

# Small-Body Lander Simulations Using the GPU

By Stefaan VAN WAL,<sup>1)</sup> Robert REID,<sup>2)</sup> and Daniel SCHEERES<sup>1)</sup>

<sup>1)</sup>Colorado Center for Astrodynamics Research, University of Colorado Boulder, Boulder, Colorado, USA

<sup>2)</sup>Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, USA

(Received June 21st, 2017)

We demonstrate the application of a contact model for the simulation of small-body lander/hopper spacecraft using the parallelization capabilities of a high-end video card (GPU). The target body surface is represented implicitly, which enables fast distance computations between high-resolution shape models and the spacecraft. The motion of the latter is propagated relative to a rotating target-fixed frame. Contacts between the spacecraft and that target are modeled using a distributed normal force and torque. Friction and rolling resistance forces and torques are applied at an effective application point, and regularized to smooth transitions between stick and slip. Simple examples of a cube bouncing and settling on a planar surface are used to verify the model; its potential applications are demonstrated using simulations of a MINERVA-II hopper analog deployed to comet 67P/Churyumov-Gerasimenko.

**Key Words:** asteroid, comet, lander, simulation, contact

## 1. Introduction

Asteroids and comets, known collectively as the *small bodies* of our Solar System, have been receiving ever-increasing attention as potential targets for both Discovery- and New Frontiers-class space missions. They are expected to provide a window through time into the conditions of the early Solar System, and may teach us about its formation and evolution.<sup>1)</sup> Similarly, these missions may provide a proving ground for planetary defense strategies<sup>2)</sup> and in-situ resource utilization techniques.<sup>3)</sup> The science returns from past missions were provided primarily through remote sensing operations from an orbiting spacecraft; this return may be increased significantly by including a lander and/or surface mobility operations. However, the delivery of a lander to the surface of a small body is often challenging, as experienced by both the Philae and Minerva landers.<sup>4,5)</sup> To convince future mission designers to include landers/rovers as primary payloads, we require methods to predict the settling statistics of various deployment strategies. Similarly, to demonstrate controlled mobility of a surface hopper and the effectiveness of on-board autonomous planning, high-fidelity simulation software is required.

Previous work by the authors on the motion of small-body landers has established techniques to model the complex gravitational field and surface of a small body at high resolution and low numerical burden.<sup>6,7)</sup> The presence of rocks on the small-body surface, as observed on asteroid Itokawa by the Hayabusa spacecraft, is accounted for using a procedural geometry generation technique.<sup>7)</sup> The previously developed software can be used to validate lander deployment strategies and to investigate sensitivities to uncertain parameters. Significant effects due to the lander mass distribution and shape, density of rocks on the surface, and surface interaction coefficients on the resulting deployment statistics have been identified. Although our CPU-based implementation of this software is effective, its performance is limited.

In this work, we develop techniques to perform lander/hopper simulations using graphics processing units (GPUs) instead, as they provide massive parallelization compared to a CPU. Furthermore, the GPUs architecture enables various novel modeling strategies. Instead of representing the surface with an explicit polyhedron, it is implicitly defined by a signed distance field (SDF), see also Fig. 1. The SDF allows for fast spatial queries, such as collision detection, and is well-suited to the addition of detailed surface features, such as rocks, craters, and small cracks, by procedurally distorting the SDF with multiple octaves of fractal noise.<sup>8)</sup> We demonstrate the use of a distributed compliant contact model capable of handling both collisions and continued contact of an arbitrarily-shaped lander/hopper with this surface. The model includes forces and torques from the surface normal reaction, friction, and rolling resistance forces and torques. The resulting lander/hopper simulation software enables full life-cycle simulations: from mothership release, to bouncing and settling on the surface, and to hopping surface mobility operations. We present the methodologies used in, and first results provided, by this novel technique.

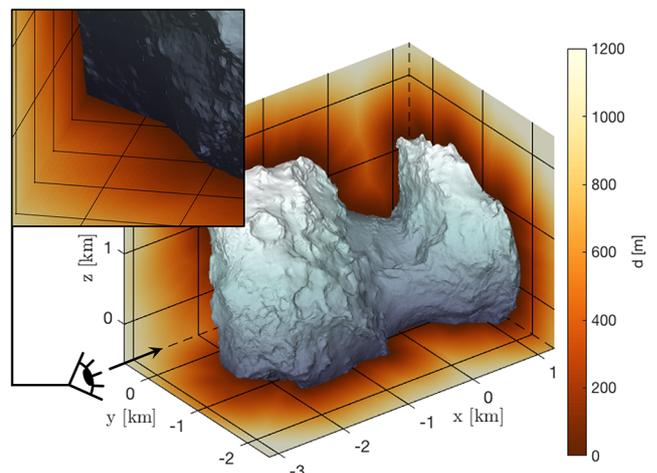


Fig. 1. Sections of the SDF of comet 67P/C-G.

## 2. Equations of Motion

We model the motion of some rigid *spacecraft* with arbitrary shape and inertia. The spacecraft moves in the neighborhood of a targeted small body, denoted the *target*, also with arbitrary shape (see also Fig. 2). The target is assumed to rotate uniformly about its axis of major inertia. The state  $\mathbf{X}$  of the spacecraft-target system is represented using six vectors, which together contain 20 state variables:

$$\mathbf{X} = [\mathbf{x} \quad \mathbf{v} \quad \mathbf{q} \quad \boldsymbol{\omega} \quad \mathbf{Q} \quad \boldsymbol{\Omega}]^T \quad (1)$$

The  $\mathbf{x}$  and  $\mathbf{v}$  vectors are the spacecraft center of mass position and velocity, expressed relative to the target, *i.e.*, relative to a rotating, non-inertial reference frame fixed to the small-body center of mass. The spacecraft attitude is also expressed relative to the target, using the quaternion  $\mathbf{q}$  and corresponding angular velocity  $\boldsymbol{\omega}$ . Finally, the attitude of the target is expressed relative to an inertial frame using the quaternion  $\mathbf{Q}$  and corresponding angular velocity  $\boldsymbol{\Omega}$ . These state vectors are illustrated in Fig. 2.

