The Study of Probabilities and Dynamics of Unfair Dice

Liang Haowei, Yang Xiaochen
Instructor: Zhang Zhonghuan
Beijing National Day School
Beijing, China

Innovation Statement

Our team claim that the paper submitted is our own accomplishment under the guidance of the instructor. To our best knowledge, the content in the paper does NOT include any results from other researchers' works, except as specifically noted and acknowledged in the text. If there is any dishonesty, we are willing to bear all relevant responsibilities.

Team members: Liang Haowei; Yang Xiaochen.

Abstract

Unfair dice are dice with nonuniform density distribution and unequal probability for each outcome of a die toss. A modified method based on Newton-Euler dynamics theories and equations are applied to solve for the outcome of a single toss after a dies motion in air and several bouncing with a set of initial variables including orientation, angular and translational speed, direction of rotational axis, height, coefficient of restitution, etc. Mathematical models made by Wolfram Mathematica are programmed to simulate dice toss with given distributions of initial variables. The possibility for each configuration is found by summarizing all final configurations obtained within given range of initial quantities. Different variables influence on the possibility is discovered. Experiments of tossing several unfair dice are conducted to find the possibility of each outcome under certain variables range. The experimental results are compared with numerical simulating results. It is discovered that the possibility of each outcome is largely depends on the density distribution, initial angular speed, coefficient of restitution, while the initial orientation has a subtle influence on probabilities as well.

Key words: **Probability**, **Newton-Euler Dynamics**, **Numerical Simulation**, **Density Distribution**

Contents

Variable Sheet

Notation	Meaning
$\xi\eta\zeta$	Embedded frame
v^0	Orientational axis of rotation
α	Angle rotated around the rotational axis
$\dot{\alpha}$	The angular velocity around the rotational axis
xyz	The fixed frame
$x'y'z'$	The fixed frame with the origin at the mass center
(v_x, v_y, v_z)	The linear velocity's components
ρ	The density distribution function
χ	The coefficient of constitution
h	The height from mass center to the ground
(ψ, θ, ϕ)	Classical Euler angles
J	Moment of inertia around the rotational axis
(J_ξ, J_η, J_ζ)	Moment of inertia about three embedded frame axes
$(J_{\xi\eta}, J_{\eta\zeta}, J_{\xi\zeta})$	Inertia products
F	External net force
τ	External net torque
P	Linear momentum
L	Angular momentum
$(\ddot{x}, \ddot{y}, \ddot{z})$	The linear acceleration components
$(\omega_\xi, \omega_\eta, \omega_\zeta)$	The angular velocity components about embedded frame
(v_1, v_2, v_3)	The components of unit vector v^0 about xyz frame
(e_0, e_1, e_2, e_3)	Euler parameters
r	vector before transformation
r'	vector after transformation
R	transformation matrix
$(\dot{\psi}, \dot{\theta}, \dot{\phi})$	time derivative of Euler angles
S_z	the z direction impulse from the ground during the collision
A	The point contact with the ground during the collision
(ξ_A, η_A, ζ_A)	The embedded frame coordinate of A
COM	Center of Math
$P(x), x = 1, 2, 3, 4, 5, 6$	True probability of getting an x
$\hat{P}(x), x = 1, 2, 3, 4, 5, 6$	Measured probability of getting an x

Table 1: Variable Sheet

1 Introduction

Its a commonly known fact that a die has six sides, and each side had a possibility of $\frac{1}{6}$ to appear after a toss. As long as the die is symmetric with respect to its six sides, chances for each side to appear are the same [1]. However, if the die has an nonuniform density distribution, the symmetry will be broken and the possibility for each side will not be the same. A die with nonuniform density distribution is called an unfair die, which has shifted center of mass (COM) away from its geometric center. Besides density distributions, many other factors can also affect a dies randomness.

The mystery of dices randomness intrigued many scientists since its invention thousands years ago. Gerolamo Cardano (1501-1576), one of the key figures in the foundation of probability, gave the first systematic treatment of probability and many lessons to cheat in gambling in his famous work *Books on Game of Chance* [2]. Following development of probability theories, Galileo Galilei (15641642) dedicated to the calculation of the number of all possible cases involving the tossing of three dice in his work *Analysis of Dice Games*. [3] In the field of analyzing rigid body dynamics, the simplified models of coin toss were discovered by Joe Keller [4] and Persi Diaconis [1]. Applying Newton-Euler dynamics equations, they derived the correlation between the final configuration of the unbiased coin with a group of initial quantities including the initial orientations, linear and angular speed, and the height of the coin while tossing. J. Strzalko, J. Grabski, A. Stefanski, and T. Kapitaniak built a model simulating fair dices motion in air [5]. Using this model analyzing many groups of variables, they gave a conclusion that a die tosss outcome is largely depended on its initial state while being released. In 2009, Shi, Zhao, Qu, and Lei investigated possibility for unfair die toss through geometric analysis [6]. They compared the sizes of solid angles above each side of the die; however, the dies bouncing was neglected and there was a large difference between their theoretical results and experimental results.

In this research, a model based on Strzalkos model is built to analyze unfair dice with nonuniform density distribution. Many revisions had been made to add density distribution as an additional initial variable. With inputs of variety of initial variables, the new model can also find correlation between these variables and the probabilities for each outcome. Instead of calculating already set variables like old model, the new one chose values of variables randomly from some given distributions, which is more efficient and closer to reality. Experiments are designed to prove the correctness of the new model. At the end, the influence of density distribution, coefficient of restitution, orientation, and angular velocities on the probabilities is discovered, and the correctness of the model is proved.

In Section 2, some basic concepts and equations are explained. Section 3 then shows how the final outcome of a single toss is derived from a set of initial variables. Section 4 explain the logic behind the modeling. All experiment and simulation results are presented in Section 5. In Section 6, all finding are analyzed and some conclusions are given.

2 Mathematical Representation of Dice

2.1 Background of Unfair Dice Modeling

Dice are frequently used in board games, lotteries, and gambling: scenarios when a random number is required. Dies motion in the air and the outcome of a toss, which is one of six numbers, are seemingly unpredictable. However, details of a dies motion and its outcome can be determined by knowing initial conditions before tossing and applying Newton-Euler kinematics. Each group of initial conditions corresponds to only one outcome. By setting ranges of initial variables, the probabilities for each outcome can be determined [4] [5] [7].

A fair dice is a cube with uniform density distribution. When initial variables of a fair dies motion are chosen from infinitely large ranges, each outcome of the motion has the same probability to appear. In contrast, an unfair dice is defined to be a cube with non-uniformed density distribution in our work. With a non-uniform density distribution, the probability of each outcome of an unfair dice tossing is not the same any more despite of the infinitely large ranges of initial variables. All unfair dice in this study are represented with an input of a density distribution function.

With inputs of die's initial conditions, a mathematical model is built to analyze dies motion in air and its states after bouncing, to determine the outcome of a single tossing, and to summarize the chances of each outcome to appear within given ranges.

Besides discovering the influence of a dies density distribution function on the possibility of each outcome, the theories and model can also be used to discover how initial variables of motion influence the possibility of each outcome.

The following quantities are inputs of the theoretical analysis and mathematical model:

1. The initial orientation in Euler angles (ψ, θ, ϕ).
2. The unit vector along the initial rotational axis v^0.
3. The initial angular velocity around the initial rotational axis $\dot{\alpha}$.
4. The initial linear velocity's components in the xyz frames (v_x, v_y, v_z).
5. The density distribution of the die ρ.
6. Coefficient of restitution χ.
7. The height of the die's mass center to the ground h_0.

During theoretical analysis and mathematical modeling, air resistance and frictional force with the ground are neglected. Air resistance is neglected due to its minimal influence compared to gravity on dies motion in air. The frictional force is exerted on the cube for a minimal time interval during the collision, so the influence of the friction before the bouncing of the cube can also be neglected.

2.2 Reference Frames of Moving Dice

While describing the translational and rotational motion of the rigid body in general, two coordinate systems altogether is necessary [12]. One of the system is called the general coordinate, or fixed coordinate, which will be used to describe the general position of the rigid body in the space. Latin letters x-y-z are used to denote general coordinate's axes in this study.

The other coordinate, embedded frame coordinate, is embedded inside the rigid body. In our model, three axes of embedded frame is denoted by Greek letters, ξ, η, and ζ, which correspond the x, y, and z in general coordinate respectively. The position of the object relative to its embedded frame will not change during the motion, so the relatively position of embedded frame to the general frame will keep changing when the object is moving in the general frame. Below is a three dimensional cube with its embedded frame initially coincide with the general frame.

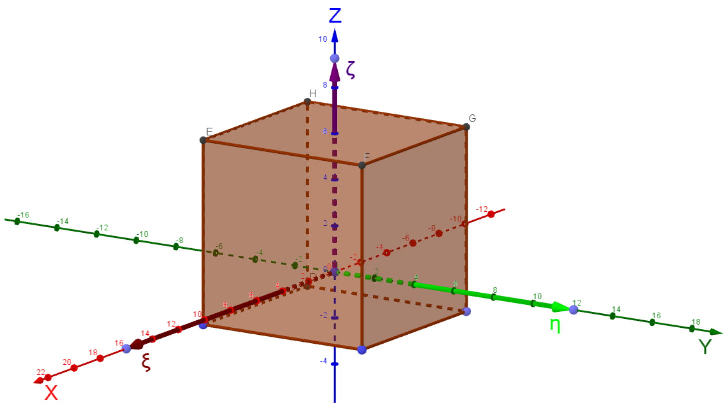

Figure 1: Embedded Frame

When the cube changes its orientation, its relative position to the embedded frame does not change.

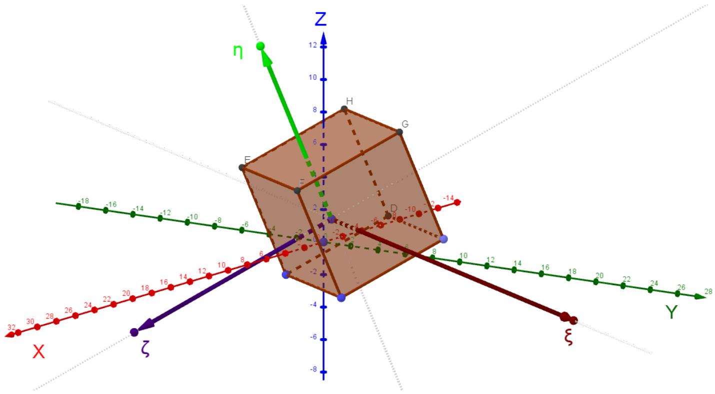

Figure 2: Embedded Frame after Rotation

Because there is no external torque exerted on the die during the motion in the air (the air resistance is neglected), the die rotates around a unit vector whose direction goes through the mass center of the die. To simplify the analysis of die's motion, the origin of the embedded frame will always be set at the mass center of the die. In the contrast, the origin of the general frame can be set at a random place. During this research, the relative position between general frame and embedded frame is used to show the orientation of the cube. To represent the relative position between two coordinates, a concept named Euler angles is introduced.

2.3 Euler Angles and Orientation of Dice

As the object rotates, its embedded frame's relative position to the general xyz frame is shifted, and the object's orientation is described by the angle difference between two coordinate systems. To show the angle difference, another fixed frame $x'y'z'$ with the origin also at the mass center will be shown together with the embedded frame $\xi\eta\zeta$. The following graph shows the position of $x'y'z'$ and of embedded frame:

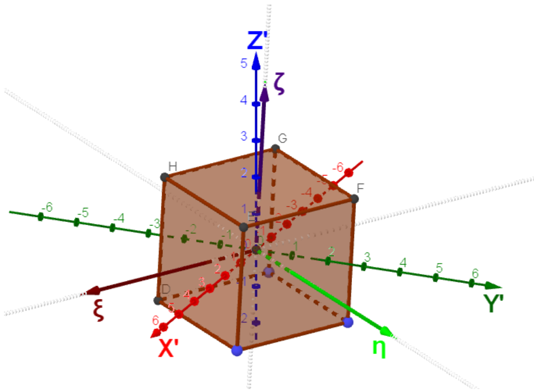

Figure 3: The Embedded Frame and xyz Frame

Those angle difference, called Euler angles, are a very common quantity used while analyzing the rigid body motion [14]. The definition of Euler angles varies from work to work, but the purpose is the same: to represent a rigid body's orientation. In this study's theoretical analysis and model, the classical Euler angles defined in James Diebel's previous work are applied [14]. Under this definition, Euler angles represent angles revolved around an embedded frame's axis. To represent the orientation of the object, the embedded frame starts from the position coincide with the fixed frame. The x' axis coincide with ξ axis; the y' axis coincide with η axis; the z' axis coincide with ζ axis. The object first rotates around z' axis (which is also the ζ axis at the initial position) for ψ degree; then, around the new position of ξ axis after the first rotation, the object rotates for θ degree; at last, the object rotates for ϕ degree around the new position of ζ axis after two former rotations. The rotated angles, ψ, θ, and ϕ are the classical Euler angles for the object, and three of them is enough to represent any orientations of a rigid body.

