rocscience

# RocFall3

# Rigid Body Dynamics

Theory Manual

# Table of Contents

# 1. Overview

In this section, I will be describing a high-level overview of RF3's rigid body physics engine. This document may evolve over time.

## 1.1. Rigid Body Simulation

Rigid bodies in RF3 are represented as closed non-degenerate triangle mesh volumes. They can have multiple contact points simultaneously with the slope surface, and thus can be influenced by multiple materials.

### 1.1.1. Rigid Body Contact Handling

Rigid body trajectories are timestepped in RF3, and the forces acting on a rigid body are determined by the contact points.

There are two separate cases for handling a contact point:

- Colliding: impact must be computed
- Separating/Resting: resting contact forces must be computed (including gravity)

If there is at least 1 colliding contact, then impact is computed, which is an instantaneous time step to update velocity. Simultaneous impact (2+ colliding contacts) is handled via an LCP formulation.

After impact, resting contacts forces such as friction and the normal force coupled with gravity must be computed. These are computed by satisfying contact constraints such as non-penetration and Coulomb friction constraints and are handled via an LCP formulation.

### 1.1.2. Rigid Body Contact Detection

Contact detection is done via a triangle mesh continuous collision detection algorithm (CCD).

## 1.2. Materials

Contact between the rock and the slope is determined by the material of the contact point(s). As of now, RF3 provides multiple ways to define materials, and in the case of overlap, there is a hierarchy that is followed.

In order of priority:

- Materials assigned to individual slope triangle IDs (ID query)
- Material regions defined by polygons in the x-y plane (position query). These may be further discretized into material textures
- Image segmentation material textures (position query)
- Default material

## 1.3. Barriers

Barriers attenuate the translational kinetic energy of a rock that passes through it. Traditional impact like we do for rock-slope contact is not computed for barrier contacts. Barriers are represented as triangle meshes. Contact between a rock and a barrier is determined by linear CCD between the rock's center of mass and the barrier's triangles.

## 1.4. Contact Detection Speedup

Contact detection is sped up through a spatial partitioning of the slope of barrier meshes using a uniform grid. This allows us to ignore contact detection with all triangles that are not intersecting with the grid nodes our rock is in. We note that this may not be the optimal solution and contact detection speedup for rigid bodies in RF3 is a major area of improvement and research for future updates.

# 2. Unconstrained Dynamics

## 2.1. Generalized Position and Velocity

The key property of rigid bodies is that they do not deform. This means that for any non-articulated rigid body, the degrees of freedom do not scale with the complexity of the geometry. These degrees of freedom are defined at the rigid body's center of mass.

We define $\mathbf{s} = \begin{bmatrix}\mathbf{x}\\ \mathbf{q}\end{bmatrix} \in \mathbb{R}^7$ as the generalized position vector for the rock where both position $\mathbf{x} \in \mathbb{R}^3$ and quaternion rotation $\mathbf{q} \in \mathbb{R}^4$**.** The position is defined to be at the rock's center of mass, while the rotation is about the rock's center of mass.

We define $\mathbf{u} = \begin{bmatrix}\mathbf{v}\\ \boldsymbol{\omega}\end{bmatrix} \in \mathbb{R}^6$ as the generalized velocity vector, where $\mathbf{v} \in \mathbb{R}^3$ is the rock's translational velocity, and $\boldsymbol{\omega} \in \mathbb{R}^3$ is the rock's angular velocity. The translational velocity is defined at the rock's center of mass, while the angular velocity is defined about the rock's center of mass.

## 2.2. Newton-Euler Equations

The Newton-Euler equations from classical mechanics generalizes the familiar *F = ma* law for the dynamics of a rigid body.

$$\begin{bmatrix}\mathbf{f}\\ \boldsymbol{\tau}\end{bmatrix} = \begin{bmatrix}mI_3 & 0\\ 0 & \mathbf{I}\end{bmatrix}\begin{bmatrix}\mathbf{a}\\ \boldsymbol{\alpha}\end{bmatrix} + \begin{bmatrix}0\\ \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}\end{bmatrix} \tag{2.1}$$

Let's break this equation down term by term:

- $\mathbf{f} \in \mathbb{R}^3$ is the total force acting on the centre of mass
- $\boldsymbol{\tau} \in \mathbb{R}^3$ is the total torque acting on the centre of mass
- $m \in \mathbb{R}^3$ is the rock mass
- $I_3$ is the 3x3 identity matrix
- $\mathbf{I} \in \mathbb{R}^{3x3}$ is the inertia matrix of the rock
- $\mathbf{a} \in \mathbb{R}^3$ is the acceleration of the centre of mass
- $\boldsymbol{\alpha} \in \mathbb{R}^3$ is the angular acceleration about the centre of mass
- $\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} \in \mathbb{R}^3$ is the inertial torque term of the rock (representing resistance to a change in angular momentum

Equation (2.1) can be further simplified with some generalized definitions:

- $\mathrm{F} = \begin{bmatrix}\mathbf{f}\\ \boldsymbol{\tau}\end{bmatrix} \in \mathbb{R}^6$ is the generalized force vector
- $\mathrm{M} = \begin{bmatrix}mI_3 & 0\\ 0 & \mathbf{I}\end{bmatrix} \mathbb{R}^{6x6}$ is the generalized mass matrix
- $\frac{\partial \mathbf{u}}{\partial t} = \begin{bmatrix}\mathbf{a}\\ \boldsymbol{\alpha}\end{bmatrix} \in \mathbb{R}^6$ is the generalized acceleration

Our Newton-Euler equation can now be simplified to the very familiar *F = ma* form:

$$\mathrm{F} = \mathrm{M}\frac{\partial \mathbf{u}}{\partial t} + \begin{bmatrix}0\\ \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}\end{bmatrix} \tag{2.2}$$

This is the fundamental equation for computing unconstrained motion, which are the projectile trajectories of the rock in RF3.

## 2.3. Symplectic Time Integration

In RF3's rigid body engine, time integration is done by first updating velocity, then using the updated velocity to update position. How to update velocity depends on multiple factors, such as contact constraints and impact. For this section, we will be focusing on unconstrained velocity updates.