The equations of motion expressing the change of these state variables are given as:

$$\dot{\mathbf{X}} = \begin{bmatrix} \mathbf{v} \\ \mathbf{g}(\mathbf{x}) - [\tilde{\boldsymbol{\Omega}}][\tilde{\boldsymbol{\Omega}}]\mathbf{x} - 2[\tilde{\boldsymbol{\Omega}}]\mathbf{v} + \frac{1}{m} \sum \mathbf{F}_c \\ \frac{1}{2}[\mathbf{B}(\mathbf{q})][0 \quad \boldsymbol{\omega}]^T \\ -[\mathbf{I}_S]^{-1}[\tilde{\boldsymbol{\omega}}^*][\mathbf{I}_S]\boldsymbol{\omega}^* - [\tilde{\boldsymbol{\omega}}^*]\boldsymbol{\omega} + [\mathbf{I}_S]^{-1} \sum \mathbf{L}_c \\ \frac{1}{2}[\mathbf{B}(\mathbf{Q})][0 \quad \boldsymbol{\Omega}]^T \\ \mathbf{0} \end{bmatrix} \quad (2)$$

where  $\boldsymbol{\omega}^*$  is the spacecraft angular velocity relative to an inertial frame, which may be computed as  $\boldsymbol{\omega}^* = \boldsymbol{\omega} + \boldsymbol{\Omega}$ . Furthermore,  $\mathbf{g}(\mathbf{x})$  is the gravitational attraction of the target on the spacecraft, and  $\sum \mathbf{F}_c$  and  $\sum \mathbf{L}_c$  are the contact forces and torques on the spacecraft. The  $[\mathbf{I}_S]$  and  $[\mathbf{I}_T]$  matrices are the inertia matrices of respectively the spacecraft and the target, and  $m$  is the spacecraft mass.

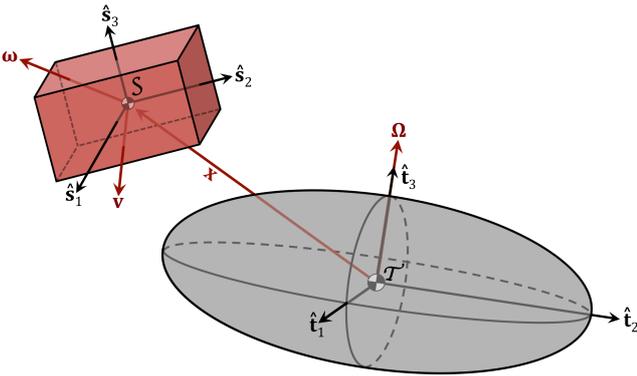


Fig. 2. Illustration of the applied state variable vectors.

## 3. Signed Distance Field

The irregular shape of small bodies is usually modeled *explicitly* using a polyhedron, a closed surface consisting of vertices connected by triangular facets. High-resolution small-body polyhedron models consist of a large number of facets; consider for example the 1.9-million facet model of comet 67P/Churyumov-Gerasimenko (67P/C-G) shown in Fig. 1. The

polyhedron can also be used to capture the irregular gravitational field of a small body, when assuming a constant density (see Werner and Scheeres<sup>9</sup>). The ability to capture both shape and gravity makes the polyhedron model a useful tool in the simulation of small-body lander/hopper spacecraft. For example, consider the lander simulation work by Tardivel et al.<sup>6</sup> and Van wal et al.<sup>7</sup>

Despite the successful implementations of the aforementioned works, the polyhedron suffers from limitations due to its explicit nature. Collision detection between a spacecraft and the polyhedral surface requires distance computations between the two. When performed naively, the model resolution may render these distance computations numerically burdensome. Furthermore, even high-resolution models such as that of Fig. 1 are limited in terms of the smallest resolved surface features. Although a collection of smaller polyhedra may be used to populate the surface with rocks, this inevitably further increases the computational cost of collision detection.<sup>6,7</sup> In an attempt to avoid these limitations, we instead represent the small-body surface *implicitly* using a *signed distance function*  $d(\mathbf{p})$  with the following properties:

$$d(\mathbf{p}) = s \cdot \min_{\mathbf{c} \in \mathcal{T}} \|\mathbf{p} - \mathbf{c}\| \quad \text{with } s = \begin{cases} > 0 & \text{if } \mathbf{p} \text{ inside} \\ 0 & \text{if } \mathbf{p} \text{ on surface} \\ < 0 & \text{if } \mathbf{p} \text{ outside} \end{cases} \quad (3)$$

In words, this function provides the signed minimum distance between a point  $\mathbf{p}$  and all points  $\mathbf{c}$  on the surface of the target  $\mathcal{T}$ . When using a trivial shape such as a sphere or ellipsoid, the corresponding signed distance functions are (relatively) simple. Unfortunately, arbitrary bodies do not have a closed-form surface expression that can be molded into a distance function. In this case, we can make use of a *signed distance field* (SDF). An SDF is a three-dimensional grid surrounding a body, with its mesh points storing the minimum signed distance between the respective points and the body.<sup>10</sup> As this only samples the signed distance at those particular points, an interpolation method must be applied to evaluate the SDF at intermediate points. Different interpolators exist; we use the simple and fast trilinear interpolator illustrated in Fig. 3. This figure shows the 8 mesh points  $d(i, j, k)$  of the SDF that define the cuboid region containing  $\mathbf{p}$ . By interpolating these 8 values, we can obtain the minimum signed distance at  $\mathbf{p}$ . Furthermore, the gradient of  $d(\mathbf{p})$  yields the direction of the surface normal  $\hat{\mathbf{N}}_p$  through  $\mathbf{p}$ .<sup>11</sup>