1. This is the initial orientation of the cube.

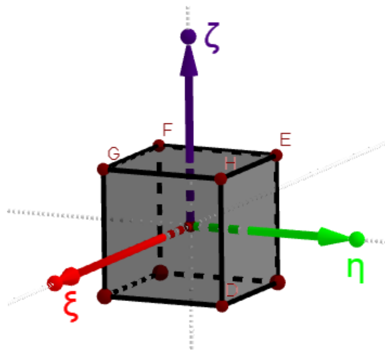

Figure 4: Before Rotation

2. Rotate around ζ axis for ψ degrees.

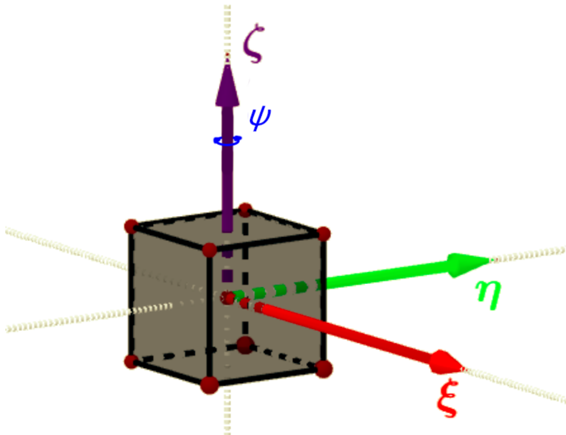

Figure 5: The First Rotation

3. Rotate around ξ axis for θ degrees.

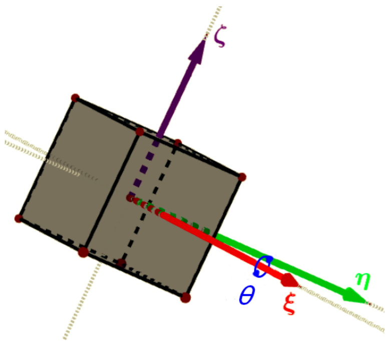

Figure 6: The Second Rotation

4. Rotate around ζ axis for ϕ degrees.

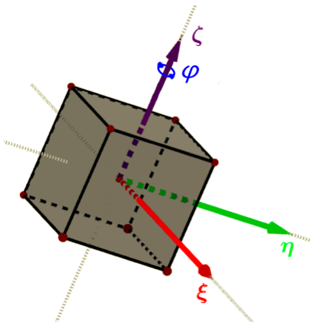

Figure 7: The Last Rotation

It is important to understand that Euler angles is only a representation of orientation or the change of the orientation. The three rotations shown in the graphs above are not the representation of how the cube rotates after the toss. The cube does not rotate around one axis then another when it moves in the air; instead, the cube will only rotate around one axis. Three rotations are just a convenient representation of the cube's orientation [14].

In classical Euler angles, three successive rotations must happen around axes in an order of ζ-ξ-ζ. It is important to note that it is not mandatory to match angles with axis in such a strict way, and any orders of rotations can be calculated, but we choose to have this classical order because it is used by most of the previous work on rigid body dynamics, and many of the formulas from the former works can only apply classic Euler angles [5] [4] [7].

In this research, initial Euler angles are given as initial quantities.

2.4 Density Distribution Function of Dice

The density distribution function of the cube is another initial condition to analyze the dynamics of the cube. The coordinate of the mass center of a cube can be derived by knowing its density distribution, and the moment of inertia and inertia product can all be calculated from the density distribution function.

In the experiment section of this study, all the specimens tested are composed of two different metal materials. To imitate the situation of the experiment, all functions for the density distribution in this study are discrete, constant functions with two values (densities of two metals). A plane divide the two domains of variables corresponding to the two values of densities; in the experiment, this plane will be the surface that two metal blocks glued together. The density function is required while calculating the mass center or moment of inertia. As the cube move in the fixed xyz frame, the density function will keep changing while the cube rotates in the general coordinate systems. To avoid the complexity, all density distribution functions are written using coordinates from embedded frame, so functions are unchanged throughout cubes' motion. The sides of cube will be parallel with the $\xi\eta$ plane and $\xi\zeta$ plane. The following graph shows a cube made out of same volume of two different materials. The graph is set in a $\xi\eta\zeta$ frame, and the plane dividing the two domains is the $\xi\eta$ plane:

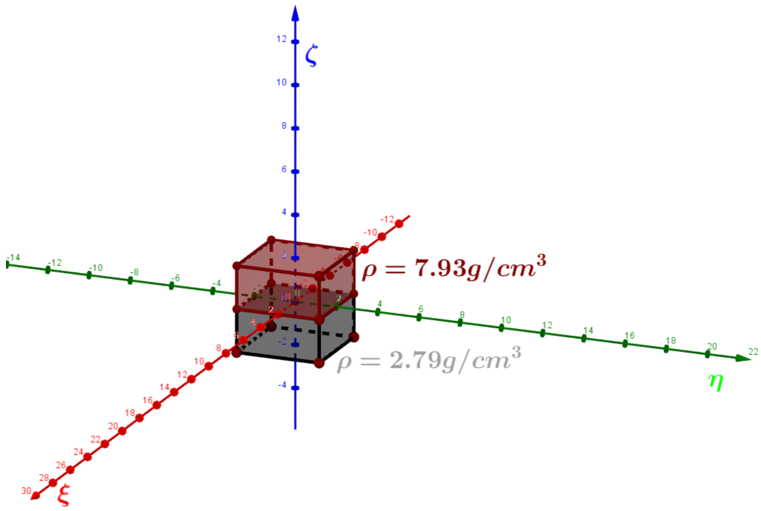

Figure 8: Density Distribution Example

$$\rho(\xi,\eta,\zeta) = 7.93(-2 \leqslant \xi \leqslant 2, -2 \leqslant \eta \leqslant 2, \zeta > 0) \tag{1}$$

$$\rho(\xi,\eta,\zeta) = 2.79(-2 \leqslant \xi \leqslant 2, -2 \leqslant \eta \leqslant 2, \zeta < 0) \tag{2}$$

With the equation for density distribution, the three coordinates of mass center of the cube in the embedded frame can be written as:

$$\xi_{cm} = \frac{\iiint_V \xi\rho(\xi,\eta,\zeta)d\xi d\eta d\zeta}{\iiint_V \rho d\xi d\eta d\zeta} \tag{3}$$

$$\eta_{cm} = \frac{\iiint_V \eta\rho(\xi,\eta,\zeta)d\xi d\eta d\zeta}{\iiint_V \rho d\xi d\eta d\zeta} \tag{4}$$

$$\zeta_{cm} = \frac{\iiint_V \zeta\rho(\xi,\eta,\zeta)d\xi d\eta d\zeta}{\iiint_V \rho d\xi d\eta d\zeta} \tag{5}$$

The moment of inertia around the rotational axis can be written as:

$$J = \iiint_V x^2 d\xi d\eta d\zeta \tag{6}$$

In this equation, x is the distance between a point inside the cube and the rotational axis. To find the value of x, the formula for the perpendicular distance from a point to a line in 3-dimension is used. In this equation, two points, v_1 and v_2, along the rotational axis is needed, and the point inside the cube is denoted by C. The expression of x is shown as the following formula.

$$x = \frac{\overrightarrow{Cv_1} \times \overrightarrow{Cv_2}}{\overrightarrow{v_1 v_2}} \tag{7}$$

The unit vector for the direction of rotational axis, v^0 will be given as one of the initial condition.

2.5 Basic Kinematic Quantities in Dice Motion

Dice's motion follows Newton-Euler equations, which are the equations describing the general motion of rigid body. Newton equations describe dice's translational motion. Euler's equations describe dice's rotational motion. Combing Newton's and Euler's equations, the kinematics of a die toss can be analyzed in both perspectives of translational motion and rotational motion. The most basic Newton-Euler equations are:

$$\frac{dP}{dt} = F \tag{8}$$

$$\frac{dL}{dt} = \tau \tag{9}$$

In Equations (8 and 9), P is the linear moment and L is the angular momentum. In this research, all air resistance is neglected. When the die is moving in the air, there is no other force except gravity, and the angular momentum is always constant. F equals to the gravitational force mg and τ equals to zero.

As vectors, force has three components along x, y and z axes in general frame, and torque has three components along ξ, η, and ζ axes in embedded frame. Knowing the magnitude of each components of force, the linear acceleration can be calculated. With known torque, components of angular acceleration and angular velocity can be determined. Express force and torque with linear and angular accelerations and angular velocities, the Newton-Euler equations are written as [13]:

$$m\ddot{x} = f_x \tag{10}$$

$$m\ddot{y} = f_y \tag{11}$$

$$m\ddot{z} = f_z \tag{12}$$

$$J_\xi\dot{\omega}_\xi + (J_\zeta - J_\eta)\omega_\eta\omega_\zeta - J_{\xi\zeta}\dot{\omega}_\zeta - J_{\xi\eta}\dot{\omega}_\eta + J_{\eta\zeta}(\omega_\zeta^2 - \omega_\eta^2) + (J_{\xi\eta}\omega_\zeta - J_{\xi\zeta}\omega_\eta)\omega_\xi = \tau_\xi \tag{13}$$

$$J_\eta\dot{\omega}_\eta + (J_\xi - J_\zeta)\omega_\xi\omega_\zeta - J_{\xi\eta}\dot{\omega}_\xi - J_{\eta\zeta}\dot{\omega}_\zeta + J_{\xi\zeta}(\omega_\xi^2 - \omega_\zeta^2) + (J_{\eta\zeta}\omega_\xi - J_{\xi\eta}\omega_\zeta)\omega_\eta = \tau_\eta \tag{14}$$

$$J_\zeta\dot{\omega}_\zeta + (J_\eta - J_\xi)\omega_\xi\omega_\eta - J_{\eta\zeta}\dot{\omega}_\eta - J_{\xi\zeta}\dot{\omega}_\xi + J_{\xi\eta}(\omega_\eta^2 - \omega_\xi^2) + (J_{\xi\zeta}\omega_\eta - J_{\eta\zeta}\omega_\xi)\omega_\zeta = \tau_\zeta \tag{15}$$

In Equations (13-15), J is the moment of inertia, J_ξ, J_η, J_ζ are the moment of inertia of the cube about the three embedded frame axes. The formulas for the moment of inertia about ξ axis, η axis, and ζ axis are [9]:

$$J_\xi = \int_B (\eta^2 + \zeta^2)\ dm \tag{16}$$

$$J_\eta = \int_B (\xi^2 + \zeta^2)\ dm \tag{17}$$

$$J_\zeta = \int_B (\xi^2 + \eta^2)\ dm \tag{18}$$

Those moments of inertia with two axis symbols are called inertia products, which are quantities used to measure the symmetry of the cube. When the cube is perfectly symmetric, the inertia products are zero [9]. However, in our research, unfair dice are considered. When a die is unfair, the geometric center of it is different from the its mass center, so the die is unsymmetrical, and inertia products are nonzero. The inertia product is defined to be [9]:

$$J_{xy} = \int_B xy\ dm \tag{19}$$

When there is no external torque exerted on, Euler's equations can be simplified as [13]:

$$J_\xi\dot{\omega}_\xi - (J_\eta J_\zeta)\omega_\eta\omega_\zeta = 0 \tag{20}$$

$$J_\eta\dot{\omega}_\eta - (J_\zeta J_\xi)\omega_\zeta\omega_\xi = 0 \tag{21}$$

$$J_\zeta \dot{\omega}_\zeta - (J_\xi J_\eta)\omega_\xi \omega_\eta = 0 \tag{22}$$

Based on equations listed in this section, all equations about the cube's dynamics in air can be derived. Because the focus of this study is more about how variety of formulas correlate and connect with each other to stimulate the toss of a die, the detailed mathematical proofs of equations presented later in the dynamics analysis part will not be presented in this paper. By using fundamental equations listed in this sections, all equations used in following sections can be proved mathematically.