### 2.3.1. Translational Velocity Update

From equation (2.1), the acceleration on the rigid body is:

$$\mathbf{f} = m\mathbf{a} \rightarrow \mathbf{a} = \frac{\mathbf{f}}{m} \tag{2.3}$$

Recall that for an explicit time integration scheme at timestep $t$ with a timestep length $h_t$, velocity can be updated via:

$$\mathbf{v}_{t+1} = \mathbf{v}_t + h_t\mathbf{a}_t \tag{2.4}$$

Define $\mathbf{g} \epsilon \mathbb{R}^3$ as the acceleration due to gravity. Using an explicit time-integration scheme with a timestep length $h$ and realizing that $\mathbf{f} = m\mathbf{g}$ in the unconstrained case, we have our unconstrained velocity update:

$$\mathbf{v}_{t+1} = \mathbf{v}_t + h\mathbf{g} \tag{2.5}$$

### 2.3.2. Angular Velocity Update

From equation (2.2), the angular acceleration of the rigid body with timestep dependencies shown is:

$$\boldsymbol{\tau}_t = \mathbf{I}_t\boldsymbol{\alpha}_t + \boldsymbol{\omega}_t \times \mathbf{I}_t\boldsymbol{\omega}_t \tag{2.6}$$

With our unconstrained motion assumption, and assumption of no external torque equivalent to gravity, we can set $\boldsymbol{\tau}_t = 0$ and then solve for $\boldsymbol{\alpha}_t$:

$$\boldsymbol{\alpha}_t = -{\mathbf{I}_t}^{-1}\boldsymbol{\omega}_t \times \mathbf{I}_t\boldsymbol{\omega}_t \tag{2.7}$$

We can then derive our angular velocity update equation:

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t + h\boldsymbol{\alpha}_t \rightarrow \boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - h{\mathbf{I}_t}^{-1}\boldsymbol{\omega}_t \times \mathbf{I}_t\boldsymbol{\omega}_t \tag{2.8}$$

### 2.3.3. Position Update

Given $\mathbf{v}_{t+1}$, we can compute our new position $\mathbf{x}_{t+1}$ through explicit time integration:

$$\mathbf{x}_{t+1} = \mathbf{x}_t + h\mathbf{v}_{t+1} \tag{2.9}$$

### 2.3.4. Rotation Update

Due to a dimensionality mismatch, it is not so straight forward to update rotation directly from the angular velocity. Instead, we need to compute the time derivative of the rotation quaternion $\mathbf{q} = [w, x, y, z]^T$. The equation for this is:

$$\dot{\mathbf{q}}_t = \frac{1}{2}\left[0, \boldsymbol{\omega}_{t_x}, \boldsymbol{\omega}_{t_y}, \boldsymbol{\omega}_{t_z}\right] * \mathbf{q}_t \tag{2.10}$$

where * is used to denote quaternion multiplication. The rotation update is then simply:

$$\mathbf{q}_{t+1} = \mathbf{q} + h\dot{\mathbf{q}}_t \tag{2.11}$$

A quaternion accurately describes a rotation only if it is a unit quaternion. To eliminate numerical error that may cause the quaternion to become non-unit, we normalize it after the update.

## 2.4. Rotation representations

We now have all the equations needed for unconstrained rigid body dynamics. Perhaps the most interesting topic here is the use of a quaternion to represent rotation. In 3D mathematics, there are three main ways we can represent rotation.

### 2.4.1. Euler Angles

The most intuitive is to specify three Euler angles for rotation about the x, y, and z-axes. There is a famous limitation of this method known as the “Gimbal lock”, where a degree of freedom can be lost.

### 2.4.2. Rotation Matrices

The next most intuitive method is to use 3x3 rotation matrices. There is both a data issue and a numerical issue with this method. The data problem is the relatively large requirement of 9 floating point numbers for rotation, which adds up quickly when we need to store each frame of animation in our RF3’s results. The numerical issue comes from the fact that rotation matrices are just a small subset of 3x3 matrices, and numerical error can easily accrue and alter our rotation matrices to something else. This can be solved by projecting the 3x3 matrix back to the special orthogonal group *SO*(3) after update, but this is also a relatively expensive operation.

### 2.4.3. Quaternions

This leads us to the method that most computer graphics practitioners have settled on, which is the use of quaternions [1]. Quaternions require only 4 floating point numbers, and their accurate representation of rotation can be ensured by normalizing the quaternion after update, which is as simple as normalizing a 4D vector.

# 3. Constrained Dynamics

In this section, we will be showing how to enforce non-penetration between a static slope and a dynamic rigid body rock in contact. We will first talk about resting contact constraints: non-penetration and friction. After, we will talk about impact.

Resting contact constraints in RF3 are based on the Siggraph 2022 course notes for contact and friction simulation [2].

## 3.1. Non-Penetration Resting Contact

### 3.1.1. Non-penetration and normal force complementarity conditions

Theoretically, we can imagine that every possible contact point between the rock and the slope has a gap function $\phi(\mathbf{s})$, where $\mathbf{s}^T = [\mathbf{x}, \mathbf{q}] \in \mathbb{R}^7$ is the generalized position vector. The gap function $\phi$ measures the distance between the rock and the slope at a potential contact point on the slope. $\phi > 0$ means they are separate, $\phi = 0$ means they are in contact, and $\phi < 0$ means the rock is penetrating the slope. One basic rule of physical realism we can follow is to constrain our rock from penetrating the slope, i.e. for all gap functions we have the constraint:

$$\phi(\mathbf{s}) \geq 0 \tag{3.1}$$

The detection of contact is a separate geometry challenge all on its own. In RF3, we have a continuous collision detection engine which we trust to give us $\phi(\mathbf{s}, t) \approx 0$ at some time t, and hope that this approximation is accurate enough.

For the purposes of this document, let's assume collision detection is perfect, so we are dealing with the $\phi = 0$ case. In this case, a contact force must be applied to make sure the rock and slope stay separate. Therefore, the constraint we must enforce is really on the rate of change of the gap function at the contact point $\dot{\phi}(\mathbf{s}, t)$, which we will shorten to just $\dot{\phi}$ :

$$\dot{\phi} \geq 0 \tag{3.2}$$

To do this, there likely needs to be a normal impulse applied in the direction of the contact normal. We work with impulses for resting forces, which is force times timestep length. (Note that this is different from the impulse in impact, because we do not assume a zero length timestep for resting and sliding.) We will call the signed magnitude of this impulse $\lambda$ and enforce the constraint that it must not be a sticking impulse, i.e. must be non-negative:

$$\lambda \geq 0 \tag{3.3}$$

There is another important physical phenomenon we need to think about here to ensure physical realism. If the rock is already separating from the slope, that is $\dot{\phi} > 0$, then the normal impulse has no reason to exist. If the normal impulse however does exist, $\lambda > 0$, it is only strong enough to prevent interpenetration. It is not an impact impulse that can directly cause separation. We can describe these two linked conditions as a complementarity condition:

$$0 \leq \dot{\phi} \perp \lambda \geq 0 \tag{3.4}$$

This a shorthand way of saying:

$$if\ \dot{\phi} > 0,\ \ then\ \ \lambda = 0$$

$$if\ \lambda > 0,\ \ then\ \ \dot{\phi} = 0 \tag{3.5}$$

which matches the physical intuition of normal forces we described. A constraint of this form is not easy to enforce with normal numerical methods for solving linear systems of equations. We will see later how we update the equations of motion while enforcing these constraints.