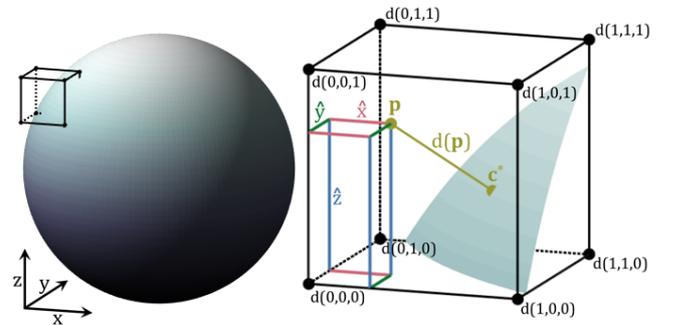


Fig. 3. Illustration of the trilinear interpolation of a signed distance field.

In order to generate the initial SDF mesh point values  $d(i, j, k)$ , we make use of the open-source *SDFGen* tool by Batty.<sup>12</sup> This allows us to sample the signed distance function

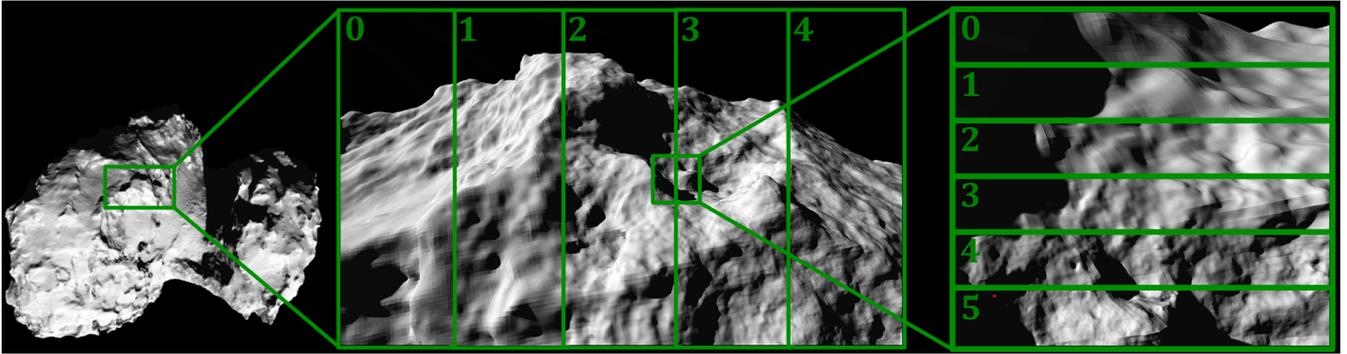


Fig. 4. Sample fBm noise generation on the SDF of comet 67P/C-G, for varying noise octaves.

of a polyhedron model at the desired mesh points, and apply the trilinear interpolator to the resulting SDF. As an example, we show slices of the SDF of comet 67P/C-G, in Fig. 1. The coloring of the three planar slices in this figure is indicative of the minimum distance to the comet surface, which was obtained by sampling its SDF. For more details on the construction of this SDF, the reader is referred to Refs. 10, 13).

When using a polyhedron shape model, distance computation requires iteration over a set of surface features. Generally speaking, the numerical burden of this iteration increases when the resolution of the underlying shape is increased. In contrast, a distance computation with the SDF involves only a single (cheap) evaluation of the interpolator. The cost of this computation is independent of the SDF resolution. Although the use of a higher resolution SDF will increase the pre-computational cost, the simulation-time effort of a low- and high-resolution model will be the same (if noise is not added). This allows us to perform fast simulations on high-resolution small-body surface models.

Furthermore, the SDF is well-suited for the inclusion of procedurally-generated noise. Various noise generation algorithms exist; see for example Ebert et al.<sup>8)</sup> We apply the commonly-used *fractional Brownian motion* (fBm). This allows us to perturb the  $d(\mathbf{p})$  values returned by an SDF with continuously varying noise, at different frequencies and amplitudes. These values can be tuned to match the surface features and irregularities observed on small bodies, allowing for the procedural generation of such features onto an SDF.<sup>14)</sup> We provide an example of procedurally-generated fBm noise on the SDF of comet 67P/C-G in Fig. 4. This figure illustrates how gradually smaller surface features are generated onto the surface as the number of noise *octaves* is increased. Rather than explicitly storing these features using *e.g.* vertices and facets, this method simply stores the parameters of the underlying distribution and generates the features ‘on-the-fly’ using random numbers. By locally varying the noise parameters across the surface of a body (possibly using additional three-dimensional grids) we can include regions of varying topography, *e.g.* sandy and rocky areas.

#### 4. The Contact Model

The model we have developed generates a distributed compliant soft contact response, using normal, friction, and rolling resistance forces and torques. In contrast to hard contact mod-

els, it robustly handles both impulsive collisions and continued, ‘rolling’ contact phases, while resolving both slip and stick behavior in the tangential spacecraft velocity. In order to generate a distributed force response, the spacecraft shape is defined through a collection of vertices assumed to be uniformly spread across its surface, as illustrated in Fig. 5.

