Physical System Description

We want to simulate a 3D system where a rigid body cube (dice) is dropped from a certain height with some initial linear and angular velocity. We assume no air resistance and that the dice has mass m and moment of inertia $\mathbf{I}_{\text{body}}$. The side length of the cube is s. The dice will experience gravitational force and will collide with the ground, which we will model as a perfectly inelastic collision to prevent endless bouncing. The dice is also uniformly dense, so each particle of the dice has the same mass and the center of mass is at the geometric center. The world frame is defined such that the ground plane is at $y = 0$ and gravity acts in the negative y direction.

Formulation

Unconstrained Motion

The Least Action Principle states that the system will follow the path that minimizes the action. The action S is the integral of the Lagrangian $L = K - U$ over time $[0, T]$. Let $\mathbf{q} = (\mathbf{c}, \mathbf{R}) \in \mathbb{R}^3 \times \mathrm{SO}(3) = Q$ be our manifold of configurations, where $\mathbf{c}$ is the center of mass of the dice and $\mathbf{R}$ is the rotation matrix representing the orientation of the dice.

The orientation evolves according to the kinematic relation $\dot{\mathbf{R}} = \mathbf{R}[\Omega]_\times$ where $\Omega \in \mathbb{R}^3$ is the angular velocity in **body coordinate** frame. We can express the kinetic energy K and potential energy U in terms of $\mathbf{q}$ and $\dot{\mathbf{q}}$.

$$K(\mathbf{q}, \dot{\mathbf{q}}) = \frac{m}{2}|\dot{\mathbf{c}}|^2 + \frac{1}{2}\Omega^\top \mathbf{I}_{\text{body}}\Omega$$

where $\mathbf{I}_{\text{body}}$ is the moment of inertia tensor in the body frame. Note that $\mathbf{I}_{\text{world}} = \mathbf{R}\mathbf{I}_{\text{body}}\mathbf{R}^\top$ where $\mathbf{R}$ is the rotation matrix from the body frame to the world frame as a function of time. We use the body frame moment of inertia to keep the inertia tensor constant, which simplifies our calculations.

We define the potential energy as

$$U(\mathbf{q}) = mg(\mathbf{e}_y^\top \mathbf{c})$$

where $\mathbf{e}_y$ is the unit vector in the y direction. This isolates the $y-$coordinate of the center of mass, which is the only coordinate that contributes to the potential energy. Thus, our Lagrangian is

$$L(\mathbf{q}, \dot{\mathbf{q}}) = \frac{m}{2}|\dot{\mathbf{c}}|^2 + \frac{1}{2}\Omega^\top \mathbf{I}_{\text{body}}\Omega - mg(\mathbf{e}_y^\top \mathbf{c})$$

The unconstrained action is then

$$S(\mathbf{q}) = \int_0^T \left(\frac{m}{2}|\dot{\mathbf{c}}|^2 + \frac{1}{2}\Omega^\top \mathbf{I}_{\text{body}}\Omega - mg(\mathbf{e}_y^\top \mathbf{c})\right) dt$$

Constrained Motion & KKT Optimality Conditions

The dice will collide with the ground plane at $y = 0$. We can model this collision as a constraint on the configuration space such that the y coordinate of the center of mass cannot go below zero. Since it would be computationally expensive to solve with this constraint for every atom, we recognize that only the surface of the dice can collide with the ground. Furthermore, since the dice is a convex polyhedron, it is suffcient to only consider the corners of the dice for collision detection.

Thus, we can model the collision constraint as a set of 8 inequality constraints, one for each corner of the dice. Let $\mathbf{v}_k$ be the position of the k-th corner of the dice in body coordinates. For a dice of side length s, the corners $\mathbf{v}_k$ are fixed in the body frame and can be expressed as follows:

$$\mathbf{v}_k = \frac{s}{2}\begin{bmatrix}\pm 1\\ \pm 1\\ \pm 1\end{bmatrix}$$

We define the function $f_{\mathbf{a}} : Q \to \mathbb{R}^3$ to express the vertices in world coordinates for any atom $\mathbf{a} \in \mathcal{A}$.

$$f_{\mathbf{v}_k}(\mathbf{q}) = \mathbf{c} + \mathbf{R}\mathbf{v}_k$$

No atom can penetrate the ground plane, so the y coordinate of each corner must be non-negative. To satisfy the KKT optimality condition form, we need to express this as an inequality constraint $h_{\mathbf{a}} \leq 0$. This is equivalent to saying that the negative of the y coordinate of each corner must be non-positive, i.e.

$$h_{\mathbf{v}_k}(\mathbf{d}) = -\mathbf{e}_y^\top(\mathbf{d}) \leq 0$$

By composition,

$$(h_{\mathbf{v}_k} \circ f_{\mathbf{v}_k})(\mathbf{q}) = -\mathbf{e}_y^\top(\mathbf{c} + \mathbf{R}\mathbf{v}_k) \leq 0$$

This provides us with the following interpretation of the constraint:

- If $h_{\mathbf{v}_k} < 0$ then the corner is above the ground plane and there is no contact force.
- If $h_{\mathbf{v}_k} = 0$ then the corner is in contact with the ground plane and there is a contact force.
- If $h_{\mathbf{v}_k} > 0$ then the corner is below the ground plane and there is a penetration, which violates our constraint.