### 3.1.2. Non-penetration Jacobian

The gap derivative function is still a theoretical entity, so let's try to get it in a form we can compute. In practice a gap function for a point on a surface measures distance to the point in the direction normal to the surface. Knowing this and assuming $\phi = 0$ at a contact point between the rock and the slope, then $\dot{\phi}$ is equivalently described as the normal component of the contact point velocity of the rock.

$$\boldsymbol{v}_{\hat{\mathrm{n}}} = \dot{\phi} = \frac{\partial \phi}{\partial t} \tag{3.6}$$

Recall that the gap function depends on the rock's generalized position. Then,

$$\boldsymbol{v}_{\hat{\mathrm{n}}} = \frac{\partial \phi}{\partial \mathbf{s}} \frac{\partial \mathbf{s}}{\partial t} \tag{3.7}$$

We are looking to make use of the term $\frac{\partial \mathbf{s}}{\partial t}$, which at first glance may seem like the generalized velocity vector $\mathbf{u} \in \mathbb{R}^6$. Recall that the generalized position vector $\mathbf{s}$ has 7 components, so there is a dimensionality mismatch between $\mathbf{s}$ and $\mathbf{u}$. To circumvent this, we imagine a mapping between $\frac{\partial \mathbf{s}}{\partial t}$, and $\mathbf{u}$:

$$\frac{\partial \mathbf{s}}{\partial t} = \mathbf{H}\mathbf{u},\ \ \mathbf{H} \in \mathbb{R}^{7\times 6} \tag{3.8}$$

Now we have:

$$\boldsymbol{v}_{\hat{\mathrm{n}}} = \frac{\partial \phi}{\partial \mathbf{s}} \mathbf{H}\mathbf{u} = \mathbf{J}\mathbf{u} \tag{3.9}$$

thus,

$$\mathbf{J}\mathbf{u} = \boldsymbol{v}_{\hat{\mathrm{n}}} \tag{3.10}$$

where

$$\mathbf{J} \in \mathbb{R}^{1\times 6} \tag{3.11}$$

is the Jacobian, which has the purpose of mapping the rock's velocity to the normal component of the contact velocity. This Jacobian in particular is known as the non-interpenetration Jacobian $\mathbf{J}_{\hat{\mathrm{n}}}$. In the world of contact simulation, Jacobians are a tool for defining constraints in a modular way. In general, velocity-based constraints can be expressed as unilateral:

$$\mathbf{J}\mathbf{u} = 0 \tag{3.12}$$

or bilateral:

$$\mathbf{Ju} \geq 0 \tag{3.13}$$

In RF3, we only have to worry about bilateral constraints. Unilateral constraints appear in mechanical systems such as hinge joints between rigid bodies.

We still don't have an actual expression for $J_{\hat{\mathrm{n}}}$, which we will just call $\mathbf{J}$ in this subsection for brevity. We can derive $\mathbf{J}$ from the description of $v_{\hat{\mathrm{n}}}$, the normal component of velocity at the contact point, and $\mathbf{u}$, the generalized velocity vector of the rock.

The velocity at a point $\mathbf{p} \in \mathbb{R}^3$ of a rigid body is $\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}$, where $\mathbf{r} = \mathbf{p} - \mathbf{x}$ [3]. The normal component of that velocity is found with the dot product against the contact normal.

$$\boldsymbol{v}_{\hat{\mathrm{n}}} = \hat{\mathrm{n}} \cdot (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}) \tag{3.14}$$

Substitute $\boldsymbol{v}_{\hat{\mathrm{n}}}$ into equation 3.10:

$$\begin{aligned} \mathbf{Ju} &= \hat{\mathbf{n}} \cdot \mathbf{v} + \hat{\mathbf{n}} \cdot \boldsymbol{\omega} \times \mathbf{r} \\ &= \hat{\mathbf{n}} \cdot \mathbf{v} - \hat{\mathbf{n}} \cdot \mathbf{r} \times \boldsymbol{\omega} \\ &= [\hat{\mathbf{n}}^{\boldsymbol{T}}, -\hat{\mathbf{n}}^{\boldsymbol{T}} \mathbf{r}^{\times}] \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\omega} \end{bmatrix} \end{aligned} \tag{3.15}$$

$$\Rightarrow \boldsymbol{J} = [\hat{\mathbf{n}}^{\boldsymbol{T}}, -\hat{\mathbf{n}}^{\boldsymbol{T}} \mathbf{r}^{\times}] \tag{3.16}$$

where we use the cross-product matrix of $\mathbf{r} = [\boldsymbol{a} \quad \boldsymbol{b} \quad \boldsymbol{c}]^T$

$$\mathbf{r}^{\times} \equiv \begin{bmatrix} \mathbf{0} & -c & b \\ c & \mathbf{0} & -a \\ -b & a & \mathbf{0} \end{bmatrix} \tag{3.17}$$

which may look confusing, but it really is just encoding the cross-product operation into a matrix.

We also want to note that $\mathbf{J}^T$ does the reverse mapping of $\mathbf{J}$, meaning $\mathbf{J}^T(\mathbf{z} \cdot \hat{\mathbf{n}})$ maps a scalar $(\mathbf{z} \cdot \hat{\mathbf{n}}) \in \mathbb{R}$ to the rock's velocity space $\mathbb{R}^6$.

### 3.1.3. LCP Formulation

Now that we have $\mathbf{J}$, we can try formulating the non-penetration constraint with the equations of motion. First, let's just get the equations of motion in impulse terms starting from Newton's 2nd law of motion:

$$\mathbf{F} = \mathrm{M} \frac{\partial \mathbf{u}}{\partial t} \tag{3.18}$$

Where $\mathrm{M} \in \mathbb{R}^{6\times6}$ is the mass matrix which encodes both the mass and inertia of the rock.

To get the impulse equations of motion, we integrate force with respect to time. We define the generalized velocity we are solving for at the end of the timestep as $\mathbf{u}^+$. The velocity that hasn't yet felt the resting impulse is $\mathbf{u}$. We denote the timestep length as $\Delta t$.

$$\int_t^{t+\triangle t} \mathbf{F}(t)dt \approx \mathrm{M}(\mathbf{u}^+ - \mathbf{u}) \tag{3.19}$$

We group external and inertial forces into the term $\mathbf{f}$. We also recall our notation for impulse magnitudes $\lambda$, and note that $\mathbf{J}^T \lambda$ maps $\lambda$ to a rigid body impulse on the center of mass. Finally, we have:

$$\mathrm{M}\mathbf{u}^+ = \mathrm{M}\mathbf{u} + \Delta t\mathbf{f} + \mathbf{J}^T \lambda \tag{3.20}$$

The system in equation (3.20) consists of 6 equations and 7 unknowns. To be able to solve this, we need to include our contact constraints:

$$0 \leq \boldsymbol{v}_{\hat{\mathrm{n}}} \perp \lambda \geq 0 \tag{3.21}$$

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}^T \\ \mathbf{J} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^+ \\ \lambda \end{bmatrix} + \begin{bmatrix} -\mathrm{M}\mathbf{u} - \Delta t\mathbf{f} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{v}_{\hat{\mathrm{n}}} \end{bmatrix} \tag{3.22}$$

This is the standard way to represent our equations with constraints, but it still needs some work to be solved numerically.