Another important fundamental formula before starting the dynamics analysis is the correlation between the angular velocity around the orientational axis and the angular velocity components along three axes of coordinates (embedded coordinate and general coordinate). During a die's motion in air, its mass center follows a trajectory without any rotation, so a die's rotation must be around the die's mass center. This rotational axis is called orientational axis (whose direction is denoted by v^0). Because the cube revolves around this axis, the axis's direction remains unchanged in both general frame and embedded frame; however, its directions in two frames are not necessarily to be same.

$$\dot{\alpha} = \sqrt{\omega_\xi^2 + \omega_\eta^2 + \omega_\zeta^2} \tag{23}$$

$$\dot{\alpha} = \sqrt{\omega_x^2 + \omega_y^2 + \omega_z^2} \tag{24}$$

v^0 in the general frame is given as an already known value during the calculations. Because two frames are different, $\omega_\xi, \omega_\eta, and \omega_\zeta$can not to directly derived. Other formulas in later sections will be used to achieve this goal.

3 Dynamics of Dice

3.1 Dice in the Air

3.1.1 Determine the Vertical Displacement and Time Duration

Because air resistance is ignored, the die undergoes a simple free-fall following Newton's laws of Kinematics. The vertical displacement and time duration are the two important values that could be determined by translastional motion analysis.

Through analyzing the rotational aspect of the motion, the time function of embedded frame angular velocities and orientation can be derived.

Linear Motion

The direction of gravitational force on a die is in the $-z$ direction, and the gravity does not affect the x component and y component of a die's velocity.

If the time taken (t) of the free fall motion is given, the z component of the die's velocity (z')is:

$$\dot{z}' = \dot{z} + gt, \tag{25}$$

where g is the acceleration due to gravity. The prime ($'$) represent "after." The notation is also used in all sections of this paper.

Given the displacement (h) of the die's free fall motion, the z component of the die's velocity can also be calculated by

$$\dot{z}' = -\sqrt{\dot{z}^2 + 2gh}. \tag{26}$$

Rotational Motion

The gravity, by definition, is acting on the die's center of gravity. On Earth's surface, for an object such as die, its mass center and center of gravity can be considered the same. Because the gravity is acting on the center of mass, there is no torque exerting on the die; therefore, the die's angular velocity remains constant before it touches the ground. Also, the angular speed (or the magnitude of the angular velocity) and the axis it rotates are constant.

Given the time of the free fall motion, the angle (α) it rotates during free fall can be calculated by

$$\alpha = ||\dot{\alpha}||t \tag{27}$$

Determine the Displacement and Time

The displacement of a die from the start of motion to its collision with the ground is not the distance between the initial position and the surface of the ground. When the die hits the ground, its center of mass is above the surface of the ground a little bit.

The true displacement (h_{true}) of the die is the solution to this equation

$$h_{true} + h_{com} = h_{tot}, \tag{28}$$

where h_{tot} and h_{com} are the heights of the center of mass from the ground when the die is in the initial position and when the die hits the ground, respectively.

To calculate the true displacement h_{true}, an algorithm is used to approximate the true distance. Even though it's an approximation, the value calculated out through this method will be accurate enough to find other values required in later steps.

At first, the value of h_{true} is set to be equal to h_{tot}. Then, the expected h_{com} from the expected orientation is calculated. The expected orientation is the orientation the die reaches when it falls a distance of h_{true}. The value of h_{true} is replaced by $h_{tot} - h_{com}$, a value closer to the real value. Since the h_{com} changes with different orientations, the new value of h_{true} will obtain a different h_{com}. A new h_{true} is calculated, and it is closer but not the exact value. Finally, this recursive process will soon determine a very close value of h_{true}.

The true time taken during the free fall motion can be calculated from the true displacement h_{true}, and the velocity of the die after the motion can also be calculated.

3.1.2 Determine the Angular Velocity as Function of Time

With the known orientational axis of rotation (v^0) and the magnitude of angular velocity ($\dot{\alpha}$), the angular velocity components around three embedded rotational axes can be solved using Euler parameters.

Euler parameters is used by many previous works about rigid body movement to represent the orientation of the object [15]. With the inputs of a unit vector and the angular velocity around it, Euler parameters can solve for the change in orientation, the angular speed components in embedded frame and general frame. Because both Euler parameters and Euler angles can be used to represent orientation, they transform into each other using a set of formulas which is introdued in next section (3.1.2).

Altmann's definition about Euler Parameters is presented here [15]:

$$p = \begin{bmatrix} e0 \\ e1 \\ e2 \\ e3 \end{bmatrix} = \begin{matrix} \cos\frac{\alpha}{2} \\ v_1 \sin\frac{\alpha}{2} \\ v_2 \sin\frac{\alpha}{2} \\ v_3 \sin\frac{\alpha}{2} \end{matrix} \tag{29}$$

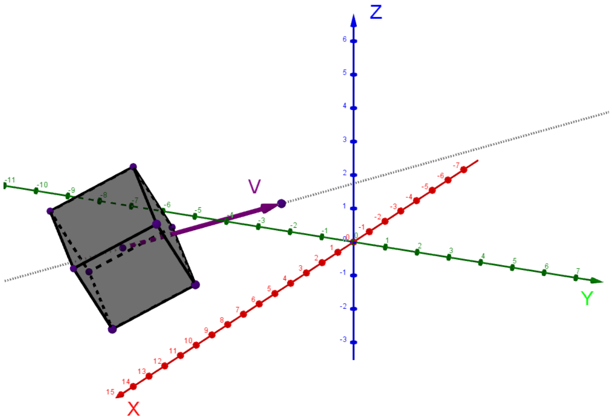

Figure 9: Object After Revolving a Single Axis

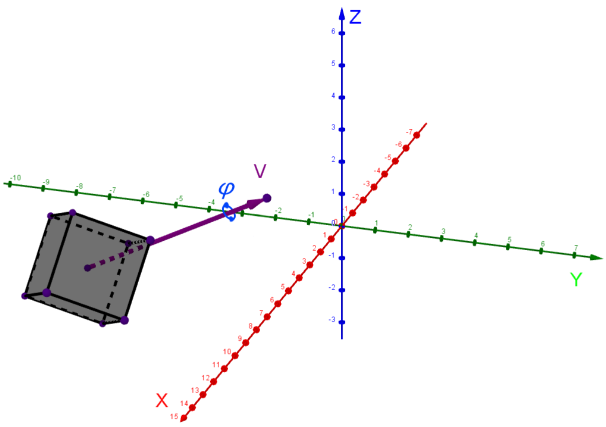

Figure 10: Object After Revolving a Single Axis

In the matrix, v_1, v_2, and v_3 are the three components of v^0, the unit vector in the direction of the orientational rotational axis. α is the total degree rotated during to rotational motion. The direction of v^0 remains unchanged relative to both frames. v_1, v_2, and v_3 are the components in the vector about x, y, and z axes respectively. A graph about v^0 and α is shown below:

In our model, unit vector v^0 and the angular speed around the central axis $\dot{\alpha}$ are given as the initial inputs. The exact correlations between Euler parameter and ω_ξ, ω_η, and ω_ζ are given below. From Strzalko's previous work, the equation looks like this [8]:

$$\omega = \begin{bmatrix} \omega_\xi \\ \omega_\eta \\ \omega_\zeta \end{bmatrix} = 2 \begin{bmatrix} \dot{e_1}e_0 - \dot{e_0}e_1 - \dot{e_3}e_2 + \dot{e_2}e_3 \\ \dot{e_2}e_0 + \dot{e_3}e_1 - \dot{e_0}e_2 - \dot{e_1}e_3 \\ \dot{e_3}e_0 - \dot{e_2}e_1 + \dot{e_1}e_2 - \dot{e_0}e_3 \end{bmatrix} \tag{30}$$

$$\dot{e_0} = -\frac{1}{2}\sin\frac{\alpha}{2}\cdot\dot{\alpha} \tag{31}$$

$$\dot{e_1} = v_1\frac{1}{2}\sin\frac{\alpha}{2}\cdot\dot{\alpha} \tag{32}$$

$$\dot{e_2} = v_2\frac{1}{2}\sin\frac{\alpha}{2}\cdot\dot{\alpha} \tag{33}$$

$$\dot{e_3} = v_3\frac{1}{2}\sin\frac{\alpha}{2}\cdot\dot{\alpha} \tag{34}$$

From equations above, time functions of three embedded frame angular velocity components can be derived. To find the three angular velocity components at particular moment, the degree of rotation from that beginning to that moment of is required. Given the magnitude of angular velocity and the time duration calculated in the last section, Euler parameter at any moment during the motion can be found.

3.1.3 Determine the Orientation as Functions of Time

While the die is rotating around the orientational axis during the motion in air, each point inside the cube does not change its coordinate during the motion in embedded frame $\xi\eta\zeta$. The cube is always at rest in embedded frame [13]. For the general xyz frame, the coordinate of each point in the cube keeps changing while the rigid body moves and revolves. To find the new coordinate of the point after a motion, a transformation matrix is needed. Transformation matrices can be used to find the new position of any vectors start from the origin of the embedded frame [13]. The origin of embedded frame is set to be the mass center of the cube. In the embedded frame, the direction of any vectors do not change no matter how the cube moves and rotates. The illustration below shows how such a vector change its direction with the cube in xyz frame:

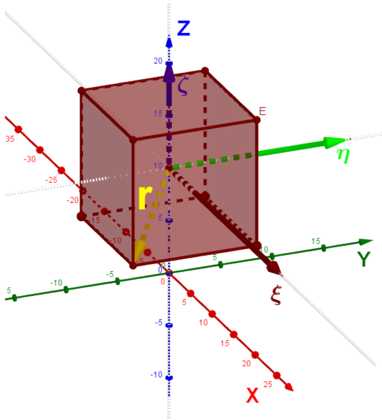

Figure 11: Object before rotation

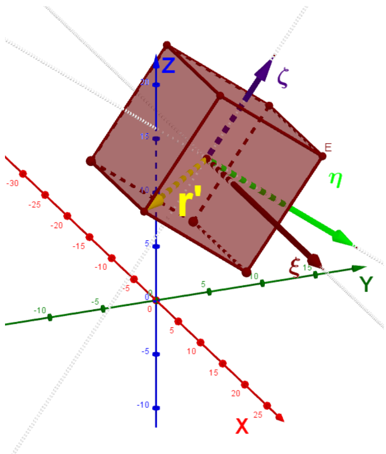

Figure 12: Object after rotation

In the first graph, r is the vector from the origin, the mass center of the cube, to one of the vertex of the cube (the end of the vector can be any point inside the cube). The second graph shows the shift of vector r after the rotation, and it is denoted by r'. The matrix that transformed r to r' is the transformation matrix, which is denoted by R during this research. The matrix that can transform r' to r is the inverse of matrix R, or R'. For any vector r [13]:

$$r' = R^{-1}r \tag{35}$$

Because transformation matrix shows how a vector rotates, it is often used as a tool to show the orientation of an object. Given an initial vector when embedded frame coincides with general frame, the orientation of the object can be represented by the transformation matrix that transforms the initial vector to its current position.

The transformation matrix can be calculated by Euler angles or Euler parameters [15]. Here, the Euler parameters are used because each component of Euler parameter is a function of time. Using Euler parameters, the transformation matrix can be also written as functions of time:

$$R = \begin{bmatrix} -1+2e_0^2+2e_1^2 & 2e_1e_2-2e_0e_3 & 2e_0e_2+2e_1e_3 \\ 2e_1e_2+2e_0e_3 & -1+2e_0^2+2e_2^2 & -2e_0e_1+2e_2e_3 \\ -2e_0e_2 & 2e_0e_1+2e_2e_3 & -1+2e_0^2+2e_3^2 \end{bmatrix} \tag{36}$$

With transformation matrix, the orientation at any time during the die's motion in air can be derived.