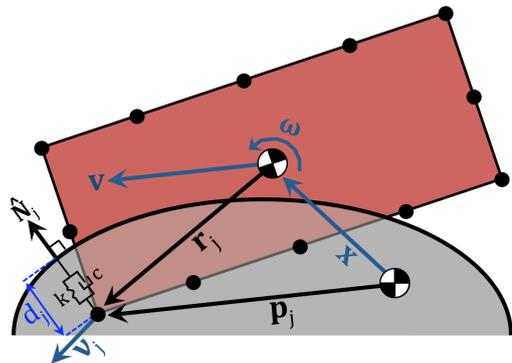


Fig. 5. Illustration of the contact geometry and normal force definition.

The position of each vertex  $\mathbf{p}_j$  is defined relative to the spacecraft center of mass using the vector  $\mathbf{r}_j$ , where  $\mathbf{p}_j = \mathbf{x} + \mathbf{r}_j$ . By evaluating the SDF at a vertex position, we can determine its penetration depth  $d_j = d(\mathbf{p}_j)$  into the surface, as well as the corresponding surface normal  $\hat{\mathbf{N}}_j = \nabla d(\mathbf{p}_j)$ . This enables a computation of that vertex’s normal velocity  $v_{N,j} = (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_j)^T \hat{\mathbf{N}}_j$ . This contains components from both the linear and angular spacecraft velocities.

##### 4.1. The Normal Force

A distributed normal force is generated by attaching spring-damper units to all active vertices; this approach is commonly used in literature,<sup>15)</sup> see also Fig. 5. The magnitudes of these forces are governed by the spring and damper coefficients,  $k$  and  $c$ . The damper coefficient is modulated to only generate a force when the corresponding vertex is moving *into* the surface, such that the applied damper force can never be cohesive. The normal forces  $\mathbf{F}_{N,j}$  generate corresponding torques  $\mathbf{L}_{N,j}$ .

##### 4.2. Coulomb Friction

The Coulomb friction force reduces the tangential surface velocity  $\mathbf{v}_T$  of the spacecraft to zero. If we apply friction forces at all active vertices where more than one vertex reaches  $v_{T,j} = 0$ , a system of inequalities must be solved in order to resolve the force magnitudes. This is most commonly cast in the form

of a *linear complementarity problem* (LCP), which must be solved iteratively.<sup>16–18</sup> This methodology further requires convergence on the slip-stick transition epochs. Both the iterative solution method and transition convergence are poorly-suited for parallel implementation in a GPU environment, as these procedures are likely to stall a warp of GPU cores, significantly reducing their computation speed. In order to avoid these issues, we follow the approach by Refs. 19–21) and apply friction at a single, effective *application point*  $\mathbf{p}_p = \mathbf{x} + \mathbf{r}_p$ , defined through a weighted sum of the active vertices:

$$\mathbf{r}_p = \frac{\sum_j (d_j \mathbf{r}_j)}{\sum_j d_j} \quad (4)$$

The application point geometry and the applied forces and torques are also illustrated in Fig. 7. Assuming that the application point moves a negligible distance between successive time steps, we can express its velocity as  $\mathbf{v}_p = \mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_p$ . The Coulomb friction force opposes the tangential component of this velocity, which is defined as  $\mathbf{v}_{T,p} = (\mathbb{I}_3 - \hat{\mathbf{N}}_p \hat{\mathbf{N}}_p^T) \mathbf{v}_p$  where  $\hat{\mathbf{N}}_p = \nabla d(\mathbf{p}_p)$  is the surface normal obtained from an SDF evaluation at the application point. When  $\mathbf{v}_{T,p}$  is non-zero, the friction force opposes it as  $\mathbf{F}_{F,slip} = -\mu F_N \hat{\mathbf{v}}_{T,p}$  where  $\mu$  is the coefficient of friction and  $\hat{\cdot}$  indicates a unit vector. This force has an associated torque  $\mathbf{L}_{F,slip} = \mathbf{r}_p \times \mathbf{F}_{F,slip}$ . When  $\mathbf{v}_{T,p}$  reaches zero, the friction force reduces to its sticking form  $\mathbf{F}_{F,stick}$ . Like  $\mathbf{F}_{F,slip}$ , it acts within the tangential plane at  $\mathbf{p}_p$ , but at some reduced magnitude  $F_{F,stick} \leq \mu_s F_N$  that ensures  $\dot{\mathbf{v}}_{T,p} = 0$ . This force can be computed from an expression of the derivative of  $\mathbf{v}_p$ .

Due to the discrete time stepping of a numerical integrator, we do not automatically step onto the precise epoch of slip-stick transitions. This leads to chatter around the sticking point, in which  $\mathbf{v}_{T,p}$  continually switches direction and effectively halts integration. Although it is possible to decrease the step size as the point of stick is approached, such convergence is again poorly-suited for parallel GPU implementation. Instead, we smooth the slip-stick transitions using the regularization variable  $\eta(v)$ , illustrated on the right side of Fig. 6. Using this variable, we define the regularized friction force as  $\mathbf{F}_F^* = \eta(v_{T,p}) \cdot \mathbf{F}_{F,slip} + (1 - \eta(v_{T,p})) \cdot \mathbf{F}_{F,stick}$ . Using this scheme, the friction force is gradually smoothed from  $\mathbf{F}_{F,slip}$  to  $\mathbf{F}_{F,stick}$  as  $v_{T,p} \rightarrow 0$ . The time histories of the regularized and unregularized tangential velocity are illustrated with a dashed line on the left side of Fig. 6.