Thus, our final KKT optimality condition is:

$$\begin{cases} \dfrac{d}{dt}\left(\dfrac{\partial K}{\partial \dot{\mathbf{q}}}\right) = \dfrac{\partial K}{\partial \mathbf{q}} - \dfrac{\partial U}{\partial \mathbf{q}} - \displaystyle\sum_{k=1}^{8}\left(\dfrac{\partial(h_{\mathbf{v}_k} \circ f_{\mathbf{v}_k})}{\partial \mathbf{q}}\right)^\top \lambda_k^\perp(t) \\ \quad h_{\mathbf{v}_k}(\mathbf{q}) \leq 0 \quad \forall k \in \{1, \ldots, 8\} \end{cases}$$

for some Lagrange multipliers $\lambda_k^\perp(t)$ indexed by both k and t so that $\lambda_k^\perp(t) \geq 0$ and $\lambda_k^\perp(t) = 0$ whenever $(h_{\mathbf{v}_k} \circ f_{\mathbf{v}_k})(\mathbf{q}) < 0$. The $\perp$ indicates that this is a normal force exerted by the ground. (Note: We use $\lambda_k^\perp$ rather than μ_k to avoid notation collision with the Coulomb friction coefficient μ).

Dissipative Forces

To ensure the dice eventually comes to rest, we model energy dissipation using Coulomb dry friction that acts strictly at the contact points.

As established before, $\lambda_k^\perp \geq 0$ is the normal impulse exerted by the ground on corner k. By Coulomb's law of friction, the magnitude of the tangential frictional impulse $\lambda_k^\parallel$ is bounded by the friction cone: $\|\lambda_k^\parallel\| \leq \mu\lambda_k^\perp$, where μ is the coefficient of friction. When the corner is sliding, the friction operates exactly at this boundary to maximize energy dissipation.

To model how much energy is dissipated at each time step, we calculate the sliding distance. Let $\Delta\mathbf{d}_k$ be the tangential displacement (the sliding distance strictly along the x, z plane) of corner k during the time step. The energy dissipated by friction at this specific corner is the frictional force multiplied by the sliding distance:

$$\Delta E_k = \mu\lambda_k^\perp\|\Delta\mathbf{d}_k\|$$

We will be able to calculate $\lambda_k^\perp$

Simulation Details

We will discretize the implicit variational integrator using the backward Euler method. At each time step, we will check for collisions and apply the appropriate contact forces if any of the corners of the dice are in contact with the ground plane. We will model Coulomb dry friction by incorporating the sliding energy dissipation into our objective function.

We will map our optimization variables to the our geometry. The state is parameterized by the center of mass $\mathbf{c} \in \mathbb{R}^3$ and the body coordinate rotation vector $\Delta\theta \in \mathbb{R}^3$. Since the equation for its rotation with respect to time is $\dot{\mathbf{R}} = \mathbf{R}[\Omega]_\times$ where Ω is body coordinate angular velocity, we can determine the next rotational state by using the exponential map

$$\mathbf{R}(t + \Delta t) = \mathbf{R}(t) \exp\left([\Omega \Delta t]_\times\right)$$

Since $\Omega\Delta t$ is the total rotational displacement during the time step, it is actually $\Delta\theta$. Thus, the new orientation of the dice is given by

$$\mathbf{R}^{(n+1)} = \mathbf{R}^{(n)} \exp\left([\Delta\theta]_\times\right)$$

The position of corner k at time n is:

$$f_{\mathbf{v}_k}(\mathbf{c}, \Delta\theta) = \mathbf{c} + \mathbf{R}^{(n)} \exp\left([\Delta\theta]_\times\right) \mathbf{v}_k$$

To find the tangential displacement $\Delta\mathbf{d}_k$, at each time step, we need to isolate the x, z components of the corner's velocity. Let $\mathbf{P}$ be the projection matrix that projects a 3D vector onto the x, z plane.

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Thus,

$$\Delta\mathbf{d}_k(\mathbf{q}) = \mathbf{P}\left(f_{\mathbf{v}_k}(\mathbf{c}, \Delta\theta) - f_{\mathbf{v}_k}(\mathbf{c}^{(n)}, \mathbf{0})\right)$$

And as $f_{\mathbf{a}}(\mathbf{q}) : Q \to \mathbb{R}^3$, $\Delta\mathbf{d}_{t,k}(\mathbf{q}) = [\alpha, \beta]^\top$ where α and β are the x and z components of the corner's tangential displacement, respectively.