3.2 Bouncing

3.2.1 Determine the Euler Angles before Bouncing

Bouncing of the die after its collision with the ground surface is a complex process involves the considerations of a variety of quantities. Many of these quantities can be calculated from the die's free fall, and there are other quantities required to be found through analyzing the die's contact with the ground.

To analyze the bouncing, the first step is to find the die's orientation in Euler angles (ψ, θ, ϕ) before the bouncing. The orientation is already calculated through the time function of Euler parameters. To find the Euler angles, an equivalent relationship between orientation representation needs to be set up. To find this relationship, the transformation matrix is applied. When a set of Euler angles and a set of Euler parameters can lead to a same transformation matrix, a relationship between Euler angles and Euler parameters can be found.

To write a transformation matrix in the form of Euler angles, the following formula is applied [13]:

$$R = R_\zeta(\psi)R_\xi(\theta)R_\zeta(\phi) \tag{37}$$

In the formula above, ψ, θ, ϕ are Euler angles. R_ζ, R_θ, and R_ϕ are the transformation matrix for each orientation's rotation around three embedded coordinate's axes. Defined as classic Euler angles, the three rotations must follow the order of ζ-ξ-ζ, so these three transformation matrices in the formula must follow the order listed above.

For the Euler angle ψ rotates around the ζ axis, the matrix is given as the following form:

$$R_\zeta = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{38}$$

For the Euler angle θ rotates around the ξ axis, the matrix is given as the following form:

$$R_\xi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{39}$$

Thus, the transformation matrix R is:

$$R = \begin{bmatrix} c\phi c\psi - c\theta s\theta s\phi & -c\psi s\phi - c\theta c\phi s\psi & s\theta s\psi \\ c\theta c\phi s\phi + c\phi s\psi & c\theta c\phi c\psi - s\phi s\psi & -c\psi s\theta \\ s\theta s\phi & c\phi s\theta & c\theta \end{bmatrix} \tag{40}$$

In the matrix above, c represents $\cos(\cdot)$, and s represents $\sin(\cdot)$

Equation [40] is equivalent to Equation 36. With two equivalent matrices, nine equations with three unknown variables is set. Through solving those equations, the Euler angles at any time can be derived.

3.2.2 Determine Euler Rates

Using Euler parameters, the time function of angular velocity components in embedded frame and Euler angles can be derived. After knowing the value of angular velocity's components and Euler angles, the time derivative of Euler angles can be found. Another name for the time derivatives of Euler angles is Euler rate.

Euler rates, $\dot{\psi}, \dot{\theta}, \dot{\phi}$, are not the same as the components of angular velocity along ξ, η, and ζ axes: Euler angles are used not angles that the cube rotates; angular velocity are really about how cube rotates in the air. The correlations between time derivatives of Euler angels and components of angular velocity given by Wolfram Mathworld are given below [20]:

$$\omega_\xi = \dot{\psi}\sin\theta\sin\phi + \dot{\theta}\cos\phi \tag{41}$$

$$\omega_\eta = \dot{\psi}\sin\theta\cos\phi - \dot{\theta}\sin\phi \tag{42}$$

$$\omega_\zeta = \dot{\psi}\cos\theta + \dot{\phi} \tag{43}$$

After solving for $\dot{\psi}, \dot{\theta}, \dot{\phi}$ from the equations listed above, the last step for the preparation before bouncing equations is to find the object's state before its contact with the ground.

3.2.3 Determine How the Die Contact with Ground

When a die lands on a surface after its motion in air, there must be a contact point and a side that project the largest shadow on the surface. A cube may also got minimal chance to hit the ground with an edge or a plane surface. However, compared to the chances of hitting the ground with a vertex, other cases are so rare that they could be neglected without causing noticeable errors.

The coordinate of the contact point A in the xyz general frame is not imporant because in this project, the surface that the die is going to land is homogeneous. The coordinate of point A in the embedded frame, in contrast, is important because the nonuniform density distribution shifts COM, and the coordinates of contact points changes when the origin shifts. Since there are only eight vertexes for a cube, and each vertex's coordinate in the embedded frame does not change, so the coordinate of A is always one of the eight coordinates: the coordinate of A can be determine as long as the contact point is determined. To determine which point contacts with the ground, eight vectors is drawn from the center of mass (the origin) across each vertex of the cube. The following graph shows how the eight vectors look like:

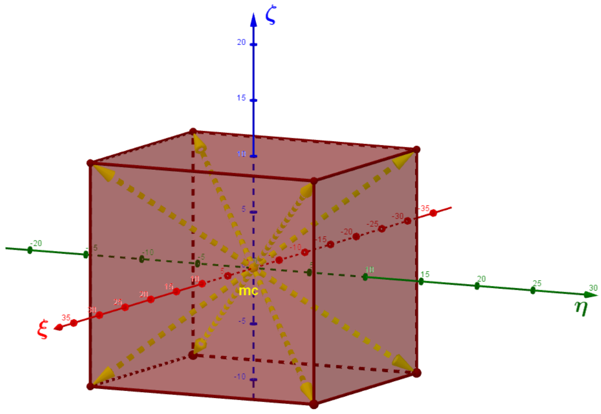

Figure 13: Eight Vectors

The yellow point m_c at the middle of the cube is the COM of the cube, and it is also the origin of the embedded frame. The eight yellow arrows are all drawn from the mass center to eight different vertexes of the cube. The unit vector for each of these eight vectors can be easily solved by applying simple geometry rules, and those are the initial directions of these vectors. The initial directions of these eight vectors is depended on the function of density distribution. To find the direction of these vectors before the bouncing, transformation matrices are used. With known Euler angles and Euler parameters, the transformation matrix can be found. The new direction of these eight vectors in the general frame will be calculated through formula $r' = R^{-1} \cdot r$. To find which vertex will hit the ground, another fixed frame $x'y'z'$, which is parallel to the general frame xyz, will be built also with the origin at the mass center. The angles between each one of the eight vectors after the rotation and the z' axis is calculated from law of cosine. The vector with the smallest angle between itself and the z' axis will be the vector that passes through the vertex contacted with the surface. The following graph shows about the determination of contact point A:

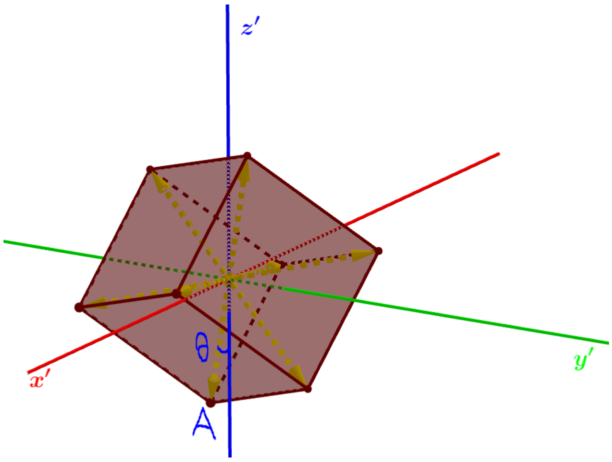

Figure 14: Eight Vectors after Rotation

In the graph above, the eight vectors are all in their new position, and from the graph it's easy to observe that different vectors have different angles with the -z' axis. The vector pointing at vertex A has the smallest angle with the z' axis, and the angle difference is denoted by a blue-colored θ, so the conclusion is that A is the point contact with the ground before bouncing, and its coordinates in the embedded frame will be used in calculation for bouncing. The general form of Law of Cosine is shown as below:

$$\cos\theta = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||||\vec{v}||} \tag{44}$$

Where $\vec{u}$ and $\vec{v}$ are two vectors that the angle in between will be solved. In this study, $\vec{u} and \vec{v}$ will be the vectors from origin to vertex and z' axis.

The side that projects the largest shadow is the side that is going to land if there is no bouncing. Because of the bouncing, the die starts off its motion in air again with lower velocities. The side projecting the largest shadow can be used to determine the final outcome of a toss when the bouncing stops. An illustration of this side is shown below:

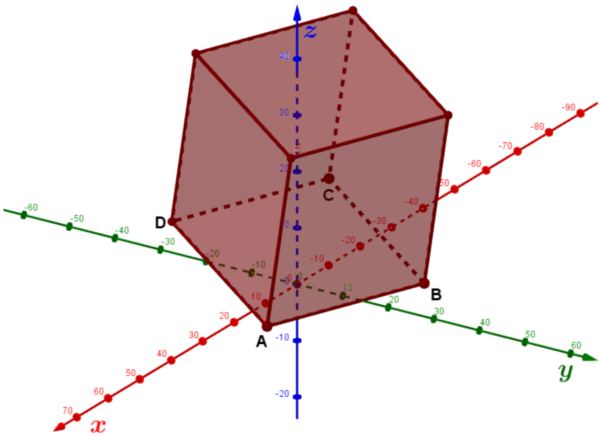

Figure 15: A Landing Die

In this graph, point A is the point of contact, and $ABCD$ is the side that projects the largest shadow.

To find this side, the similar method for determining contact point A is applied. Instead of drawing vector from origin to the eight vertices of the cube, six vectors from the origin (COM) are drawn in the directions perpendicular to each side of the cube. To find the vector's components after the rotation, the transformation matrix is used. After finding the direction of those six vectors at the moment just before bouncing, anther fixed frame parallel to the xyz frame is used again, and the angles between the six vectors and the z' frame will be compared. The vector will the smallest angle with z' axis will be the vector passing through the side that is most likely to land. The scenario is shown in the graph below:

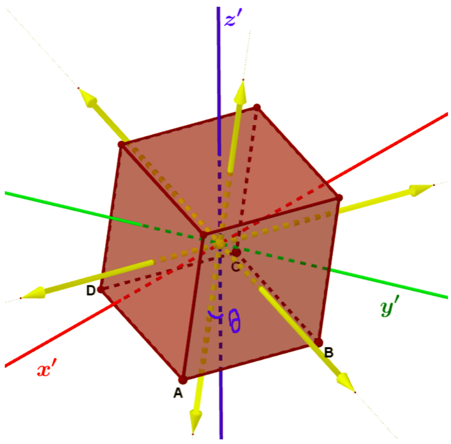

Figure 16: The Six Vectors

In the graph above, the six yellow vectors are those vectors which are perpendicular to each side of cube, and one of the vector has the minimal θ degree with the z' axis. That means the surface passed through by this vector, $ABCD$, will be the side project the most shadow to the ground.

3.2.4 Determine the Energy Loss

When the die bumps with the surface, part of its energy will dissipate during the collision. The amount energy that will be dissipated during each collision can determine how many times the die can bounce and how the angular velocities of the die changes during the collision. The coefficient of restitution is defined to be [16]:

$$-\chi = \frac{v'_{Az}}{v_{Az}} \tag{45}$$

In this equation, v_{A_z} represent the speed of the contact point A in the direction of z axis before the collision, while v'_{A_z} represents the speed of point A in the direction of z axis after the collision. During this research, the size of die is very small compared to the distance it travels, so v_{Az} is assumed to be equal to v_{COM}. The coefficient of restitution depends on the material's properties of two objects that collide. To make the model more accurate, the cube has two coefficient of restitution because it's made up of two materials. In our experiments, only two materials will be used, the stainless iron and aluminum alloy (Type 6061). The detailed procedures for calculating the coefficient of restitution for those two materials with the landing surface will be presented later in the experiment section.

3.2.5 Determine the Euler Rates After Bouncing

After determining the values of moment of inertia (about the orientational axis), Euler angles, time derivative of Euler angels, embedded frame coordinate of contact point A, and the coefficient of restitution at the moment before the bouncing, amd the transnational velocity and acceleration, it is finally well-prepared for the bouncing equation.