First solve for $\mathbf{u}^+$ in the first row:

$$\mathbf{u}^+ = \mathrm{M}^{-\mathbf{1}}\mathbf{J}^T\lambda + \mathrm{M}^{-\mathbf{1}}(\mathrm{M}\mathbf{u} + \Delta t\mathbf{f}) \tag{3.23}$$

Then substitute it into the second row:

$$\mathbf{J}\mathrm{M}^{-\mathbf{1}}\mathbf{J}\lambda + \mathbf{J}\mathrm{M}^{-\mathbf{1}}(\mathrm{M}\mathbf{u} + \Delta t\mathbf{f}) = \boldsymbol{v}_{\hat{\mathrm{n}}} \tag{3.24}$$

Substitute into equation 3.21 and notice that some terms which are known at the beginning of the timestep can be grouped together:

$$\mathbf{J}\mathrm{M}^{-\mathbf{1}}\mathbf{J}\lambda + \mathbf{J}\mathrm{M}^{-\mathbf{1}}(\mathrm{M}\mathbf{u} + \Delta t\mathbf{f}) \geq \mathbf{0} \perp \lambda \geq 0 \tag{3.25}$$

$$Let \begin{cases} \mathrm{A} = \mathbf{J}\mathrm{M}^{-\mathbf{1}}\mathbf{J} \\ \mathrm{x} = \lambda \\ \mathrm{b} = \mathbf{J}\mathrm{M}^{-\mathbf{1}}(\mathrm{M}\mathbf{u} + \Delta t\mathbf{f}) \end{cases} \tag{3.26}$$

(Don't confuse this x with the position vector x) Now we have:

$$\mathrm{A}\mathrm{x} + \mathrm{b} \geq 0 \perp \mathrm{x} \geq 0 \tag{3.27}$$

This is the form of a linear complementarity problem (LCP) of which there are numerical solvers that can be used to compute a solution. LCPs have been studied well before rigid body contact constraints and were a convenient mathematical tool that early rigid body simulation researchers found useful. The algorithm that RF3 uses to solve this LCP is known as Lemke's algorithm, which is based on matrix pivoting. This specific algorithm is necessary when adding in friction constraints.

### 3.1.4. Multiple contact points

To expand this method to work with multiple contact points, we need to enforce multiple nonpenetration contact constraints. This just means we need more non-penetration Jacobians in our LCP. For $k$ contact points, we have $k$ Jacobians $\mathbf{J_1}, \mathbf{J_2}, \ldots, \mathbf{J}_k$, all $\in \mathbb{R}^{1\times 6}$ . Then let's redefine $\mathbf{J}$ as the collection of all non-penetration Jacobians:

$$\mathbf{J} = \begin{bmatrix} \mathbf{J_1} \\ \mathbf{J_2} \\ \vdots \\ \mathbf{J}_k \end{bmatrix} \epsilon \ \mathbb{R}^{k \times 6} \tag{3.28}$$

We similarly have $k$ normal impulse magnitudes $\lambda_1, \lambda_2, \ldots, \lambda_k$ corresponding to each contact point we need to solve for.

$$\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_k \end{bmatrix} \epsilon \ \mathbb{R}^{k} \tag{3.29}$$

Likewise, we now have $k$ contact normal velocities to keep track of:

$$\mathbf{v}_{\hat{\mathrm{n}}} = \begin{bmatrix} \mathbf{v}_{\hat{\mathrm{n}},1} \\ \mathbf{v}_{\hat{\mathrm{n}},2} \\ \vdots \\ \mathbf{v}_{\hat{\mathrm{n}},\mathrm{k}} \end{bmatrix} \tag{3.30}$$

Our new multi-contact definitions for $\mathbf{J}$, $\boldsymbol{\lambda}$, and $\mathbf{v}_{\hat{\mathrm{n}}}$ can be substituted into equation 3.22. This demonstrates the modularity of the solving contact constraints with the LCP formulation. You do not need to consider the case of one, two, three, or $k$ contacts as separate edge cases. Multiple contact points appear very frequently when using 3D meshes to represent our rock and slope.

## 3.2. Friction constraints

### 3.2.1. Planar Dry Friction Model

In RF3, both the rock and slope are approximated as triangle meshes. Since the main elements of triangle meshes, faces, are planar, we choose a planar friction model to model frictional contact.

### 3.2.2. Coulomb friction law

Let $\boldsymbol{\lambda}p$ denote the tangential friction impulse at a contact point. Note that it is bold because even though it is constrained to the 2D plane, it is still a vector. We use the subscript p to denote that it is a planar impulse. The theory of Coulomb friction states that $\boldsymbol{\lambda}_p$ is coupled to the normal impulse $\lambda_{\hat{n}}$. This coupling is a non-linear quadratic cone constraint:

$$\| \boldsymbol{\lambda}_p \|_2 \leq \mu \lambda_{\hat{n}} \tag{3.31}$$

The isotropic Coulomb friction law can be written as follows for the single contact case [2]:

- **Slip**: If the rock is sliding on the slope, then the direction of the friction force is opposed to the tangential relative velocity and its magnitude is $\mu\lambda_{\hat{n}}$
- **Stick**: If the rock is stationary on the slope, then the friction force can have any direction as along as it stays within the friction cone

The principle of maximal dissipation [7] [2] states that the frictional force maximally dissipates energy subject to the friction cone constraint, which gives us the result that it acts opposite to tangential contact velocity in the single contact case. To generalize this principle to the multi-contact case, the

direction of the friction forces at each contact must be such that together they maximally dissipate energy.

If we do not linearize our friction cone in equation (3.31), then we are left with a non-linear constraint which will eventually lead to a non-linear complementarity problem (see section 1.5 of [2]). To utilize the same LCP framework we used with non-penetration constraints, we must linearize the friction cone.

### 3.2.3. Linearized friction cone

The original constraint in equation (3.31) is a friction cone constraint because it can be visualized as an upside down cone. The base of the friction cone is a smooth circle. To linearize it, we approximate the non-linear circular base as a convex polygon with equal edge lengths, as shown in Figure 1.

Given two perpendicular unit vectors $\hat{\mathrm{x}}_p, \hat{\mathrm{y}}_p \in \mathbb{R}^2$ defined in the friction plane with local coordinate system at the contact point, we can sample points on the circle defined in the same coordinate system for some angle $\theta$ with:

$$\hat{\mathbf{d}}(\theta) = \cos(\theta)\,\hat{\mathrm{x}}_p + \sin(\theta)\,\hat{\mathrm{y}}_p \tag{3.32}$$

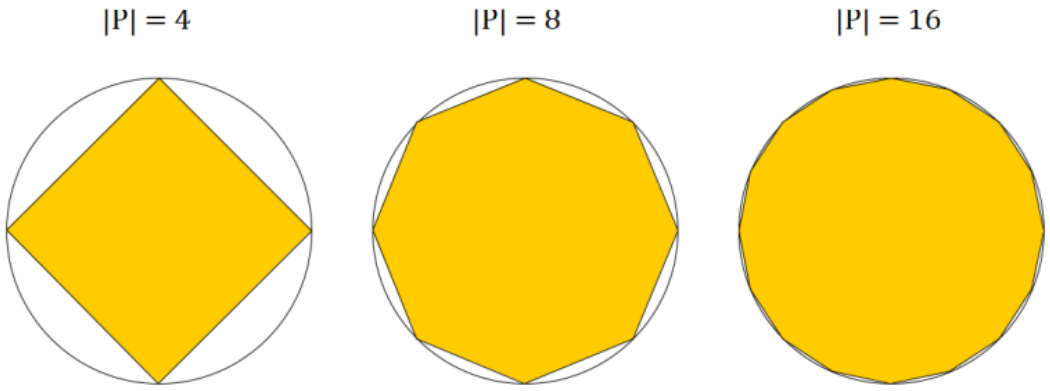


Figure 1: The base of the polyhedral friction cone with different numbers of friction directions

Let's use this to define a positive span $P = \{\hat{\mathrm{d}}_1, \hat{\mathrm{d}}_2, \ldots, \hat{\mathrm{d}}_{|P|}\}$ of the friction plane, meaning each friction plane vector is given an opposite direction counterpart. So if $\hat{\mathrm{d}}_1(\theta) \in P$, then also $\hat{\mathrm{d}}_1(\theta + \pi) \in P$. Clearly this implies $|P|$ must be an even number.

