principle modeling insights for various laboratory and space-based dust interaction measurements. The main features of the code are highlighted through a six-grain setup that illustrates how photoelectron and/or secondary electron emission within the microcavities of neighboring dust particles can result in large-enough repulsive forces to mobilize and loft dust grains.

1 Introduction

The lunar surface is covered by a relatively thin layer of dust particles, called the regolith, which can be disturbed by natural processes, such as meteoroid impacts, electrostatic lofting, or by robotic and/or human exploration activities [1]. The Apollo astronauts reported that lunar dust particles can readily stick to surfaces, such as spacesuits, optical lenses and thermal blankets, causing a wide series of potential problems [2, 3].

A critical element that drives the behavior of lunar dust is its interaction with ultraviolet (UV) radiation and the local plasma environment, which leads to electrostatic charging, mobilization, and transport of dust grains. One of the most well-known examples of this process is the lunar horizon glow, first observed by the Surveyor missions, and believed to be forward-scattered sunlight from a cloud of dust particles that are naturally lofted and transported by the small-scale surface electric fields [4, 5].

Recently, a ‘patched charge model’ was proposed to explain the micro- and grain-to-grain scale charging processes that affect insulating dust particles on a dusty surface [6]. The experimentally-verified model clarified that within the microcavities between the regolith particles, electron impacts and UV radiation drive the generation of photo- and secondary electrons. These electrons can be absorbed by other surfaces in the cavity, which leads to the buildup of patches of substantial negative charge within the microcavity that can result in an electrostatic repulsion force strong enough to overcome the cohesive forces and launch the particle from the surface.

From a numerical point of view, the current capabilities to study the dynamics of grain-scale charging processes and their interactions are limited due to the need to selfconsistently merge several orders of magnitude in length and time. For example, it was shown that grain-grain charge differences can exceed classical sheath predictions and reach electric field magnitudes close to the dielectric breakdown level [7]. In support of the patched charge model, a complementary study suggested that dust ejection by electrostatic forces requires soil irregularities [8].

In this proceedings, a status update on Beans3D is presented. Beans3D is a novel numerical tool to study micro- and grain-to-grain scale charging processes.

2 Numerical Methods

Beans3D is a recursive selfconsistent particle-particle code that is based on the Barnes-Hut approximation algorithm for n-body simulations [9]. The framework is written in C++ using an object-oriented approach and fully parallelized using the Message Passing Interface (MPI) [10, 11].

Recursive Field Solver and Particle Mover

To make use of advanced multi-core/multi-node distributed-memory computing systems, the simulated three-dimensional physical region of Beans3D is divided into distinct Cartesian sections using a uniform domain decomposition approach. This allows for the Barnes-hut octree to be constructed along the same boundaries as the Cartesian sections that distribute the MPI processes. During the base step of the recursive Barnes-Hut algorithm, all MPI processes execute the same workload. In the subsequent recursive steps further down the octree, the MPI processes are divided equally among the number of branches on that level. When the available MPI processes can no longer be divided further to meet the number of branches on a given level, the recursive algorithm continues within the single MPI process allocated for that particular Cartesian section. The advantage of this approach is that each MPI process needs to communicate information on the plasma particle and charge density only on a need-to-know basis while still being fully aware of the general simulation parameters and surface/grain-segment setup (see also below). The possibility to inject computational particles that are sampled from a Maxwellian distribution with user-defined densities, temperatures, and velocities at the boundaries of the computational domain allows for a clean flowing or steady-state stationary plasma within the simulated region.

The recursive Barnes-Hut field solver computes the electric potential and electric field using a brute force, direct-sum calculation when the distance between the respective plasma particles and/or surface/grain segments is smaller than a distance d. The parameter d is set by the user and determines the accuracy of the solver on the smallest scales. On spatial scales greater than d, the fast multipole method is employed to reduce the need for computational resources [12].