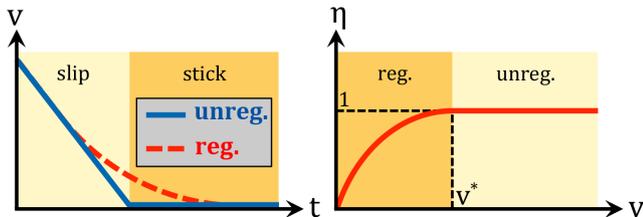


Fig. 6. Time history of (left) regularized and unregularized tangential velocity and (right) regularization variable.

Using this regularization, an integrator can *smoothly* transition between slip and stick without needing to precisely converge on the epoch of transition. Furthermore, any deviation in the tangential application point velocity resulting from errors in

the computed sticking friction force are damped out through the combination of sticking and sliding friction forces.

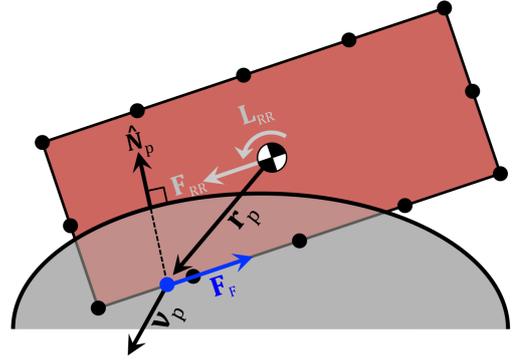


Fig. 7. Illustration of the application point, and the friction and rolling resistance forces and torques.

### 4.3. Rolling Resistance

In addition to the normal and friction forces, we also include spinning friction or *rolling resistance* in our model. Rolling resistance, as also included in *e.g.*,<sup>6, 19, 22)</sup> generates a torque that opposes the angular velocity  $\boldsymbol{\omega}$  of the spacecraft. Whenever the angular velocity  $\boldsymbol{\omega} \neq 0$  (the spacecraft is “rolling”), this torque is given as  $\mathbf{L}_{RR,roll} = -C_{RR} F_N \hat{\boldsymbol{\omega}}$  where  $C_{RR}$  is the coefficient of rolling resistance. The similarity between this formulation of rolling resistance and friction is notable; the interested reader is referred to<sup>22)</sup> for a detailed discussion of the mechanics of rolling resistance. When the spacecraft angular velocity  $\boldsymbol{\omega} = 0$ , it is said to be in *rolling stick*. In analogy with Coulomb friction, rolling resistance will act at some reduced magnitude during rolling stick to ensure the spacecraft angular velocity remains zero, *i.e.*,  $\dot{\boldsymbol{\omega}} = 0$ . An expression for this torque can be derived from the equations of motion in Eq. 2.

If the rolling resistance torque were to be applied by itself (regardless of the value of  $\boldsymbol{\omega}$ ), the tangential application point velocity would change. This would, in turn, create a coupling between the Coulomb friction and rolling resistance torques. In order to avoid this, we follow the approach of<sup>22)</sup> and also include a rolling resistance *force*  $\mathbf{F}_{RR}$ , applied in the tangential plane and at the spacecraft center of mass, such that rolling resistance has a zero net effect on  $\mathbf{v}_{T,p}$ . Finally, the rolling resistance torque is regularized near the  $\boldsymbol{\omega} = 0$  point, following the same regularization law applied to the Coulomb friction force.

## 5. Applications

By integrating the three sources of contact interactions mentioned above, *i.e.* the normal reaction, Coulomb friction, and rolling resistance forces and torques, we can implement a simulation framework for the motion of a lander/hopper spacecraft. In this section, we provide sample simulations to verify the model and demonstrate its applications.

### 5.1. Sample Motion on a Plane

As a first example, we simulate the motion of a cube impacting and settling on a plane, subject to a uniform gravity field. Vertices are placed on all eight corners of the cube. Starting from the same set of initial conditions, we perform a simulation for three relevant gravitational regimes, *i.e.*, Phobos, 67P/C-G,

and Itokawa. The resulting trajectories are plotted in Fig. 8. We observe the distinctly different scales at which trajectories in these three regimes take place. This is especially true for the Phobos-like trajectory, which covers an almost negligible distance compared to the 67P/C-G- and Itokawa-like trajectories. Nonetheless, all three exhibit a similar degree of apparent randomness, similar to a die thrown on Earth.

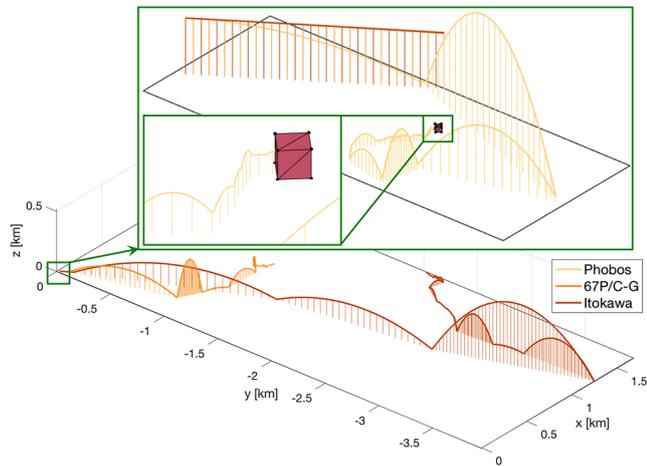


Fig. 8. Sample simulations of a cube in the considered gravitational regimes.

As discussed, we detect the settling of a spacecraft by tracking the magnitude of its linear and angular velocities. On the top of Fig. 9, we plot the variation of the spacecraft linear and angular velocity magnitude during the settling phase of the Phobos-like trajectory, where the contact phases are highlighted in blue. The final parts of this phase consist of a quick succession of several short contacts at linear velocities between roughly  $10^{-2}$  and  $10^{-5}$  m/s, during which the spacecraft velocities show significant variation. This is then followed by one final, unending contact during which both  $v$  and  $\omega$  monotonically decrease towards zero. This shows that a tracking of  $v$  and  $\omega$  successfully detects settling of the spacecraft, assuming the  $v^f$  and  $\omega^f$  values are sufficiently small.