Let $\theta_0, \theta_1, \ldots, \theta_P$ be a uniform sampling of $P$+1 angles covering the circle. Then define $\hat{d}_i = \hat{d}(\theta_i)$. We now have a set of positive friction directions $\{\hat{\mathrm{d}}_1, \hat{\mathrm{d}}_2, \ldots, \hat{\mathrm{d}}_{|P|}\}$ which span the plane of friction. Assign a positive scalar $\lambda_{\hat{d}_i} \in \mathbb{R}$, $\forall i \in \{1, 2, \ldots, |P|\}$ to denote the magnitude of the frictional impulse in that direction. Then we can linearize equation (3.31) to obtain the polygonal friction cone constraint:

$$0 \leq \mu\lambda_{\hat{n}} - \sum_{\mathrm{i}} \lambda_{\hat{d}_i} \tag{3.33}$$

This linearized constraint better approximates the original nonlinear constraint as the number of friction directions increases.

### 3.2.4. Friction Complementarity Conditions

Determining the transition between stick and slip for a contact point is probably the most complicated part of simulation friction. Let $v_{\hat{d}} = \left[v_{\hat{d}_1} \dots v_{|\hat{d}_i|}\right]$ measure the sliding velocity along each friction direction vector. Since the friction direction opposes velocity by the principle of maximum dissipation, the sign of these velocities must be non-positive, that is $v_{|\hat{d}_i|} \leq 0$. Now consider a slack variable $\beta$ which measures the maximum magnitude of velocity along any of the friction directions. An equality that will give us $\beta$ is:

$$0 \leq \beta + v_{\widehat{d_i}}, \forall i \in \{1, 2, \dots, |P|\} \tag{3.34}$$

Notice that $\beta$ can only be zero if all $v_{\hat{d}_i}$ is zero. If any of $v_{\hat{d}_i} < 0$, then $\beta$ must match the largest of those magnitudes. A more compact way of stating equation (3.34) is:

$$0 \leq \beta \boldsymbol{e} + v_{\hat{d}} \tag{3.35}$$

where $\mathbf{e} \in \mathbb{R}^{|P|}$ is a $|P|$ dimensional vector of ones.

Though the slack variable $\beta$ does have a physical interpretation, what it is used for in practice is just determining the "on" and "off" state sliding friction. That is, when there is sliding ($v_{\hat{d}} \neq 0$), then $\beta > 0$, and thus the frictional impulse will be on the boundary of the friction cone following the definition of slip and principle of maximum dissipation, setting the non-strict inequality in equation (3.33) to an equality to solve for $\lambda_{\hat{d}}$.

If the frictional impulse $\lambda_{\hat{d}}$ is positive at some component, we do not yet know if there is sliding. To do that, we set the inequality in equation (3.35) to an equality to solve for $\beta$.

Finally, we arrive at the complete set of complementarity conditions, including nonpenetration, for the frictional contact case:

$$0 \leq v_{\hat{n}} \perp \lambda_{\hat{n}} \geq 0 \tag{3.36a}$$

$$0 \leq \beta \boldsymbol{e} + v_{\hat{d}} \perp \lambda_{\hat{d}} \geq 0 \tag{3.36b}$$

$$0 \leq \mu \lambda_{\hat{n}} - \textstyle\sum_{\mathrm{i}} \lambda_{\hat{d}_i} \perp \beta \geq 0 \tag{3.36c}$$

For more in-depth explanation, please refer to the Siggraph Contact and Friction Simulation Course [2].

### 3.2.5. Frictional Contact LCP Formulation

We now have the task of coupling our complementarity conditions and equations of motion into a form solvable by a computer. We begin by defining our Jacobians, which have the 20 task of mapping vectors between body space and contact space.

The non-penetration Jacobian maps between contact normal velocity and center of mass generalized velocity. For K contact points, we define the i'th non-penetration Jacobian as:

$$\mathbf{J}_{\hat{n},i} = \begin{bmatrix} \hat{n}_i^T & -\hat{n}_i^T r_i^{\times} \end{bmatrix} \in \mathbb{R}^{1\times 6} \tag{3.37}$$

We define the total non-penetration Jacobian $\mathbf{J}_{\hat{n}} \in \mathbb{R}^{K\times 6}$ as:

$$\mathbf{J}_{\hat{n},i} = \begin{bmatrix} \mathbf{J}^T{}_{\hat{n},1} & \cdots & \mathbf{J}^T{}_{\hat{n},K} \end{bmatrix}^T \tag{3.38}$$

Similarly, we need a frictional Jacobian to map center of mass generalized velocity to tangent plane velocity. For K contact points, given *h* friction directions, we define the i'th friction Jacobian as:

$$\mathbf{J}_{\hat{d},i} = \begin{bmatrix} \hat{\mathrm{d}}_1, & -\hat{\mathrm{d}}_1 r_i{}^{\times} \\ \hat{\mathrm{d}}_2, & -\hat{\mathrm{d}}_2 r_i{}^{\times} \\ & \vdots \\ \hat{\mathrm{d}}_h, & -\hat{\mathrm{d}}_h r_i{}^{\times} \end{bmatrix} \in \mathbb{R}^{h\times 6} \tag{3.39}$$

We define the total frictional Jacobian $\mathbf{J}_{\hat{d}} \in \mathbb{R}^{hK\times 6}$ as:

$$\mathbf{J}_{\hat{d},i} = \begin{bmatrix} \mathbf{J}^T{}_{\hat{d},1} & \cdots & \mathbf{J}^T{}_{\hat{d},K} \end{bmatrix}^T \tag{3.40}$$

Define the normal impulses vector $\lambda_{\hat{n}} \in \mathbb{R}^K$ and the friction impulses vector $\lambda_{\hat{d}} \in \mathbb{R}^{hK}$. Then we have our equations of motion including both normal and friction impulses:

$$\mathrm{M}\mathbf{u}^+ = \mathrm{M}\mathbf{u} + \Delta t\mathbf{f} + \mathbf{J}^T{}_{\hat{n}}\lambda_{\hat{n}} + \mathbf{J}^T{}_{\hat{d}}\lambda_{\hat{d}} \tag{3.41}$$