At the beginning of each time step, we calculate the **unconstrained predicted state** based purely on inertia and gravity:

$$\mathbf{c}_p = \mathbf{c}^{(n)} + \Delta t \dot{\mathbf{c}}^{(n)}$$
$$\Delta\theta_p = \Delta t \Omega^{(n)}$$

Lastly, before we express the discretized optimization problem, we need to describe the body frame moment of inertia $\mathbf{I}_{\text{body}}$. The moment of inertia tensor for a cube of side length s and mass m is given by

$$\mathbf{I}_{\text{body}} = \frac{ms^2}{6} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \frac{ms^2}{6} \mathbf{I}_{3\times3}$$

We see that the rotational kinetic energy term can be simplified to

$$\begin{aligned} K_{\text{rot}}^{(n+1)}(\mathbf{R}^{(n)}, \dot{\mathbf{R}}^{(n)}) &= \frac{1}{2}\Omega^\top \mathbf{I}_{\text{body}}\Omega \\ &= \frac{1}{2}\Omega^\top \left(\frac{ms^2}{6}\mathbf{I}_{3\times 3}\right)\Omega \\ &= \frac{ms^2}{12}\|\Omega\|^2 \\ &\approx \frac{ms^2}{12\Delta t^2}\|\Delta\theta - \Delta\theta_p\|^2 \end{aligned}$$

The system is solved at each time step by finding the next configuration state $\mathbf{q}^{(n+1)}$ that satisfies the following KKT optimality conditions:

$$(\mathbf{c}^{(n+1)}, \Delta\theta^*) = \underset{\mathbf{c},\Delta\theta\in\mathbb{R}^3}{\operatorname{argmin}} \left(\underbrace{\frac{m}{2\Delta t^2}\|\mathbf{c} - \mathbf{c}_p\|^2}_{\substack{\text{translational} \\ \text{kinetic energy}}} + \underbrace{\frac{ms^2}{12\Delta t^2}\|\Delta\theta - \Delta\theta_p\|^2}_{\substack{\text{rotational} \\ \text{kinetic energy}}} + \underbrace{mg(\mathbf{e}_y^\top \mathbf{c})}_{\substack{\text{potential} \\ \text{energy}}} + \underbrace{\sum_{k=1}^{8} \mu\lambda_k^\perp \|\Delta \mathbf{d}_k(\mathbf{c}, \Delta\theta)\|}_{\substack{\text{energy dissipated} \\ \text{by friction}}} \right)$$

subject to:

$$0 \geq -\mathbf{e}_y^\top f_{\mathbf{v}_k}(\mathbf{c}^{(n+1)}, \Delta\theta^*) \qquad \forall k \in \{1, 2, \ldots, 8\}$$

Implementation Details

The simulation was written in Python 3.14 using the following libraries:

- `NumPy`: For efficient numerical computations and array manipulations.
- `SciPy`: For optimization routines and spatial transformations.
- `Polyscope`: For 3D visualization of the dice simulation.
- `Dataclasses`: For cleaner code organization and management of the dice's state.

Polyscope was used to visualize the dice in 3D and observe its motion as it falls and interacts with the ground plane. A "ground plane" was added at $y = 0$ for visual reference with a checkerboard pattern. SciPy provides the `Rotation` class, which was used to handle the rotation of the dice using rotation vectors and the exponential map. The optimization problem at each time step was solved using SciPy's `minimize` function with the SLSQP method, which allows for nonlinear constraints and exposes the multipliers.

NumPy was used to efficiently compute the objective function and its gradients, as well as to manage the state of the dice across time steps. The state of the dice, including its position, orientation, velocity, and angular velocity, was encapsulated in a dataclass for cleaner code organization.

The objective function is implemented as a Python function that takes in the optimization variables and returns the total energy of the system. The constraints are implemented using a similar approach, defining a function that returns the values of the constraint equations. These constraints are then passed to the `minimize` function to ensure that the optimization respects the physical limits of the system.

Optimization is performed in two passes. First, the system predicts the next state of the system based purely on inertia and gravity, without considering any contact forces. This predicted state serves as the initial guess for the second pass, where we solve the full optimization problem that includes the contact forces and energy dissipation. At the end of each time step, we update the state of the dice based on the optimized configuration.

If the dice has come to rest (its velocity is below a certain threshold), we reset the simulation to its initial state to allow for repeated observation of the dice's behavior.

A telemetry panel was implemented to display the state of the dice at each time step, including its position, orientation, velocity, and angular velocity. This allows us to observe how the state evolves over time and verify that the simulation behaves as expected.

Addendum

During the implementation of the simulation, I realized that I really wanted there to be a way to have the cubes bounce off the ground plane. This was not as interesting to me as modeling the inelastic collision.