The code advances the trajectories of all computational particles using a second-order leapfrog algorithm. The Lorentz force acting on the plasma particles is interpolated from the computed electric field, combined with any external static electric and magnetic fields the user may have set [13]. When a particle trajectory intersects an object, the particle is deleted and its charge is added to the respective object segment or re-emitted as a secondary electron if this functionality is enabled. A particle is also deleted when its new position is located outside the bounds of the computational domain, unless periodic boundary conditions have been selected. In the latter case, the particle is re-entered at the opposite side of the domain. Finally, if the simulation stopping criteria are not reached (see below), the new plasma particle locations and segment charge distributions continue as the input data for the next computational cycle.

Secondary Particle Effects

Currently, four different plasma populations are available in Beans3D. These are free ions and electrons (e.g., flowing solar wind protons and electrons), photoelectrons resulting from UV illumination, and secondary electrons created when an incident electron is absorbed by an object segment.

Photoelectrons can be emitted from all sunlit surfaces within the computational domain. Its tangential velocity components (v_{t1}, v_{t2}) are sampled from a Maxwellian distribution, and the velocity component normal to the emitting surface (v_n) is sampled from a half-Maxwellian distribution using an acceptance-rejection method [14, 15]:

$$f(v_{t1}, v_{t2}) = \left(\frac{m}{2\pi kT}\right) \exp\left(-\frac{m(v_{t1}^2 + v_{t2}^2)}{2kT}\right) \quad (1)$$

$$f(v_n) = \left(\frac{m}{2\pi kT}\right)^{1/2} \exp\left(-\frac{m(v_n^2)}{2kT}\right), \quad (2)$$

where k is the Boltzman constant, and m and T are the electron mass and photoelectron temperature, respectively. A standard ray-casting algorithm is run as part of the initial simulation setup to determine which object segments are illuminated. The photoelectron yield on each segment is set taking into account the segment area and the computational time step. The yield is also scaled by a factor $\cos\theta$, with θ defined as the angle the incident sunlight and the surface normal.

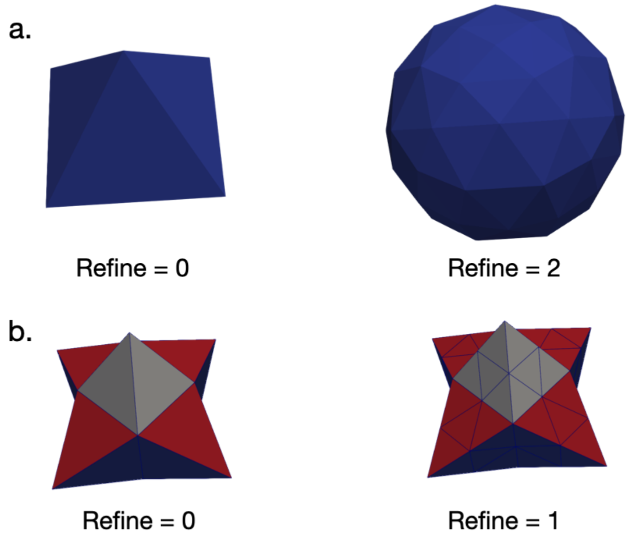

Figure 1. Illustration of the mesh refinement algorithm. a. A spherical grain defined using eight equilateral triangles (left) and the twice refined version (right). b. An irregularly-shaped grain in its most basic shape (left) and the once refined version (right).

The secondary electron distribution is modeled by adopting an empirical expression for the emission yield [16]:

$$\delta_e(E,\theta) = \delta_{e_{max}} \frac{E}{E_{max}} \exp\left(2 - 2\sqrt{\frac{E}{E_{max}}}\right) \exp\left(2(1-\cos\theta)\right), \quad (3)$$

where the primary electron impacts with an energy E at an angle θ, measured from the surface normal. $\delta_{e_{max}}$ and E_{max} are the maximum yield and energy, respectively. Note that in the latter case an incident electron is not absorbed by the object segment, but rather re-emitted as a member of the secondary electron population [17].