For this equation, it is assumed that there is no fiction between the die and the surface that the die is going to land on. It is also assumed that during the bouncing there is only an impulse in the z-direction denoted by S_z. The embedded frame coordinates are denoted by (ξ_A, η_A, ζ_A) in the formulas. With these assumptions, the equation for bouncing is given as follows [8]:

$$\begin{aligned}
&\dot{z}' + \dot{\theta}' + (-\zeta_A \cos{(\phi+\psi)} \sin\theta + \cos\theta \cos(\phi+\psi)(\eta_A \cos\phi \xi_A \sin\phi) \\
&(\xi_A \cos\phi - \eta_A \sin\phi) \sin(\phi+\psi)) \\
&+\dot{\psi}' \sin\theta(\xi_A \cos\theta \cos\psi \sin^2\phi + \eta_A \cos^2{(\tfrac{\theta}{2})} \cos\psi \sin{(2\phi)} - \eta_A \sin^2\phi \sin\psi \\
&+\xi_A \cos^2{(\tfrac{\theta}{2})} \sin(2\phi) \sin\psi + \cos^2\phi(-\xi_A \cos\psi + \eta_A \cos\theta \sin\psi) - \zeta_A \sin\theta \sin(\phi+\psi)) \\
&= -\chi(\dot{z} + \dot{\theta} + (-\zeta_A \cos(\phi+\psi) \\
&\sin\theta + \cos\theta \cos(\phi+\psi)(\eta_A \cos\phi + \xi_A \sin\phi) + (\xi_A \cos\phi \\
&-\eta_A \sin\phi) \sin(\phi+\psi)) + \dot{\psi} \sin\theta(\xi_A \cos\theta \cos\psi \sin^2\phi \\
&+\eta_A \cos^2{(\tfrac{\theta}{2})} \cos\psi \sin(2\phi) - \eta_A \sin^2\phi \sin\psi \\
&+\xi_A \cos^2{(\tfrac{\theta}{2})} \sin(2\phi) \sin\psi + \cos^2\phi(-\xi_A \cos\psi \\
&+\eta_A \cos\theta \sin\psi) - \zeta_A \sin\theta sin(\phi+\psi))
\end{aligned} \tag{46}$$

$$\dot{x}' = \dot{x} \tag{47}$$

$$\dot{y}' = \dot{y} \tag{48}$$

$$m\dot{z}' - S_z = m\dot{z} \tag{49}$$

$$J(\dot{\theta}' \cos\psi + \dot{\phi}' \sin\theta \sin\phi) = J(\dot{\theta} \cos\psi + \dot{\phi} \sin\theta \sin\phi) + S_z\eta_A \cos\theta - S_z\zeta_A \cos\phi \sin\theta \tag{50}$$

$$J(\dot{\theta}' \sin\psi - \dot{\phi}' \cos\theta \sin\phi) = J(\dot{\theta} \sin\psi - \dot{\phi} \cos\theta \sin\phi) - S_z\xi_A \cos\theta + S_z\zeta_A \sin\phi \sin\theta \tag{51}$$

$$J(\dot{\phi}' \cos\theta + \dot{\psi}') = J(\dot{\phi} \cos\theta + \dot{\psi}) + S_z\xi_A \cos\phi \sin\theta - S_z\eta_A \sin\theta \sin\phi \tag{52}$$

In equations (47 - 52), $\dot{x}, \dot{y}, \dot{z}, \dot{\psi}, \dot{\theta}, \dot{\phi}, \psi, \theta, \phi, \xi_A, \eta_A, \zeta_A, J,$ and χ are the quantities that have been found before. $\dot{x}', \dot{y}', \dot{z}', S_z, \dot{\psi}', \dot{\theta}', \dot{\phi}'$ are the unknowns that need to be solved. The quantities with $'$ sign at the right upper head means that those quantities are values after the bouncing, like $\dot{z}', S_z, \dot{\psi}', \dot{\theta}', \dot{\phi}'$. Because there is no acceleration on the linear fixed frame's x and y direction, their solutions after bouncing is very obvious. For other quantities, five equations with five unknowns are solved by Wolfram Mathematica.

3.2.6 Cycles of Calculation

Now a new set of Euler rates and the new z-component of translational speed are derived. Euler angles does not change during the process of collision. Because the dice will bounce again, the same calculations will be done again and again until the terminal statement, in our research is when $\dot{z} \leqslant 0.001\text{m/s}^2$. To start the calculation all over again, the new group of "initial" quantities need to be derived:

1. The orientation in Euler angles (ψ, θ, ϕ)

   Euler angles remain unchanged during collisions.

2. The unit vector along orientational axis v^0

   In equations (41) - (43), with known Euler rates and Euler angles, $\omega_\xi, \omega_\eta,$ and ω_ζ can be solved. After solving the exact value for three angular velocity's components, the new v^0 will be the unit vector of vector $[\omega_\xi, \omega_\eta, \omega_\zeta]^T$. Without a torque exerted on the dice after bouncing, the direction of v^0 will keep unchanged till next collision.

3. The magnitude of angular velocity around the orientational axis of rotation $\dot{\alpha}$

   $\dot{\alpha}$ is the vector resultant of $\omega_\xi, \omega_\eta,$ and ω_ζ. Cause the three components of the vector is already known, the Equations 23 will be applied, and the value for $\dot{\alpha}$ will be easily solved.

4. The initial linear velocity's components in the xyz frames (v_x, v_y, v_z)

   The components v_x and v_y do not change their values because the value of $\ddot{x}$ and $\ddot{y}$ keeps constant when there is no external force exerted on these two directions. The new $\dot{z}$ is determined from the bouncing equations and used as the new initial $\dot{z}$ for the new row of calculation. $\ddot{z}$ always remains the same as $g = 9.8\text{m/s}^2$.

5. The density distribution of the dice

   The distribution function does not change throughout the whole process of calculation.

6. Coefficient of restitution

Coefficient of restitution only depends on which vertex is going to hit the surface. The vertex hitting the surface is determined following the same procedures presented in Section 3.2.3

7. Height

The height of the mass center will be determined by the Euler angles and contact point A's coordinate. The height of point A is assumed to be zero, so from geometric deduction process presented in Section 3.2.3 the height of mass center can be determined.

Now the new group of initial quantities are all found. Following all procedures presented from Section 2.5 to Section 3.2.6 repeatedly, the model will work in a loop until the terminal condition is reached and the final outcome.

3.3 Determination of Final Outcomer the bouncin

When the z-component of translational speed turns less than or equal to 0.001 m/s, the critical condition is reached. At such a small speed, a 30 mm$\times$ 30 mm$\times$30 mm die is unable to bounce up, and the side that projects the largest shadow before the critical condition is reached become the final outcome of the calculation about a single toss.

4 Numerical Simulation

4.1 Flowchart of the Program

The main principle of the program is demonstrated in the flowchart below.

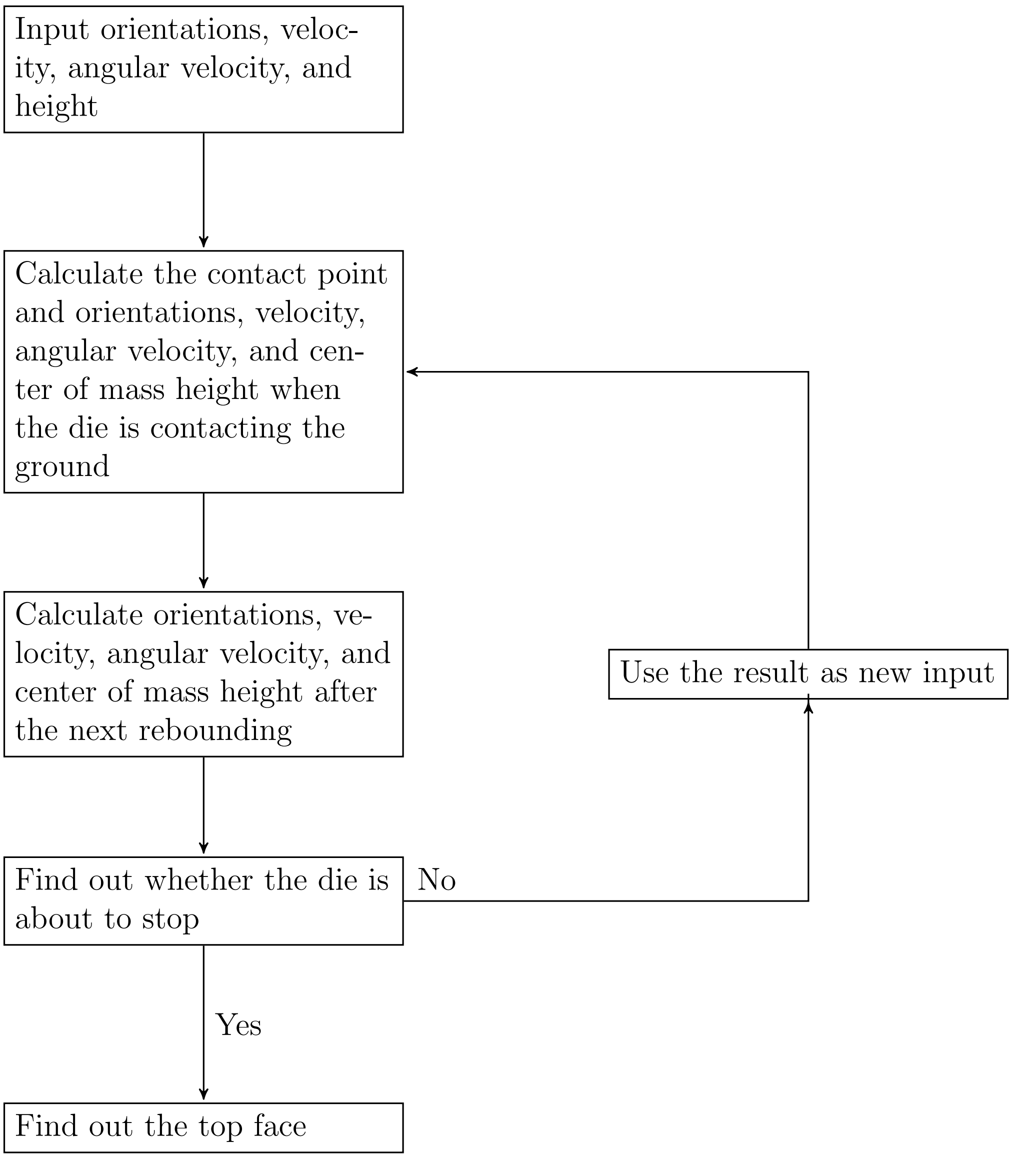

Figure 17: Flowchart of the Program

4.2 Variables

We used a random number generator to determine random input values.

Some variables follow normal distribution with the some mean and variance, and some follow uniform distribution.

In random case, the variables often have a large variance. If we set the initial values of some variables, the variables then have a small variance.

The tables below show the distribution of variables. $\mathrm{N}(\mu, \sigma^2)$ is normal distribution with mean μ and variance σ^2. $\mathrm{Uni}(a, b)$ is uniform distribution in the interval (a, b).

Variable	**Distribution**
ξ	$\mathrm{Uni}(0, 2\pi)$
η	$\mathrm{Uni}(0, 2\pi)$
ζ	$\mathrm{Uni}(0, 2\pi)$
v_x	$\mathrm{N}(0, 0.5)$
v_y	$\mathrm{N}(0, 0.5)$
v_z	$\mathrm{N}(0, 0.5)$
ω_x	$\mathrm{N}(0, 18)$
ω_y	$\mathrm{N}(0, 18)$
ω_z	$\mathrm{N}(0, 18)$
h	$\mathrm{N}(1, 0.2)$

Table 2: The Distributions of Variables in Random case

Variable	**Distribution**
orientation	Fixed
v_x	$\mathrm{N}(\mu_{v_x}, 0.05)$
v_y	$\mathrm{N}(\mu_{v_y}, 0.05)$
v_z	$\mathrm{N}(\mu_{v_z}, 0.05)$
$\|\|\omega\|\|$	$\mathrm{N}(\mu_\omega, 4)$
direction of ω	Random direction
h	$\mathrm{N}(\mu_h, 0.05)$

Table 3: The Distributions of Variables in Non-Random case

4.3 Statistical Analysis

Our program can simulate the toss of the die with some random initial conditions.

We use statistical Analysis to calculate the probability. The principle of this method is, in order to find out the true probability, we take a very large random sample, and we use the sample to infer the range to which the true probability belongs.

We can use our program to "toss" lots of times (for example, ten thousand times). Then, we use a statistic analysis to construct a confidence interval of the probability. Probability is a proportion, so we can use the method of constructing a confidence interval for proportion.

A confidence interval with $m\%$ confidence level means for $m\%$ probability, the true value is in the confidence interval.

The data we got from our program are considered as "random sampling."