We can reformulate this as a matrix equation with our complementarity conditions:

$$\begin{bmatrix} M & -\mathbf{J}^T{}_{\hat{n}} & -\mathbf{J}^T{}_{\hat{d}} & 0 \\ J_{\hat{n}} & 0 & 0 & 0 \\ J_d & 0 & 0 & \mathbf{E} \\ 0 & \bar{\mu} & -\mathbf{E}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^+ \\ \lambda_{\hat{n}} \\ \lambda_{\hat{d}} \\ \beta \end{bmatrix} + \begin{bmatrix} -M\mathbf{u} - \Delta t\boldsymbol{f} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{v}_{\hat{\mathrm{n}}} \\ \mathbf{v}_{\hat{d}} \\ 0 \end{bmatrix} \tag{3.42}$$

$$0 \le v_{\hat{n}} \perp \lambda_{\hat{n}} \ge 0 \tag{3.43a}$$

$$0 \le \mathbf{E}\beta + v_{\hat{d}} \perp \lambda_{\hat{d}} \ge 0 \tag{3.43b}$$

$$0 \le \bar{\boldsymbol{\mu}}\lambda_{\hat{n}} - \mathbf{E}^T\lambda_{\hat{d}} \perp \beta \ge 0 \tag{3.43c}$$

where:

- $\mathbf{E} = diag([\mathbf{e}_1^T \quad \dots \quad \mathbf{e}_K^T])$ is a block diagonal matrix containing $\mathbf{e} \in \mathbb{R}^h$ vectors of ones
- $\bar{\boldsymbol{\mu}} = diag([\mu_1 \quad \cdots \quad \mu_K])$ is a diagonal matrix of friction coefficients for each contact.

This system can be reduced to:

$$(C + GM^{-1}G^T)\begin{bmatrix} \lambda_{\hat{n}} \\ \lambda_{\hat{d}} \\ \beta \end{bmatrix} + GM^{-1}(M\mathbf{u} + \Delta t\mathbf{f}) = \begin{bmatrix} \mathbf{v}_{\hat{\mathrm{n}}} \\ \mathbf{v}_{\hat{d}} \\ 0 \end{bmatrix} \tag{3.44}$$

where

$$C = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \mathbf{E} \\ \bar{\boldsymbol{\mu}} & -\mathbf{E}^T & 0 \end{bmatrix} \tag{3.45}$$

and

$$G^T = \begin{bmatrix} \mathbf{J}^T{}_{\hat{n}} & \mathbf{J}^T{}_{\hat{n}} & 0 \end{bmatrix} \tag{3.46}$$

Equation (3.44) with the complementarity constraints (3.43) matches the LCP formulation of

$$Ax + b \geq 0 \perp x \geq 0$$

The solution to this equation will give us the updated velocities for a rigid body sliding timestep of duration Δt without violating our complementarity constraints.

## 3.3. Impact

In this section, we describe RF3's impact algorithm which is provably energy bounded, symmetry preserving, and non-penetrating. Not coincidentally, we end up with an LCP formulation for impact.

### 3.3.1. Physical Desiderata for Correct Algorithmic Impact

We measure the quality of our impact algorithm using the criteria mentioned in "Reflections on Simultaneous Impact" [6]. These criteria are break-away (for multi-body systems), symmetry preservation, energy bounded, momentum conservation (for multi-body systems), and one-sidedness of impulses. RocFall3 deals with a single rigid body system (the rock), and boundary geometry (the slope). Therefore, two of these criteria, break-away and momentum conservation, cannot be properly assessed. We describe the three criteria will we assess below:

- (SYM) Symmetry preservation
- (KIN) Energy bounded
- (ONE) One-sided impulses.

### 3.3.2. Impact Impulse

Consider an impact with *K* planar contacts. The impulse that acts on the rigid body for each contact point must be non-negative in the normal direction of each contact. We now seek an equation that relates the impulses $\lambda_{\hat{n}_i}$ to the change in general velocity of the rigid body **u**.

Recall the non-penetration Jacobian used for resting contact:

$$\mathbf{J}_{\hat{n},i} = \begin{bmatrix} \hat{n}_i{}^T & -\hat{n}_i{}^T r_i{}^\times \end{bmatrix} \in \mathbb{R}^{1\times 6} \tag{3.47}$$

$$\mathbf{J}_{\hat{n}} = \begin{bmatrix} \mathbf{J}_{\hat{n},1} \\ \mathbf{J}_{\hat{n},2} \\ \dots \\ \mathbf{J}_{\hat{n},k} \end{bmatrix} \in \mathbb{R}^{k\times 6} \tag{3.48}$$

Recall that this Jacobian maps the generalized velocity to the contact point velocities.

$$\mathbf{J}_{\hat{n}}\mathbf{u}^+ = \mathbf{J}_{\hat{n}}^T \begin{bmatrix} v_{\hat{n},1}^+ \\ v_{\hat{n},2}^+ \\ \vdots \\ v_{\hat{n},k}^+ \end{bmatrix} \tag{3.49}$$

Likewise, the transpose of $\mathbf{J}_{\hat{n}}$ maps the contact point velocities to the generalized velocity. Furthermore, it also maps the contact point impulses to a generalized impulse on the rock's center of mass. The generalized impulse from this impact can be expressed as $M(\mathbf{u}^+ - \mathbf{u}^-)$. Now we can derive the effect of the impact impulses on the generalized impulse:

$$M(\mathbf{u}^+ - \mathbf{u}^-) = \mathbf{J}_{\hat{n}}^T \begin{bmatrix} \lambda_{\hat{n},1} \\ \lambda_{\hat{n},2} \\ \dots \\ \lambda_{\hat{n},k} \end{bmatrix} \tag{3.50}$$

Rearranging gives us the proper velocity update equation when given the vector of impact impulses $\boldsymbol{\lambda}_{\hat{\boldsymbol{n}}} \in \mathbb{R}^K$.

$$\mathbf{u}^+ = \mathbf{u}^- + M^{-1}\mathbf{J}_{\hat{n}}^T \boldsymbol{\lambda}_{\hat{\boldsymbol{n}}} \tag{3.51}$$

### 3.3.3. Coefficient of Restitution

Now that we know how impact impulses will affect our generalized velocity u, we must derive how to compute the impact impulses such that they satisfy our physical desiderata for impacts. In RocFall3 Rigid Body, we follow the simple rule of a coefficient of restitution:

$$coefficient\ of\ restitution\ (c_r) = \frac{|relative\ velocity\ after\ impact|}{|relative\ velocity\ before\ impact|} \tag{3.52}$$

Consider the case of *K* contacts. The *i*'th contact has a coefficient of restitution $c_{r,i}$ which relates its relative normal velocity before and after impact. The desiderata of (ONE) implies that a velocity after impact must have a positive relative normal component. Recall also that the impact impulse will be in the positive normal direction. Therefore:

$$v_{\hat{n},i}^- < 0 < \lambda_{\hat{n},\mathrm{i}} \ \rightarrow v_{\hat{n},i}^+ = -c_{r,i} v_{\hat{n},i}^- \tag{3.53}$$