Object Design

A triangular boundary-element approach is used to construct segmented objects, such as surfaces and dust grains (Figure 1). These can be designed without any numerical restrictions, because the adopted numerical methods outlined above are not bound by a predefined fixed grid. The resolution of the field solver and particle mover can be increased by decreasing the parameter d and the numerical time step dt, respectively. When constructing an object, it is sufficient to define the most basic triangle structure needed to represent the shape. For example, a spherical dust grain can be formed with only six points that are connected with eight equilateral triangles. A mesh refinement algorithm then refines the basic shape to meet the minimum spatial resolution set by the user. Two different strategies are available within Beans3D that split all basic triangles of the object into four smaller triangles, either by defining the new corners at the halfway intersections of each side, or by pushing the new corners towards a better approximation of a predefined analytical shape. Note that currently the code does not support redistributing collected segment charges among other objects or object segments, for example to ensure an equipotential surface or establish an electrical coupling between objects.

Stopping Criteria

Assuming that the maximum number of time steps allowed by the user has not been reached, at the end of each computational cycle Beans3D evaluates a force-balance model that includes cohesion, adhesion, and microgravity effects to decide whether any of the modeled objects (or grain agglomerates) detach from each other or from the surface boundary. If so, the simulation is halted and the lofting parameters, such as the acceleration vector, are computed and written to file together with the electromagnetic fields and segment charge information in a standardized hdf5 format. The code also has the ability to create a hdf5 time stamp that contains all relevant information at predefined intervals while the program is being executed.

The Coulomb force between two objects is assessed using a direct-sum calculation that compiles the respective charges into a force vector. This process is repeated for all possible object combinations. Each combination is then compared to the sum of the cohesion/adhesion and microgravity forces to decide whether or not an object (or partial grain agglomerate) is mobilized. If so, the numerical paradigm to model objects as immovable is no longer valid and the simulation is stopped after computing the lofting parameters and completing the remaining data output steps.

3 Proof-of-Principle Simulation

As a first benchmark test, a proof-of-principle simulation has been developed that provides direct numerical evidence for the patched charge model. The initial setup contains six insulating spherical dust grains with a diameter of 100 µm that are exposed to a plasma flow representative of the solar wind conditions at 1 astronomical unit from the Sun (Figure 2). The top three grains are placed such that a microcavity is formed on top of a regolith layer represented by the three grains at the bottom. The topmost grain casts a partial shadow on the underlying grains, which locally stifles the generation of photoelectrons in the leftmost part of the microcavity (Figure 2a).

Three plasma species are included in the simulation: flowing solar wind protons and electrons are injected from the top boundary, and photoelectrons are created at the illuminated grain segments. Note that only the relevant part of the simulation domain is shown in Figure 2. The total computational volume measures $1000 \times 1000 \times 1000\,\mu m^3$ and the plasma distributions for the three species are represented by 50,000 computational particles in total. The benchmark test completed in less than one hour using eight Intel Xeon W cores at 2.5 Ghz. For simplicity reasons, secondary electron emission has been omitted from this proof-of-principle test.

The test simulation reached its stopping criteria after approximately 10,000 computational cycles. The final grain