Suppose there are n tosses. We calculate the variance of the proportion by

$$\sigma_{\hat{P}(x)} = \sqrt{\frac{\hat{P}(x)\left(1 - \hat{P}(x)\right)}{n}}, \tag{53}$$

Where $\hat{P}(x)$ is the probability calculated from the tosses. The true probability is denoted by $P(x)$.

Find the critical value of Binomial Distribution, or the B-value which depends on the confidence level (CL) and number of trials (n). They can be easily found from a table of B-values or from a calculator. For $n > 30$, Binomial Distribution is very similar to Standardized Normal Distribution, so we can instead use the critical value of Standardized Normal Distribution, or Z-values, which can also be found from a table of Z-values or from a calculator.

If the confidence level is 95%,

$$Z_{0.95} \approx 1.96\ . \tag{54}$$

The confidence interval is

$$P(x) \in \left(\hat{P}(x) \pm \sigma_{\hat{P}(x)} \cdot Z_{\mathrm{CL}}\right), \tag{55}$$

and as Equation (53), $\sigma_{\hat{P}(x)} = \sqrt{\frac{\hat{P}(x)\left(1-\hat{P}(x)\right)}{n}}$.

For a 95% confidence interval,

$$P(x) \in \left(\hat{P}(x) \pm 1.96\sqrt{\frac{\hat{P}(x)\left(1 - \hat{P}(x)\right)}{n}}\right). \tag{56}$$

In our programs, we "toss" for 10,000 times. After calculation, our confidence interval is very narrow, causing an error bound of less than 0.01. The narrow confidence interval indicates that the probability we have calculated is very similar to the true probability.

4.4 Special Cases

One of the special cases may occur when the die reaches into equilibrium. It is nearly impossible for a die to stop with only one contact corner or edge instead of a face. But in our program, this scenario may occur, though the probability is very small. In order to fixed this problem, we exert a very tiny force with random direction on the die. Because this scenario rarely occur and the force is small and random, it will not change the overall probability.

Another special case is when the die contacts the ground, the whole edge or face may contact at the same time, though the probability is very small. This may not happen in the real world, but may happen in our program. In this case, we consider the contact point as one of the corners on this edge or face and select randomly. Because of the randomness and rareness, this will not affect the overall result.

5 Experiments, Simulations, and Results

5.1 Determine the Coefficient of Restitution

Purpose: Find the coefficient of restitution between the stainless metal and plastic and the coefficient of restitution between aluminum alloy and plastic.

5.1.1 Introduction

The coefficient of restitution is a constant that determines how much energy is lost after each collision between a die and the ground. The coefficient of restitution determines how many times the die will bounce up before it stops moving. This constant also plays a significant role when determine the dice's angular velocity and Euler rates after the collision with the ground. Coefficient of restitution is usually determined by the materials of two collided objects. In this research object, all experimental specimen is made of two different metals: stainless steel and aluminum alloy (Type 6061), and all specimens land on the same surface: a plastic tennis court. There will be two coefficient of restitution involved in this experiment: one is between stainless steel and plastic; the other is between aluminum alloy and plastic. The purpose of this experiment is to find those two coefficients of restitution and put those values into the stimulating program to make the results more realistic.

By definition, the coefficient of restitution is given as [16]:

$$-\chi = \frac{v'_{Az}}{v_{Az}} \tag{57}$$

In the formula above, χ is the coefficient of restitution, v_{A_z} is the z-component of contact point A's velocity before bouncing; v'_{A_z} is the z-component of contact point A's velocity after bouncing.

Because the cube is comparatively small compared to the height it is dropped from, the velocity of contact point A is approximately the same as the velocity of the cube as a whole. To find the coefficient of restitution, the z-components of a cube's velocity before and after the die bounce up are required be found through experiments.

5.1.2 Procedures

To prepare for this experiment, a cube composed of two same-volume metal prisms, a smooth and elastic surface (plastic tennis court), and a slow motion camera are needed. For the experiment in this research project, the specimen has the following property:

15 mm × 30 mm × 30 mm stainless iron and 15 mm × 30 mm × 30 mm aluminum alloy

Figure 18: The Sample Die

To conduct the experiment, a person needs to hold the specimen and another person is responsible for shooting the scene. After the camera starts to work, the person holding the cube release it from his/her hand without giving it any initial translational and angular speed. The person keeps picking up the cube and dropping it ollowing the same instruction. After shooting around 20 valid scenes, experimenters need to upload their videos to a software named "Logger Pro" and draw the motion line of the cube. From the slow motion, experimenters can know which collision are between stainless steel and surface and which collision are between aluminum alloy and the surface. Calculating the derivatives of the cube's motion lines, velocities before and after the bouncing can be found. For each type of bouncing, there should be at least six valid groups of velocities. A value is calculated from each group of velocities, and the coefficient of restitution is determined through finding average values among the values calculated.

5.1.3 Results

The following two graphs are chosen from 12 of our valid videos as samples. The first graph shows a bouncing when stainless-steel corner of the cube hits the ground.

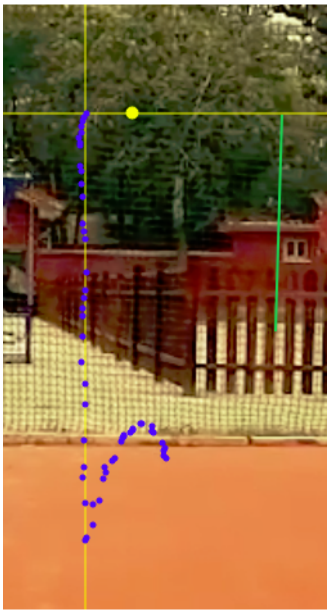

Figure 19: Motion of Die with Steel

Figure 20: Motion of Die with Aluminum

The time function of vertical distance of those two motions are shown below. In those two graphs, the blue line is the function for the z direction displacement, and the red line is time function of the displacement in x-y plane.

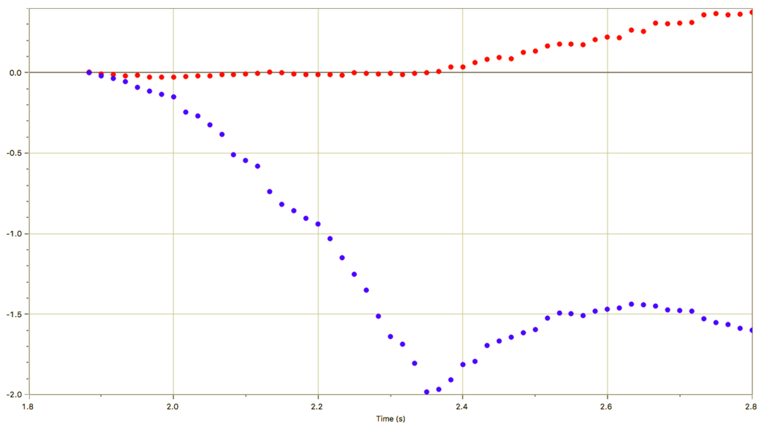

Figure 21: Positions of The Die-Contact Point is Steel

The second graph shows a bouncing when aluminum alloy corner of the cube hits the ground.

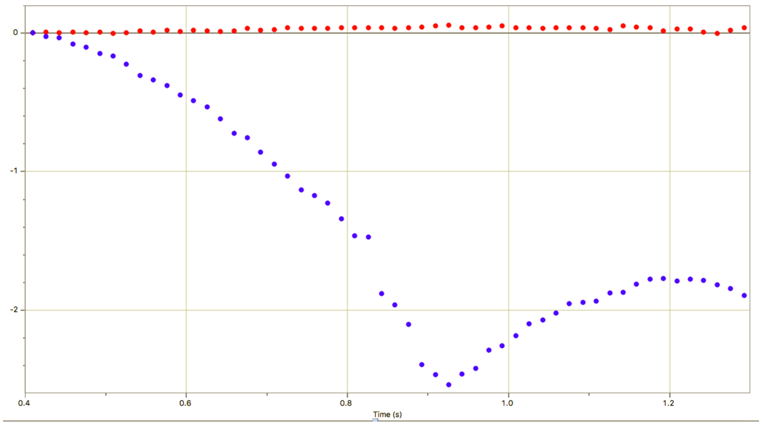

Figure 22: Positions of The Die-Contact Point is Aluminum

The first graph is for the collision with stainless steel. The die gained some speed in x-y direction after bounced up due to the frictional force, but compared to the speed in z-direction, the speed in x-y direction is comparatively small, so it is ignored during the calculation. In the second graph, the one for aluminum alloy, there is no motion in x-y plane.

The derivatives of those two time functions before and after the lowest points in the graphs are the speeds before and after collision. The data from 6 collisions between stainless steel and plastic tennis court are shown below:

Trial	1	2	3	4	5	6
V	6.2556 m/s	6.5341 m/s	6.7341m/s	6.3472 m/s	6.1022 m/s	6.4235 m/s
V'	3.1422 m/s	3.0543 m/s	3.2453 m/s	3.1254 m/s	2.9583 m/s	3.1654 m/s

Table 4: Data of Measured Speeds of Stainless Steel

From the data shown in the table above, the average value of coefficient of restitution between stainless steel and plastic tennis court is calculated as:

$$\chi_{\text{steel}} = 0.48949 \tag{58}$$

The data from 6 collisions between aluminum alloy and plastic tennis court as shown below:

Trial	1	2	3	4	5	6
V	6.3721 m/s	6.1352 m/s	6.2412 m/s	6.1252 m/s	6.2315 m/s	6.4312 m/s
V'	3.3253 m/s	3.5423 m/s	3.4831 m/s	3.4081 m/s	3.3462 m/s	3.5312 m/s

Table 5: Data of Measured Speeds of Aluminum

From the data shown in the table above, the average value of coefficient of restitution between aluminum alloy and plastic tennis court is calculated as:

$$\chi_{\text{aluminum}} = 0.48949 \tag{59}$$

5.2 Probabilities of Dice with Different Density Distribution

Purpose: To find the probabilities for each final outcomes under three different density distributions.

5.2.1 Introduction

When a die's density distribution is not uniform anymore, the probabilities for its outcomes will all be changed. The influence of density distribution on the possibility of each outcome of a die toss is one of the most important topics in this research project. By having three specimens with different density distributions, a qualitative tendency of how density distribution changes possibilities can be found. Besides finding a tendency, the experiment also serves as a tool to prove the correctness of the simulating mathematical model in the research project. By comparing the results from this experiments and the data calculated through modeling, the validness of the model can be tested. In this experiment, all specimens are cubes with different volume ratio between stainless steel and aluminum alloy. Two out of three specimens are made up of two rectangular prisms, which have the simplest density distribution function. The special one is made up of two triangular prisms, and the volume ratio is 1:1. The density for stainless steel in this project is 7.848g/cm^3; the density for aluminum alloy in this research is 2.7143g/cm^3. Due to the large difference between the two metals, the COM of the cube will be effectively shifted away from its geometric center by a nonuniform density distribution, and the change of probability for each outcome will be easy to observe. Another reason for choosing stainless steel and aluminum alloy is that they are all stiff and can withstand many collisions without deforming. Each side of a die if denoted by one of the six numbers. The side with full stainless steel is denoted by number 1. The side with full aluminum alloy is denoted by number 6. The side with two metals mixing up is denoted by number 2, 3, 4, and 5, which side 2 is parallel to side 5, and side 3 is parallel to side 4.

5.2.2 Procedures

In order to prepare for this experiment, specimens with following properties are needed: Specimen 1: 15 mm × 30 mm × 30 mm stainless steel and 15 mm ×

30 mm × 30 mm aluminum alloy

Figure 23: The First Die

Specimen 2: 24 mm × 30 mm × 30 mm stainless steel and 6 mm × 30 mm × 30 mm aluminum alloy

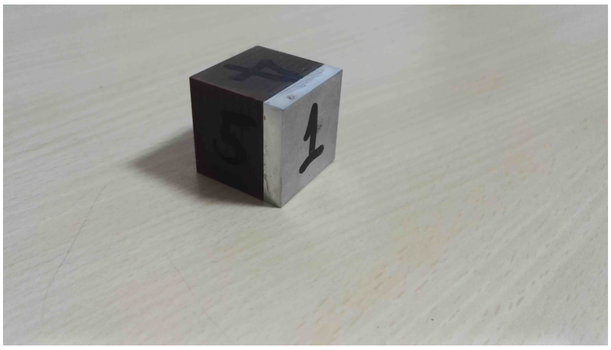

Figure 24: The Second Die

Specimen 3: right triangular prism 30 mm × 30 mm × 30 mm stainless steel and right triangular prism 30 mm × 30 mm × 30 mm aluminum alloy.