One measure that yields insight into the mechanics of contact is the total spacecraft energy  $E$ , which consists of contributions from the gravitational potential, linear kinetic, angular kinetic, and elastic potential energies. In Fig. 10, we plot the variation of this energy during the Phobos-like trajectory. We note that, in this collision, the spacecraft touches the surface at only one of its corner vertices. The elastic potential energy increases to a maximum as the spring is compressed, and then decreases as it expands again. Corresponding variations are present in the linear and angular kinetic energies. Over the course of the collision, energy is dissipated by the forces and torques from the damper, friction, and rolling resistance. We observe that energy appears to only dissipate during the compression phase; the total energy curve remains flat during the restitution phase. This behavior appears as the dampers only dissipate energy during springs compression, and the friction force only dissipates energy while the tangential application point velocity is non-zero.

In the development of our friction model, we claimed that our regularization method is robust against variations in the application point position  $\mathbf{r}_p$ . To investigate these claims,

we modify the initial conditions of the Phobos-like trajectory such that four different vertices contact the surface over the course of some contact phase. The tangential application point velocity variation during this multi-contact situation is shown on the bottom of Fig. 9, for varying values of the friction regularization tolerance  $v_{T,p}^*$ . This figure clearly shows that the size of the slip-stick transition region decreases when  $v_{T,p}^*$  is decreased. For all tested values of  $\epsilon$ , the sliding velocity tracks the sticking point where  $v_{T,p} = 0$  more closely when the regularization tolerance  $v_{T,p}^*$  is decreased. This illustrates that our friction force regularization is indeed robust against errors in the sticking friction resulting from changes in  $\mathbf{r}_p$ .

## 5.2. Sample Motion on Comet 67P/C-G

The motion of a spacecraft impacting with and settling on a plane provides an excellent environment for verification purposes. However, it has little other use due to its simple geometry. In order to provide an example more relevant to the motion of a spacecraft in the small-body environment, we consider the MINERVA-II-1A/B spacecraft as illustrated on the right side of Fig. 11. This 1.2 kg micro-hopper is included on the Hayabusa-2 sample return mission to asteroid Ryugu, and provides a relevant test model for our simulation methodology. As the current shape model of asteroid Ryugu is limited in resolution, we choose to simulate a deployment of MINERVA-II-1A/B to comet 67P/C-G instead. We use the 25-meter resolution SDF illustrated in Fig. 1 to capture the comet's shape, and use a low-resolution polyhedron model for gravity field evaluations. The motion, which is plotted in Fig. 11, takes place in the *Imhotep* basin on the comet's large lobe. We note that although the comet's SDF was used for all simulation purposes, its polyhedron model is used for visualization in Fig. 11.

The ability to simulate the motion of an arbitrarily-shaped spacecraft to a targeted small body enables various relevant investigations. First and foremost, it allows for the simulation of specific spacecraft such as the MINERVA-II and MASCOT

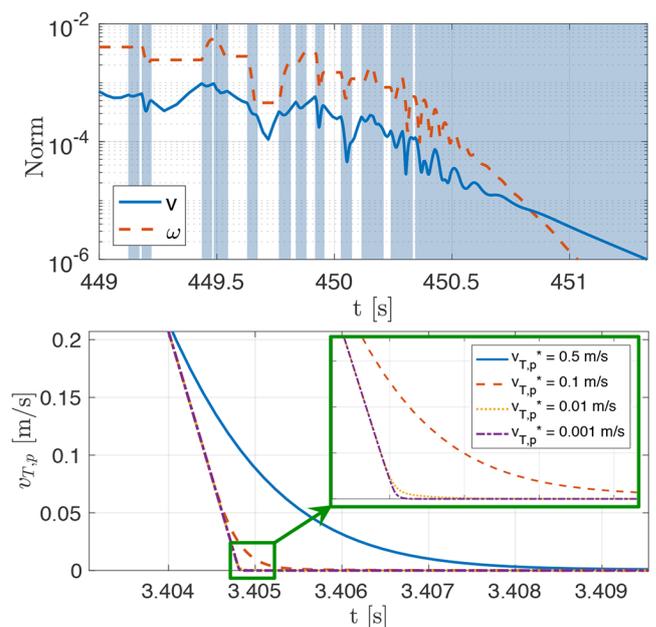


Fig. 9. (top) Velocity and (bottom) application point sliding velocity variation in the Phobos-like trajectory.