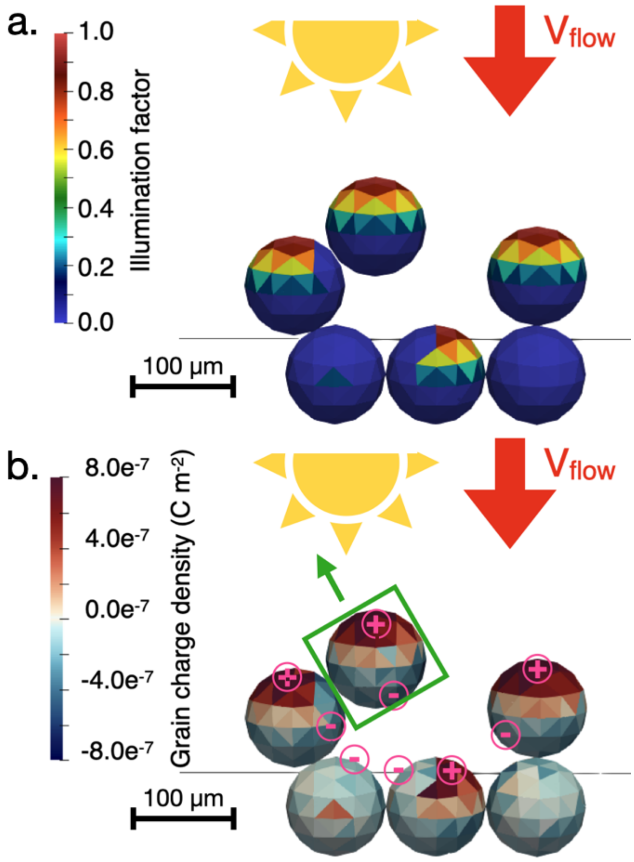

Figure 2. Proof-of-principle benchmark using six spherical insulating dust grains. a. Grain illumination as determined by the ray-casting algorithm. b. Final charge state of the configuration when the stopping criteria has been met. The grain determined by the algorithm to loft is indicated by the green box, showing also the acceleration vector.

charge state is shown in Figure 2b. As expected for the lunar near-surface environment, the photoelectron-emitting surface segments are charged positive. The surface segments within the microcavity that are not directly exposed to sunlight, on the other hand, charged negative as photoelectrons generated within the cavity are unable to escape and are recaptured primarily on the underside of the topmost grains. This creates an electric field that eventually becomes large enough to break the cohesion forces between the top grain and its underlying neighbor, causing it to loft and move away from the initial configuration. This proof-of-principle test shows the viability of the mechanism proposed by the patched charge model from a numerical perspective.

4 Concluding Remarks

Beans3D is a new particle-particle code that uses an intertwined approach between the domain decomposition prescribed for the MPI parallelization and its Barnes-Hut field solver algorithm. The code aims to provide the user with the option to simulate micro- and grain-to-grain scale charging processes from first-principle. This is highlighted through a proof-of-principle simulation, in which a configuration of six spherical grains is exploited to show that insulating dust particles on a regolith surface can be mobilized when the dust is exposed to photo- and/or secondary electron emission and when microcavities form between neighboring dust particles below the surface. It also shows the potential of the code to provide valuable new insights for various laboratory and space-based dust interaction measurements.

5 Acknowledgements

This work was supported by NASA's Early Stage Innovation (ESI) Appendix of SpaceTech-REDDI, Grant No. 80NSSC21K0226, and the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing (NAS) Division at Ames Research Center.

References

[1] H.-W. Hsu and M. Horányi, "Ballistic motion of dust particles in the lunar roving vehicle dust trails," *American Journal of Physics*, vol. 80, no. 5, pp. 452–456, May 2012. [Online]. Available: http://dx.doi.org/10.1119/1.3699957

[2] J. R. Gaier, "The Impact of Dust on Lunar Surface Equipment During Apollo," in *The Impact of Lunar Dust on Human Exploration*, vol. 2141, Feb. 2020, p. 5002. [Online]. Available: https://www.hou.usra.edu/meetings/lunardust2020/pdf/5002.pdf

[3] J. T. James and N. Khan-Mayberry, *The Human Research Program Evidence Book*, ser. NASA-SP-2009-3045. NASA, 2009, ch. Risk of adverse health effects from lunar dust exposure. [Online]. Available: https://humanresearchroadmap.nasa.gov/evidence/reports/lunar%20dust.pdf