This states that if a contact is colliding, and an impact impulse has been generated at that contact, then the input and output velocities are related via the coefficient of restitution. Note however that the impact impulse at a contact point affects the generalized velocity of the rigid body, which in turn affects other contact point velocities. Thus, it is not required that $\lambda_{\hat{n},\mathrm{i}} > 0$ for $v_{\hat{n},i}^+ = -c_{r,i} v_{\hat{n},i}^-$. This leads us to our complementarity condition:

$$0 \leq \lambda_{\hat{n},\mathrm{i}} \ \perp \ v_{\hat{n},i}^+ + c_{r,i} v_{\hat{n},i}^- \geq 0 \tag{3.54}$$

If $\lambda_{\hat{n},\mathrm{i}} > 0$, then the *i*th condition requires $v_{\hat{n},i}^+ = -c_{r,i} v_{\hat{n},i}^-$. If the *i*th condition is more than sufficiently satisfied, that is $v_{\hat{n},i}^+ > -c_{r,i} v_{\hat{n},i}^-$ , then that means that other impact impulses $\lambda_{\hat{n},\mathrm{j}}$ , $j \neq i$ have already resulted in a satisfactory outbound velocity $v_{\hat{n},i}^+$ via equation (3.51). It could also mean that the incoming contact velocity $v_{\hat{n},i}^-$ was already separating.

Let's rewrite the complementarity condition in matrix form:

$$0 \leq \boldsymbol{\lambda}_{\hat{\boldsymbol{n}}} \ \perp \ \mathbf{J}_{\hat{n}}\mathbf{u}^+ + C_r \mathbf{J}_{\hat{n}}\mathbf{u}^- \geq 0 \tag{3.55}$$

where $C_r \in \mathbb{R}^{K \times K}$ is a diagonal matrix of restitution coefficients. Finally, using the impact update equation (3.51), we can arrive at the LCP of the form $Ax + b \geq 0 \ \perp \ 0 \leq x$:

$$0 \leq \boldsymbol{\lambda}_{\hat{\boldsymbol{n}}} \perp \mathbf{J}_{\hat{n}}(\mathbf{u}^- + M^{-1}\mathbf{J}_{\hat{n}}^T \boldsymbol{\lambda}_{\hat{\boldsymbol{n}}}) + C_r \mathbf{J}_{\hat{n}}\mathbf{u}^- \geq 0$$

$$\Rightarrow 0 \leq \boldsymbol{\lambda}_{\hat{\boldsymbol{n}}} \perp \mathbf{J}_{\hat{n}} M^{-1}\mathbf{J}_{\hat{n}}^T \boldsymbol{\lambda}_{\hat{\boldsymbol{n}}} + (\mathbb{1} + C_r)\mathbf{J}_{\hat{n}}\mathbf{u}^- \geq 0 \tag{3.56}$$

Where $A = \mathbf{J}_{\hat{n}} M^{-1} \mathbf{J}_{\hat{n}}^T$, $x = \boldsymbol{\lambda}_{\hat{\boldsymbol{n}}}$, and $b = (\mathbb{1} + C_r)\mathbf{J}_{\hat{n}}\mathbf{u}^-$.

There is nothing new here. While we can use the same LCP solvers as we have used previously for resting contact, RF3 uses a OSQP solver here instead for numerical stability. Once we have solved for the impact impulses $\boldsymbol{\lambda}_{\hat{\boldsymbol{n}}}$, we can substitute them into the impact velocity update equation (3.51) to get the velocity after impact. By construction, this model for impact satisfies (SYM), (KIN), and (ONE) [6].

### 3.3.4. Frictional Impact

We now must apply friction to our impact. Recall Couloumb's law which relates the frictional impulse to the normal impulse in resting contact in equation (3.31). We can similarly relate the frictional impact impulse to the normal impact impulse. The key difference is in the instantaneous setting. In instantaneous impact, the direction of tangential dissipation remains fixed. Thus, we do not have to worry about a friction cone or linearizing it. By the maximal dissipation law of friction [5], our frictional impulses for each contact point will always be applied in the opposite direction of the tangential contact velocity pre-impact. Note that we are making an approximation to simplify the friction constraint in the non-smooth setting. In smooth impact mechanics, the frictional impulse is numerically integrated as a function of the normal impulse [8], and direction of the frictional impulse does not remain fixed throughout impact in general.

Let's define the generalized basis of instantaneous friction directions $\boldsymbol{D} \in \mathbb{R}^{6\times k}$ :

$$\mathbf{D} = \begin{bmatrix} \Gamma_1^T \hat{d}_1 & \cdots & \Gamma_k^T \hat{d}_k \end{bmatrix} \tag{3.57}$$

Let's explain this matrix one component at a time. $\Gamma_k = \begin{bmatrix} \mathbb{1} & -r_k^{\times} \end{bmatrix} \in \mathbb{R}^{3\times 6}$ here is the relative velocity Jacobian for the *k*'th contact point, which as its name implies can be used to find the relative velocity at the *k*'th contact point $v_k \in \mathbb{R}^3$ from the generalized velocity:

$$\mathbf{v}_k = \Gamma_k \mathbf{u}^- \tag{3.58}$$

where the − superscript denotes that this is on pre-impact velocities. The tangential velocity $\mathbf{v}_{t,k} \in \mathbb{R}^3$ can be found by projecting the relative velocity $\mathbf{v}_k$ onto the plane of contact with normal $\hat{\mathbf{n}}_k$:

$$\mathbf{v}_{t,k} = \mathbf{v}_k - (\mathbf{v}_k \cdot \hat{\mathbf{n}}_k)\hat{\mathbf{n}}_k \tag{3.59}$$

Finally we get the friction directions as the directions that oppose the normalized tangential velocities $\hat{d}_k = -\mathbf{v}_{t,k} / \|\mathbf{v}_{t,k}\|$, and in the columns of $\mathbf{D}$, we map them back to the generalized velocity space with $\Gamma_k^T$. We can now use $\mathbf{D}$ to complete the impact velocity update with friction impulses:

$$\mathbf{u}^+ = \mathbf{u}^- + M^{-1}\mathbf{J}_{\hat{n}}^T \boldsymbol{\lambda}_{\hat{\boldsymbol{n}}} + M^{-1}\mathbf{D}\boldsymbol{\lambda}_{\hat{\boldsymbol{d}}} \tag{3.60}$$

where the frictional impact impulses for *K* contact points are denoted as $\boldsymbol{\lambda}_{\hat{\boldsymbol{d}}} \in \mathbb{R}^k$.

$$\boldsymbol{\lambda}_{\hat{\boldsymbol{d}}} = \operatorname{argmin}_{\mathbf{z}} \frac{1}{2} \mathbf{z}^T \mathbf{D}^T M^{-1} \mathbf{D}\mathbf{z} + \mathbf{z}^T \mathbf{D}^T (\mathbf{u}^- + M^{-1}\mathbf{J}_{\hat{n}}^T \boldsymbol{\lambda}_{\hat{\boldsymbol{n}}}) \tag{3.61}$$

subject to:

$$0 \le \mathbf{z} \le \mu \boldsymbol{\lambda}_{\hat{\boldsymbol{n}}} \tag{3.62}$$

where $\mu \in \mathbb{R}^{k\times k}$ is the diagonal matrix of friction coefficients.