Figure 25: The Third Die

To find the possibility of each final outcome, each die is thrown for 500 times. The model is used to simulate real-life scenarios when testing different volume ratio, so it's not required to throw the die with some certain initial orientations or velocities: just throw the dies like how people normally do.

Die Sample	Die 1	Die 2	Die 3
$P(1)$	0.27 ± 0.0389	0.208 ± 0.0356	0.246 ± 0.0378
$P(2)$	0.166 ± 0.0326	0.17 ± 0.0329	0.158 ± 0.0320
$P(3)$	0.148 ± 0.0311	0.16 ± 0.0321	0.156 ± 0.0318
$P(4)$	0.17 ± 0.0329	0.182 ± 0.0338	0.164 ± 0.0325
$P(5)$	0.162 ± 0.0323	0.166 ± 0.0326	0.172 ± 0.0331
$P(6)$	0.086 ± 0.0246	0.114 ± 0.0279	0.108 ± 0.0272

Table 6: Results of Tosses of The Die Samples

5.3 Probabilities of Dice at Different Heights

Purpose: To find how the vertical displacement from the die to the landing surface influences the probabilities of each final outcome of die toss.

5.3.1 Introduction

Besides the density distribution, the height of the dice is another possible factor that may influence the possibilities for each final outcome. By tossing the same die from three different heights, a qualitatively tendency of how height influence the possibilities can be found. In this experiment, a special scenario is discussed: all other initial variables excluding height and orientation are intended to kept as constant because finding the direct influence of height is the goal of this experiment. However, in real life it's hard to kept throwing a dice at different heights with same velocities, so both translational and angular velocities are kept as zero. How each number corresponds to each side of the cube is the same as Section 5.2.

5.3.2 Procedures

In this experiment, the cube with 15 mm $\times$ 30 mm $\times$ 30 mm stainless steel and 15 mm $\times$ 30 mm $\times$ 30 mm is used. Another tool required for this experiment is a one-meter long ruler. While doing the experiment, release the dice from 1 m, 0.8 m, 0.4 m: each for 500 times. The height of a die is the displacement between its COM to the surface ground, so while holding the dice, make sure it's the COM at 1 m. 0.8 m, or 0.4 m, instead of the geometric center. It's hard to release the die from exactly that much, so a small variance is also included for the height's distribution when simulating the scenario. While releasing the die from the hand, try to give the die no initial translational velocity and angular velocity.

Figure 26: The Sample Die

5.3.3 Results

The data for toss from 1 m is shown below:

Height (m)	1	0.8	0.4
$P(1)$	0.086 ± 0.025	0.086 ± 0.025	0.108 ± 0.0272
$P(2)$	0.166 ± 0.033	0.166 ± 0.033	0.158 ± 0.0320
$P(3)$	0.148 ± 0.031	0.148 ± 0.031	0.156 ± 0.0318
$P(4)$	0.17 ± 0.033	0.17 ± 0.033	0.164 ± 0.0325
$P(5)$	0.162 ± 0.032	0.162 ± 0.032	0.172 ± 0.0331
$P(6)$	0.27 ± 0.039	0.27 ± 0.039	0.246 ± 0.0378

Table 7: Results of Tosses from Different Heights

5.4 Numerical Simulation of Die Toss with Different Volume Ratios

Purpose: To observe how different volume ratio influences the possibilities of each outcome through computer simulation.

5.4.1 Data Sampling

In this simulation, the volume ratio is the volume of aluminum alloy over the volume of the whole cube. The value of the volume ratio varies from $\frac{1}{30}$to$\frac{30}{30}$, and there are 30 different volume ratios in total. For other variables: the three initial Euler angles, $\psi, \theta, \text{and} \phi$ are chosen from random within the range of $[0, 2\pi]$. The magnitude of angular velocities, $\dot{\alpha}$ follows a normal distribution with an expected value of 0 rad and a variance of 18 rad. Three components of orientational axis, $v^1, v^2, \text{and} v^3$ are random numbers chosen from a range of [0,1], after being chosen, those three components are normalized. Three components of translational velocity, $v_{x0}, v_{y0}, \text{and} v_{z0}$ follow normal distributions with an expected value of 0 and variance of 0.5 m/s. The density ratio is kept constant as $\frac{2714}{7848}$. The coefficients of restitution, χ are kept constant as 0.49 (aluminum) and 0.55 (steel). The height, h_0 follows a normal distribution with an expected value of 1m and a variance of 0.2m. For each volume ratio, 10,000 groups of initial variables are calculated.

5.4.2 Results of Simulation

The probabilities of each die's six outcomes are presented in the following table and graph:

Ratio	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$
1/30	0.1564	0.1613	0.1833	0.1565	0.1583	0.1842
2/30	0.1529	0.1624	0.1848	0.1633	0.1543	0.1823
3/30	0.1591	0.1612	0.1732	0.1654	0.1568	0.1843
4/30	0.1513	0.1575	0.1841	0.1571	0.1616	0.1884
5/30	0.1377	0.16	0.1831	0.1649	0.1637	0.1906
6/30	0.1416	0.1604	0.1821	0.1603	0.1628	0.1928
7/30	0.1408	0.1697	0.1733	0.1695	0.161	0.1857
8/30	0.1459	0.1605	0.1655	0.1611	0.1623	0.2047
9/30	0.1268	0.1646	0.1843	0.1585	0.1637	0.2021
10/30	0.1308	0.16	0.1816	0.163	0.1613	0.2033
11/30	0.1335	0.1652	0.1708	0.1689	0.1651	0.1965
12/30	0.1222	0.1632	0.1782	0.1587	0.1638	0.2139
13/30	0.1339	0.1607	0.1669	0.1598	0.1634	0.2153
14/30	0.1262	0.1543	0.1732	0.1667	0.1568	0.2228
15/30	0.1209	0.1627	0.1723	0.1632	0.1562	0.2247
16/30	0.1266	0.1598	0.1653	0.1597	0.1605	0.2281
17/30	0.1155	0.1584	0.1703	0.158	0.1623	0.2355
18/30	0.0981	0.1664	0.1693	0.1607	0.1574	0.2481
19/30	0.0802	0.1556	0.1839	0.1622	0.159	0.2591
20/30	0.083	0.1605	0.1829	0.1587	0.1618	0.2531
21/30	0.0945	0.1601	0.1661	0.1622	0.1568	0.2603
22/30	0.1066	0.1552	0.1847	0.1591	0.1645	0.2299
23/30	0.111	0.166	0.1812	0.1542	0.1648	0.2228
24/30	0.1133	0.1622	0.1807	0.153	0.1645	0.2263
25/30	0.1247	0.1554	0.1733	0.1626	0.1649	0.2191
26/30	0.1321	0.1579	0.181	0.162	0.1589	0.2081
27/30	0.1461	0.1655	0.1817	0.1603	0.1582	0.1882
28/30	0.1616	0.156	0.1748	0.1642	0.1608	0.1826
29/30	0.1696	0.1558	0.1754	0.1598	0.1621	0.1773
30/30	0.1633	0.1581	0.1841	0.1598	0.1629	0.1718

Table 8: The Probabilities of Dice with Different Volume Ratios

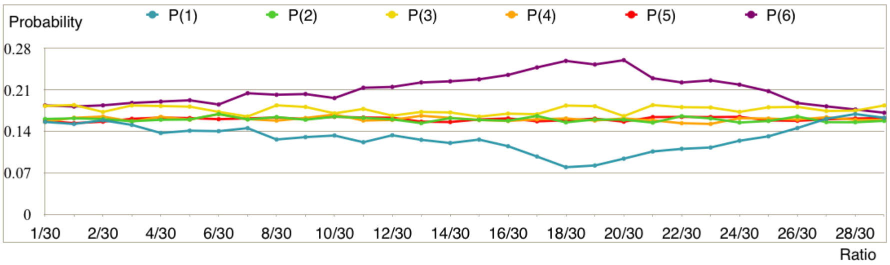

Figure 27: The Probabilities of Dice with Different Volume Ratios

In Section 4.3, the confidence interval for program data are shown to be very narrow due to the large amount of initial variable groups, so all confidence interval is omitted in the table above.

In the graph above, x-coordinate of each point is the vertical displacement between the die's COM and the landing surface (from 0.4 m to 4.2 m). The 6 lines in the graph are the graphs for the possibilities of each one of the six sides of the cube. How each number corresponds to each side of the cube is the same as experiment in Section 5.2.

In graph above, line 6 is higher than other lines for most of the volume ratios except those ratios at the margin of the range. In contrast, line 1 is lower than other lines for all ratios except those at the margin. When line 6 has a positive derivative, line 1 has a negative derivative; when probability for side 6 is decreasing, the probability for side 1 is increasing. The graph is not asymmetrical: from ratio $= \frac{1}{30}$ to ratio $= \frac{19}{30}$, the probability for side 6 is increasing while the probability for side 1 is deceasing; from ratio $= \frac{20}{30}$ to ratio $= \frac{30}{30}$, both line 1 and line 6 increases or decreases back to their original probability at the ratio $= \frac{1}{30}$. At ratio $= \frac{19}{30}$, line 6 reaches its maximum and line 1 reaches its mininum. For other 4 lines, they are approximately held at constant throughout the ratios, and line 3 is a little bit higher than three coincided lines, but compared to line 1 and 6, line 3 look more similar to line 2, 4, and 5.

5.5 Numerical Simulation of Die Toss from Different Heights

Purpose: To observe how the height of tosses influence the results of die tosses through computer simulation.

5.5.1 Data Sampling

In this simulation, the initial heights, h_0 , follow normal distributions with expected values varies from 0.4 m to 4.2 m, and for each height, the variance is set to be 0.1 m. 20 different heights are calculated during this simulation. For other variables: the value of the volume ratio is set to be constant at $\frac{1}{2}$. The three initial Euler angles, $\psi_0, \theta_0, \text{and} \dot{\phi}_0$ are chosen from random within the range of $[0, 2\pi]$. The magnitude of angular velocities, $alpha$ follows a normal distribution with an expected value of 0 rad and a variance of 18 rad/s. Three components of orientational axis, v^1, v^2, and v^3 are random numbers chosen from a range of [0,1], after being chosen, those three components are normalized. Three components of translational velocity, v_{x_0}, v_{y_0}, and v_{z_0}, follow normal distributions with an expected value of 0 and variance of 0.5 m/s. The density ratio is kept constant as $\frac{2714}{7848}$. The coefficients of restitution, χ are kept constant as 0.49 (aluminum) and 0.55 (steel). For each height, 10,000 groups of data are calculated

5.5.2 Results of Simulation

The probabilities of each die's six outcomes are presented in the following table and graph :

Height	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$
0.4	0.1312	0.161975	0.180075	0.157475	0.1606	0.208675
0.6	0.131414	0.161935	0.184141	0.147333	0.165889	0.209288
0.8	0.137	0.1544	0.1813	0.1604	0.1597	0.2072
1	0.1295	0.1571	0.1823	0.1655	0.1598	0.2058
1.2	0.13458	0.160297	0.186928	0.146575	0.164973	0.206648
1.4	0.1306	0.1612	0.1856	0.1581	0.1608	0.2037
1.6	0.1343	0.163	0.1826	0.1552	0.1547	0.2102
1.8	0.126701	0.167107	0.182234	0.148426	0.163452	0.212081
2	0.141054	0.158254	0.189395	0.145125	0.162121	0.20405
2.2	0.1271	0.1675	0.1889	0.1511	0.1585	0.2096
2.4	0.137528	0.165481	0.183777	0.145355	0.162025	0.205835
2.6	0.135981	0.164111	0.187062	0.149792	0.157002	0.206053
2.8	0.130377	0.163633	0.187532	0.143903	0.161599	0.212956
3	0.132405	0.16604	0.191037	0.141144	0.158724	0.210649
3.2	0.135652	0.160057	0.193207	0.14094	0.162803	0.207342
3.4	0.131092	0.164342	0.186138	0.150183	0.162062	0.206183
3.6	0.1348	0.1569	0.1874	0.1602	0.1544	0.2063
3.8	0.1361	0.1561	0.1804	0.1603	0.1577	0.2094
4	0.137	0.1572	0.1784	0.158	0.1583	0.2111
4.2	0.135012	0.1611	0.184753	0.141712	0.161811	0.215613

Table 9: The Probabilities of Dice with Different Heights

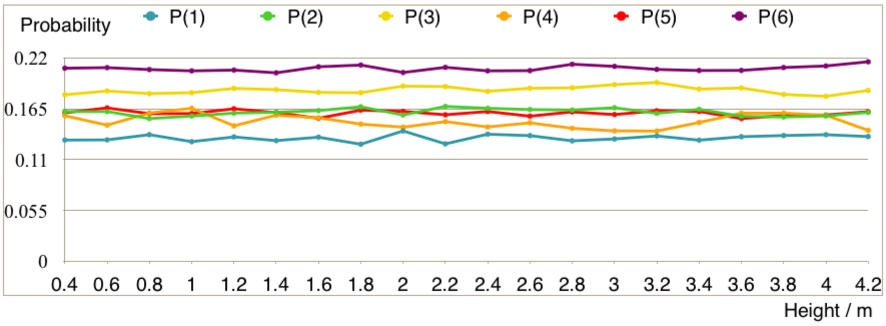

Figure 28: The Probabilities of Dice with Different Heights

In the graph above, x-coordinate of each point is the vertical displacement between the die's COM and the landing surface (from 0.4 m to 4.2 m). The 6 lines in the graph are the graphs for the possibilities of each one of the six sides of the cube. How each number corresponds to each side of the cube is the same as the experiment in Section 5.2.