[4] J. J. Rennilson and D. R. Criswell, "Surveyor Observations of Lunar Horizon-Glow," *Moon*, vol. 10, no. 2, pp. 121–142, Jun. 1974. [Online]. Available: https://doi.org/10.1007/BF00655715

[5] J. E. Colwell, S. Batiste, M. Horányi, S. Robertson, and S. Sture, "Lunar surface: Dust dynamics and regolith mechanics," *Reviews of Geophysics*, vol. 45, no. 2, p. RG2006, Jun. 2007. [Online]. Available: https://doi.org/10.1029/2005RG000184

[6] X. Wang, J. Schwan, H.-W. Hsu, E. Grün, and M. Horányi, "Dust charging and transport on airless planetary bodies," *Geophysical Research Letters*, vol. 43, pp. 6103–6110, Jun. 2016. [Online]. Available: https://doi.org/10.1002/2016GL069491

[7] M. I. Zimmerman, W. M. Farrell, C. M. Hartzell, X. Wang, M. Horanyi, D. M. Hurley, and K. Hibbitts, "Grain-scale supercharging and breakdown on airless regoliths," *Journal of Geophysical Research: Planets*, vol. 121, no. 10, pp. 2150–2165, 2016. [Online]. Available: https://doi.org/10.1002/2016JE005049

[8] A. Champlain, J.-C. Matéo-Vélez, J.-F. Roussel, S. Hess, P. Sarrailh, G. Murat, J.-P. Chardon, and A. Gajan, "Lunar dust simulant charging and transport under uv irradiation in vacuum: Experiments and numerical modeling," *Journal of Geophysical Research: Space Physics*, vol. 121, no. 1, pp. 103–116, Jan 2016. [Online]. Available: http://dx.doi.org/10.1002/2015JA021738

[9] J. Barnes and P. Hut, "A hierarchical O(N log N) force-calculation algorithm," *Nature*, vol. 324, pp. 446–449, Dec. 1986. [Online]. Available: https://doi.org/10.1038/324446a0

[10] P. Lucas and A. . T. B. Laboratories, *The C++ Programmer's Handbook*, ser. Ellis Horwood Books in Organic. Prentice Hall, 1992. [Online]. Available: https://isbnsearch.org/isbn/9780131182332

[11] F. Nielsen, *Introduction to MPI: The Message Passing Interface*. Springer International Publishing, 2016, pp. 21–62. [Online]. Available: https://doi.org/10.1007/978-3-319-21903-5_2

[12] L. Greengard and V. Rokhlin, "A Fast Algorithm for Particle Simulations," *Journal of Computational Physics*, vol. 73, no. 2, pp. 325–348, Dec. 1987. [Online]. Available: https://doi.org/10.1016/0021-9991(87)90140-9

[13] C. Birdsall and A. Langdon, *Plasma physics via computer simulation*. McGraw-Hill, 1985. [Online]. Available: https://isbnsearch.org/isbn/9780070053717

[14] K. L. Cartwright, J. P. Verboncoeur, and C. K. Birdsall, "Loading and Injection of Maxwellian Distributions in Particle Simulations," *Journal of Computational Physics*, vol. 162, pp. 483–513, Aug. 2000. [Online]. Available: https://doi.org/10.1006/jcph.2000.6549

[15] J. von Neumann, *John von Neumann, Collected Works*. Pergamion, 1963, vol. 5, p. 768. [Online]. Available: https://isbnsearch.org/isbn/9780080095660

[16] D. Hastings and H. Garrett, *Spacecraft-Environment Interactions*. Cambridge University Press, Aug. 2004. [Online]. Available: https://isbnsearch.org/isbn/9780521607568

[17] J. Deca, G. Lapenta, R. Marchand, and S. Markidis, "Spacecraft charging analysis with the implicit particle-in-cell code iPic3D," *Physics of Plasmas*, vol. 20, no. 10, p. 102902, Oct. 2013. [Online]. Available: https://doi.org/10.1063/1.4826951