This minimization problem comes from the principle of maximum dissipation, which in this case means that the frictional impulse maximizes the energy dissipation [5]. To better understand equation (3.61), we can rewrite the minimization term as:

$$\boldsymbol{\lambda}_{\hat{\boldsymbol{d}}}^{*} \cdot \mathbf{v}_t^{+}(\boldsymbol{\lambda}_{\hat{\boldsymbol{d}}}^{*}) \tag{3.63}$$

where $\boldsymbol{\lambda}_{\hat{\boldsymbol{d}}}^{*}$ is the maximally dissipating frictional impulse and is the parameter into $\mathbf{v}_t^{+}(\mathbf{z}) \in \mathbb{R}^k$, which are the post impact tangential velocities. By definition, the elements of $\mathbf{v}_t^{+}$ will be non-positive, and the frictional impulse can only oppose them in the opposite direction. Minimizing the term $\boldsymbol{\lambda}_{\hat{\boldsymbol{d}}}^{*} \cdot \mathbf{v}_t^{+}(\boldsymbol{\lambda}_{\hat{\boldsymbol{d}}}^{*})$ means computing the frictional impulses that maximizes the dissipation. The solution is not trivial because the frictional impulses also affect the elements of $\mathbf{v}_t^{+}$ , and thus need to be solved with a quadratic program.

# 4. Environment and Infrastructure Factors

## 4.1. Materials

There are 2 material properties that will affect the contact dynamics between our rock and the slope. They are the coefficient of friction $\mu$ and coefficient of restitution $c_r$, or $R_n$, as its known in RF3. The materials are only defined on the slope, and only come into effect when there is contact between the rock and the slope. There are multiple methods in RF3 for defining materials on the slope, and these are in order of priority: a triangle to material mapping, material polygon regions, image segmentation material textures, and a default material.

1. Triangle to Material Mapping: each triangle is assigned a material id, which is an index into a material list. Multiple triangles can share the same material id. If there is contact at an edge or vertex, then the minimum material id attached to one of the neighboring faces will be selected.
2. Material Polygon Regions: This method defines material geometry uncoupled from mesh geometry through polygon regions defined in the horizontal X-Y plane. The material at a contact point is found by projecting the 3D coordinate of the point to the X-Y plane and then computing an inside-polygon intersection to find the polygon region the point is in. Each material polygon also has a priority id, so there is a deterministic selection process for overlapping material polygon regions. Once the polygon region is determined, it is easy to find the material id attached to the region.
3. Material textures: This method defines a texture of material IDs. The texture is mapped to the slope in a top-down projection. A slope contact point is easily converted to a texture coordinate to get the corresponding material ID. The generation of the material texture itself is done in the modeler side and uses an image segmentation feature.
4. Default material: If none of the other methods found a material ID, then the user specified default material is used.

## 4.2. Barriers

Barriers in RF3 are defined as triangle meshes, but they do not count as rock-slope contacts. Contact detection with barriers happens when the center of mass of the rock intersects with the barrier geometry. Instead of computing impact or resting contact impulses, an attenuation function is applied to the rock's translational kinetic energy, effectively reducing it's linear velocity as it passes through the barrier.

# 5. Simulation Design for Numerical Stability

## 5.1. Unconstrained rotational energy

A known instability in rigid body simulation is in the time integration of inertial forces for rigid bodies, though this fact is not very well documented. Much like all stability problems in physical simulation, they can be resolved by adding damping forces or by taking smaller timesteps.

### 5.1.1. Angular Drag

We apply a drag torque on the rock's center of mass:

$$\tau_d = c_d \boldsymbol{\omega}^- \tag{5.1}$$

where $\tau_d$ is the drag torque, $c_d$ is the drag coefficient, and $\boldsymbol{\omega}^-$ is the angular velocity before drag and inertial forces are applied. From our tests, we find that an angular drag coefficient of 0.1 leads to numerical stability for reasonable angular velocities.

### 5.1.2. Adaptive Timestepping

When rock's gain large angular velocity, angular drag is no longer enough to ensure simulation stability. Instead we adaptively shrink our simulation timestep each time we detect an increase in rotational kinetic energy during unconstrained motion. When our rotational kinetic energy is non-increasing, we can then slowly increase our simulation timestep each timestep until it reaches the initial max simulation timestep.

## 5.2. Stopping Conditions

Our rigid body engine must be able to determine when rock's have stopped, so they can send the results back to the RF3 user application. We have four separate conditions for stopping our rock path.

### 5.2.1. Position Stopping Condition

The first condition to stop our rock path is if our rock leaves the axis-aligned bounding box of the slope mesh through the bottom and side faces. If this stopping condition is met, then the user will need to extend their slope model to see the full extent of the rock path.

### 5.2.2. Velocity Stopping Condition

The second condition to stop our rock is if our rock's velocity is close enough to zero (measured by a numerical epsilon), and the forces acting on the rock are in equilibrium (again measured by a numerical epsilon).

### 5.2.3. Jitter-Tolerant Stopping Condition

The velocity stopping condition by itself is not enough due to numerical jitters which may keep a rock's velocity or external forces oscillating above the epsilon threshold. We use the method described in Chapter 23 "A Jitter-Tolerant Rigid Body Sleep Condition" described in Game Engine Gems 2 [4]. This method works by growing an axis-aligned bounding box at three distinct points on the rock. One box at the center of mass, one box on a point along the rock's x-axis, and one box on a point along the rock's y-

axis. As the rock moves, these boxes grow to encapsulate these three points' trajectories. After a certain amount of time has elapsed, we check if any of the bounding boxes lengths are smaller than a numerical epsilon. If they are, we can confidently put our rock to sleep. If not, we reset the boxes and continue this process of detecting jitters.

### 5.2.4. Maximum Timesteps

Finally, we make sure our rock path ends by putting a cap on the maximum number of timesteps. If none of the other stopping conditions are met, we can at least guarantee our rock path computation will inevitably end.

# References


[1] 3blue1brown. Visualizing quaternions (4d numbers) with stereographic projection, 2018.

[2] Andrews, S., Erleben, K., and Ferguson, Z. “Contact and friction simulation for computer graphics.” In *ACM SIGGRAPH 2022 Courses* (New York, NY, USA, 2022), SIGGRAPH ’22, Association for Computing Machinery.

[3] Baraff, D., and Witkin, A. P. An introduction to physically based modeling: Rigid body simulation I—unconstrained rigid body dynamics.

[4] Lengyel, E. *Game Engine Gems 2*, 1st ed. A. K. Peters, Ltd., USA, 2011.

[5] Preclik, T., Eibl, S., Rüde, U. “The maximum dissipation principle in rigid-body dynamics with inelastic impacts.” *Computational Mechanics*, 62, 81-96.

[6] Smith, B., Kaufman, D. M., Vouga, E., Tamstorf, R., and Grinspun, E. “Reflections on simultaneous impact.” *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2012)*. 31, 4 (2012), 106:1–106:12.

[7] Stewart, D. E. “Rigid-body dynamics with friction and impact.” *SIAM Review*. *42*, 1 (2000), 3–39.

[8] Stronge, W. J. *Impact Mechanics*, 2 ed. Cambridge University Press, 2018.