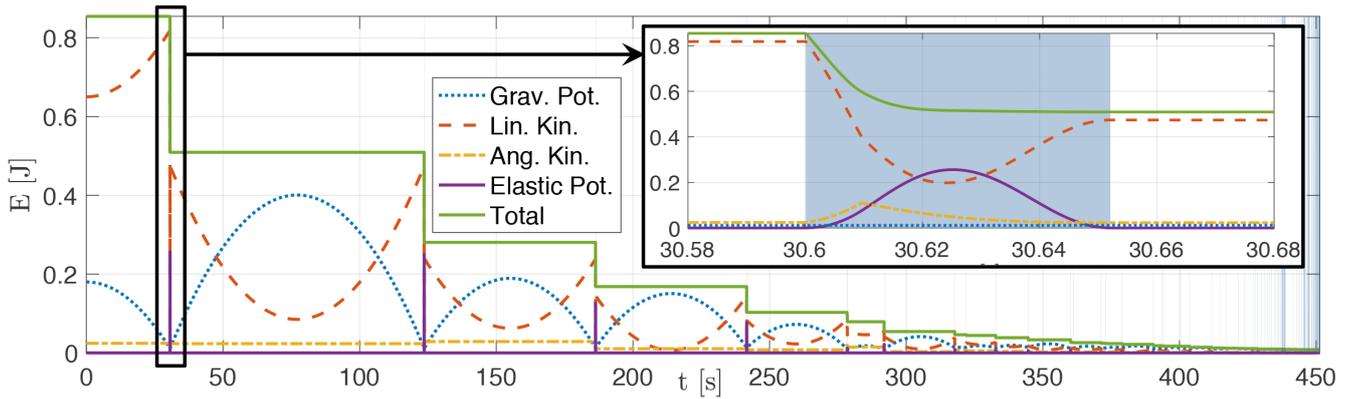


Fig. 10. Energy variation in the Phobos-like trajectory.

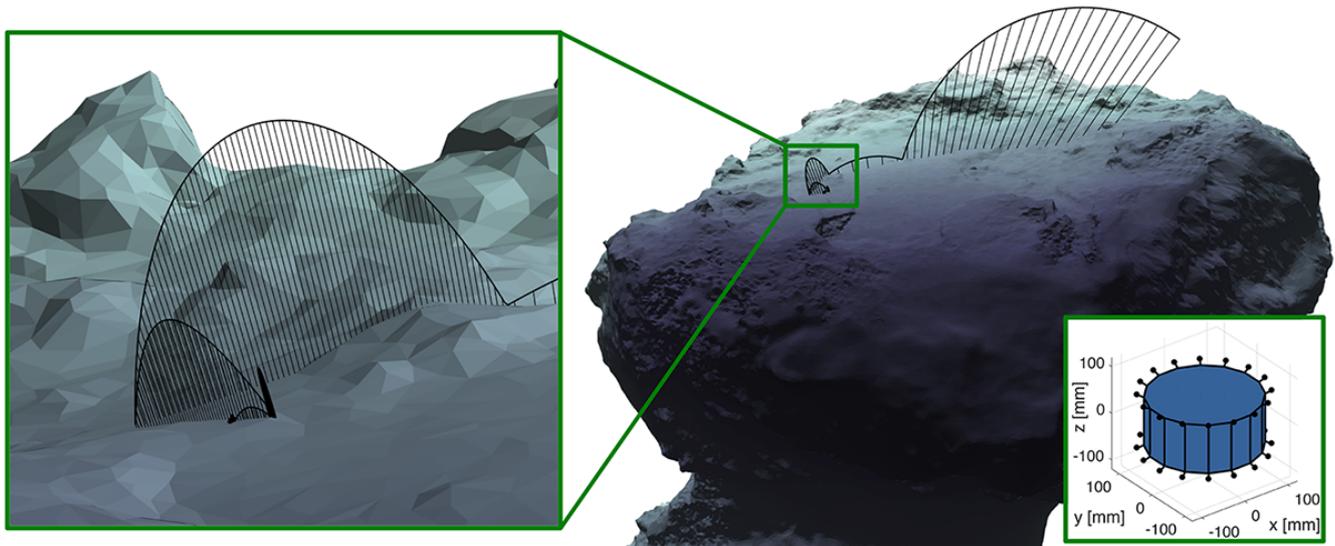


Fig. 11. Sample trajectory of the MINERVAII-1A/B hopper settling on comet 67P/C-G.

hoppers on the Hayabusa-2 mission. Taking into account the relevant uncertainties in release condition and surface interaction properties, Monte Carlo-type simulations enable prediction and analysis of *e.g.*, the spacecraft landing ellipse, settling time, and illumination conditions. At a more abstract level, a simulation framework with this contact model can be used to perform simulations where a single relevant parameter is tuned and its effects on the resulting spacecraft motion analyzed. For example, we may investigate how the spacecraft mass distribution, spacecraft shape, or surface topography affect its motion. The latter can be performed by using the previously mentioned procedural fBm noise generation techniques. This may, in turn, provide design guidelines for the hardware of a lander/hopper spacecraft.

Finally, such a framework also enables the simulation of surface mobility operations, where momentum exchange devices on board the spacecraft allow it to hop across its target and obtain scientific measurements in multiple locations, as envisioned for by the Hedgehog hopper demonstrator.<sup>23)</sup> By simulating these mobility operations, it is possible to develop and test relevant planning and control algorithms. GPU implementations of this simulation framework, which our model is well-suited for, have the potential of significantly increasing the speed at which these simulations can be performed, due to their ability to run thousands of simulations in parallel.

## 6. Conclusion

We have presented a distributed compliant contact model for the surface interactions between an arbitrarily-shaped lander/hopper spacecraft and a targeted small-body, such as an asteroid, comet, or small moon. The spacecraft motion is propagated relative to a rotating, target-fixed reference frame; its shape is represented by an inertia matrix and a collection of vertices distributed across its shell. The target surface is represented implicitly using a signed distance function, which yields the minimum distance between a spacecraft vertex and the target surface, as well as the corresponding surface normal.

A distributed normal force and torque are generated using spring-damper units attached to all spacecraft vertices. Friction and rolling resistance forces and torques are applied at a single application point, defined as the weighted sum of the active vertices. A regularization method smooths the transition between sliding and sticking and is robust against deviations in the computed sticking forces. The contact model is non-stiff and avoids the need to iteratively convergence on contact changes, rendering the model suitable for implementation on the GPU. It robustly handles both collisions and continued contact, without requiring active switching between ‘flying’ and ‘contact’ arcs. The spacecraft linear and angular velocities are tracked and used to detect the spacecraft settling on the target surface.

The integration of this contact model into a small-body lander/hopper simulation framework enables a variety of relevant applications. By simulating a particular spacecraft, such as the MINERVA-II or MASCOT hoppers on the Hayabusa-2 mission, predictions of its descent trajectory, surface dispersion, settling time, and landing site illumination may be made, in the presence of uncertainties in the release conditions and surface interaction parameters. By separately varying these parameters, we can investigate their effects on the resulting spacecraft motion. This may reveal guidelines for in the hardware and mission planning design of a small-body lander/hopper. Finally, the simulation of surface mobility operations may be used to develop planning and control strategies to reach specific targets on the small-body surface. We plan to perform these investigations in future work.

### Acknowledgments

This research was supported by NASA's SSERVI program (Institute for the Science of Exploration Targets) through institute grant number NNA14AB03A. Part of this work was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA. Support for this work was provided by the JPL Strategic University Research Partnership program.

### References

- 1) Timothy J McCoy, David W Mittlefehldt, and Lionel Wilson. Asteroid differentiation. *Meteorites and the early solar system II*, pages 733–745, 2006.
- 2) AF Cheng, P Michel, Martin Jutzi, AS Rivkin, A Stickle, O Barnouin, C Ernst, J Atchison, P Pravec, DC Richardson, et al. Asteroid impact & deflection assessment mission: Kinetic impactor. *Planetary and space science*, 121:27–35, 2016.
- 3) John S Lewis. Mining the sky: untold riches from the asteroids, comets, and planets. *Reading, Mass.: Addison-Wesley Pub. Co., c1996.*, 1, 1996.
- 4) Jens Biele, Stephan Ulamec, Michael Maibaum, Reinhard Roll, Lars Witte, Eric Jurado, Pablo Muñoz, Walter Arnold, Hans-Ulrich Auster, Carlos Casas, et al. The landing (s) of philae and inferences about comet surface mechanical properties. *Science*, 349(6247):aaa9816, 2015.
- 5) Tetsuo Yoshimitsu, Takashi Kubota, Tadashi Adachi, and Yoji Kuroda. Advanced robotic system of hopping rovers for small solar system bodies. In *Proc. 11th Int. Symp. Artif. Intel., Robot. Autom. in Space*, 2012.
- 6) Simon Tardivel, Daniel J Scheeres, Patrick Michel, Stefaan Van wal, and Paul Sánchez. Contact motion on surface of asteroid. *Journal of Spacecraft and Rockets*, 51(6):1857–1871, 2014.
- 7) Stefaan Van wal, Simon Tardivel, and Daniel Scheeres. High-fidelity small-body lander simulations. In *Proceedings of the 6th International Conference on Astrodynamics Tools and Techniques*, 2016.
- 8) David S Ebert. *Texturing & modeling: a procedural approach*. Morgan Kaufmann, 2003.
- 9) Robert A Werner and Daniel J Scheeres. Exterior gravitation of a polyhedron derived and compared with harmonic and mascon gravitation representations of asteroid 4769 castalia. *Celestial Mechanics and Dynamical Astronomy*, 65(3):313–344, 1996.
- 10) Mark W Jones, J Andreas Baerentzen, and Milos Sramek. 3d distance fields: A survey of techniques and applications. *IEEE Transactions on visualization and Computer Graphics*, 12(4):581–599, 2006.
- 11) Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.
- 12) Christopher Batty. Sdfgen, 2015.
- 13) Christian Sigg, Ronald Peikert, and Markus Gross. Signed distance transform using graphics hardware. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, page 12. IEEE Computer Society, 2003.
- 14) James M Keller, Susan Chen, and Richard M Crownover. Texture description and segmentation through fractal geometry. *Computer Vision, Graphics, and image processing*, 45(2):150–166, 1989.
- 15) Werner Goldsmith. *Impact*. Courier Corporation, 2001.
- 16) Peng Song, Jong-Shi Pang, and R Vijay Kumar. A semi-implicit time-stepping model for frictional compliant contact problems. *Departmental Papers (MEAM)*, page 47, 2004.
- 17) David E Stewart and Jeffrey C Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal for Numerical Methods in Engineering*, 39(15):2673–2691, 1996.
- 18) Peng Song and Vijay Kumar. Distributed compliant model for efficient dynamic simulation of systems with frictional contacts. In *ASME 2003 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 1009–1018. American Society of Mechanical Engineers, 2003.
- 19) Yves Gonthier, John McPhee, Christian Lange, and Jean-Claude Piedboeuf. A contact modeling method based on volumetric properties. In *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 477–486. American Society of Mechanical Engineers, 2005.
- 20) Yves Gonthier, John McPhee, Christian Lange, and Jean-Claude Piedboeuf. A regularized contact model with asymmetric damping and dwell-time dependent friction. *Multibody System Dynamics*, 11(3):209–233, 2004.
- 21) Shoichi Hasegawa and Nobuaki Fujii. Real-time rigid body simulation based on volumetric penalty method. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2003. HAPTICS 2003. Proceedings. 11th Symposium on*, pages 326–332. IEEE, 2003.
- 22) Stefaan Van wal, Simon Tardivel, Paul Sánchez, Darius Djafari-Rouhani, and Daniel Scheeres. Rolling resistance of a spherical pod on a granular bed. *Granular Matter*, 19(1):17, 2017.
- 23) Benjamin J Hockman, Andreas Frick, Robert G Reid, Issa AD Nesnas, and Marco Pavone. Design, control, and experimentation of internally-actuated rovers for the exploration of low-gravity planetary bodies. *Journal of Field Robotics*, 34(1):5–24, 2017.