In graph above, the line for side 6 is above all other lines for all the heights. The line for side 1 is below all other lines for all the heights within the range. The lines for side 2 and side 5 coincide with each other, while the line for side 4 is a little bit higher, and the line for side 3 is a little bit lower. The derivatives for those six lines are all very small. Even though there are some subtle variances, a regular oscillation or linear increase or linear decrease cannot be found. All six lines show the properties of constant functions, and there is no great change in possibilities for each outcome when the height is changed. From the data shown above, for all the heights, the average probability for side 6 is 0.21, and the variance is 0.005. The average probability for side 1 is 0.13, and the variance is 0.005. The average probability for side 2 is 0.16, and the variance is 0.005. The average probability for side 5 is 0.16, and the variance is 0.005. The average probability for side 3 is 0.185, and the variance is 0.005. The average probability for side 4 is 0.145, and the variance is 0.01. All variances are relatively small compare to the value of average probability.

5.6 Numerical Simulation of Die Toss with Different Angular Speed

Purpose: To observe how the magnitude of the angular velocity along orientational axis of the die influence the possibility for each outcome.

5.6.1 Data Sampling

In this simulation, the magnitude of angular velocities, $\dot{\alpha}$, has expected value varies from 1 rad/s to 50 rad/s, and the variance is 0.1 rad/s. 50 different magnitudes are considered. The directions of angular velocity, v^0 follows the same distribution as other simulations: three components of orientational axis are random numbers chosen from a range of [0,1], after being chosen, those three components are normalized. For other variables: the value of the volume ratio is set to be constant at $\frac{1}{2}$. The three initial Euler angles ψ_0, θ_0, and ϕ_0 are chosen randomly within the range of $[0, 2\pi]$. Three components of translational velocity, v_{x_0}, v_{y_0}, and v_{z_0} follow normal distributions with an expected value of 0 and variance of 0.5 m/s. The density ratio is kept constant as $\frac{2714}{7848}$. The coefficients of restitution χ are kept constant as 0.49 (aluminum) and 0.55 (steel). For each magnitude of angular velocity, 10,000 groups of data are calculated.

5.6.2 Results of Simulation

The probabilities of each die's six outcomes are presented in the following table and graph :

A.S.	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$
1	0.0638	0.0835	0.1814	0.152	0.2012	0.3181
2	0.1501	0.0536	0.1639	0.1391	0.2288	0.2645
3	0.1782	0.0774	0.1267	0.2031	0.2549	0.1596
4	0.1482	0.0581	0.1602	0.1531	0.2426	0.2378
5	0.1003	0.1501	0.2023	0.086	0.1707	0.2906
6	0.0396	0.1171	0.1596	0.2645	0.1545	0.2647
7	0.0511	0.1818	0.1763	0.2027	0.0902	0.2979
8	0.1194	0.2326	0.1643	0.1308	0.0599	0.293
9	0.1443	0.2029	0.2739	0.1387	0.0861	0.1541
10	0.145	0.1934	0.2806	0.1293	0.0887	0.163
11	0.1017	0.2285	0.1648	0.1212	0.061	0.3228
12	0.0427	0.173	0.1717	0.2304	0.1	0.2822
13	0.0403	0.1112	0.1626	0.2487	0.1594	0.2778
14	0.0909	0.0676	0.165	0.1247	0.2245	0.3273
15	0.1294	0.0985	0.2907	0.1175	0.1919	0.172
16	0.1716	0.0777	0.1414	0.1811	0.2548	0.1734
17	0.1218	0.0527	0.1751	0.1347	0.239	0.2767
18	0.0604	0.0818	0.1717	0.1853	0.1841	0.3167
19	0.0644	0.1407	0.1584	0.2336	0.1449	0.258
20	0.0843	0.1599	0.1919	0.088	0.167	0.3089
21	0.1377	0.2359	0.1569	0.1492	0.0576	0.2627
22	0.1716	0.2698	0.125	0.1953	0.0902	0.1481
23	0.1363	0.2367	0.1625	0.1431	0.0549	0.266
24	0.0654	0.2009	0.1636	0.1451	0.0777	0.3473
25	0.0684	0.1476	0.1545	0.2387	0.1408	0.25
26	0.0526	0.09	0.1735	0.1769	0.1837	0.3233
27	0.1212	0.0556	0.1604	0.1397	0.2291	0.294
28	0.1675	0.0762	0.1454	0.1759	0.2615	0.1735
29	0.1383	0.1004	0.289	0.1111	0.1875	0.1737
30	0.0926	0.1553	0.2019	0.0877	0.1681	0.2944
31	0.0446	0.1158	0.157	0.2447	0.1623	0.2756
32	0.0425	0.1634	0.1557	0.235	0.1037	0.2997
33	0.0987	0.2226	0.1576	0.1338	0.0645	0.3228
34	0.1584	0.2475	0.1554	0.1747	0.0638	0.1992
35	0.146	0.1919	0.3015	0.1238	0.087	0.1498
36	0.1062	0.2283	0.1548	0.1357	0.0603	0.3147
37	0.0501	0.1774	0.1572	0.2206	0.1028	0.2919
38	0.0414	0.1162	0.1447	0.2787	0.1578	0.2612
39	0.0868	0.1664	0.1979	0.0797	0.1663	0.3029
40	0.129	0.0971	0.2945	0.1127	0.1781	0.1886

Table 10: The Probabilities of Dice with Different Angular Speeds-1

A.S.	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$
41	0.1663	0.0848	0.1424	0.1875	0.2627	0.1563
42	0.1219	0.0578	0.1637	0.1468	0.2282	0.2816
43	0.0588	0.0902	0.1664	0.1655	0.1815	0.3376
44	0.0387	0.1389	0.1282	0.3136	0.1339	0.2467
45	0.0806	0.1492	0.1855	0.0948	0.1684	0.3215
46	0.1181	0.2302	0.1564	0.1427	0.0597	0.2929
47	0.1618	0.2765	0.1412	0.181	0.0802	0.1593
48	0.1367	0.24	0.1639	0.152	0.0591	0.2483
49	0.0917	0.1552	0.1978	0.0892	0.1586	0.3075
50	0.0649	0.1368	0.1491	0.2478	0.1433	0.2581

Table 11: The Probabilities of Dice with Different Angular Speeds-2

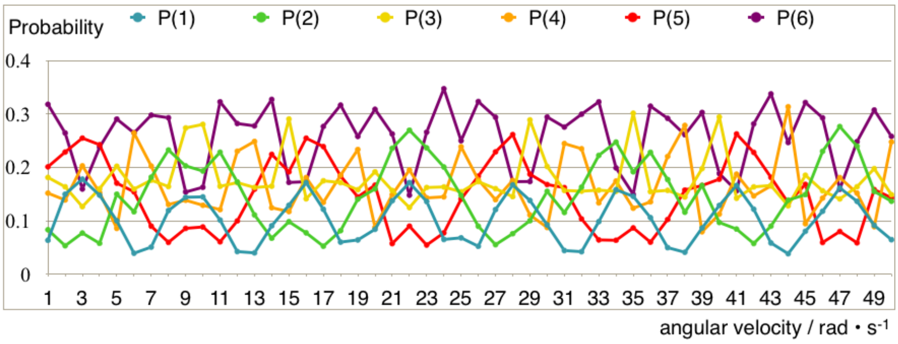

Figure 29: The Probabilities of Dice with Different Angular Speeds-2

In the graph above, x-coordinates are the magnitudes of angular velocities (from 1 rad/s to 50 rad/s). Same as other simulations, the six lines are the graphs of possibilities for each one of the six sides. How each number corresponds to each side of the cube is the same as the experiment in Section 5.2.

All six lines show a pattern similar to wave equations, which the points of the function move up and down in a circular way. Different from other simulations, line 6 is not always above all other lines. The regional maximums of line 6 are always higher than all other lines, but the regional minimums can sometimes become lower than other lines. In the contrast, line 1 is not always the lowest. At line 1's regional maximums, line 1 can be higher than all other lines. Even though the value of possibility for each line moves up and down drastically, the average possibility for each line follows the same pattern as other simulations: the average of line 6 is the highest at around 0.22, and the average of line 1 is the lowest at around 0.12. The average values of line 2, 3, 4, and 5 are similar to each other at a value of

0.18. However, the oscillation patterns of those mixed sides' functions varies a lot from each other. Line 3 and line 4 have long wavelength and high amplitude. In average, for line 3 and line 4, the difference between two crests or two troughs of the function is 11 rad/s. In average, the amplitude of the functions are around 0.17. In the contrast, line 2 and 5 have a more complicated circular function pattern. From 1 rad/s to 4 rad/s, the oscillation is very subtle: the difference between two crests is about 4 rad/s, and the amplitude is around 0.05. From 5 rad/s to 19 rad/s, the oscillation turns more drastic: the difference between two crests is about 6 rad/s. The amplitude of the functions are around 0.11. From 19 rad/s to 23 rad/s, the oscillation turns subtle again with similar wavelength and amplitude as from 1 rad/s to 4 rad/s. From 23 rad/s to 43 rad/s, the oscillation turns drastic again, and from 43 rad/s to 50 rad/s, the oscillation turns subtle. Two forms of oscillations appear alternatively. Line 1 has a much more regular circular pattern with an average wavelength of 7 and an average amplitude of 0.07. Line 6 has two types of regional minimum: one is a regular one like the minimum at $x = 0.9$ rad/s; the other is a relatively higher minimum between two very close maximums, like the one at $x = 19$ rad/s.

5.7 Numerical Simulation of Die Toss with Different Translational Speed

Purpose: To observe how initial translational velocity influences the probability for each side of a die.

5.7.1 Data Sampling

In this simulation, the translational velocity in y and z components is ignored. In reality, all dice are tossed up, so there must be a moment when the z-component speed is zero. That moment is considered in this simulation, so the z-component velocity is ignored. When a die moves in air, it follows a trajectory in a plane, so it's unnecessary to consider two components of velocity if the x direction for each toss is automatically set to be same as the die's initial direction. The x-component translational speeds follow normal distribution with expected values varies from 0.5 m/s to 15 m/s. The speeds that are products of 0.5 m/s are involved in calculations. Each speed has the variance of 0.05 m/s. For other variables, the volume ratio is set as $\frac{1}{2}$. The three initial Euler angles, $\psi, \theta, and \phi$ are chosen from random within the range of $[0, 2\pi]$. The magnitude of angular velocities, $\dot{\alpha}$ follows a normal distribution with an expected value of 0 rad and a variance of 18 rad/s. Three components of orientational axis, v^1, v^2, and v^3 are random numbers chosen from a range of [0,1], after being chosen, those three components are normalized. The density ratio is kept constant as $\frac{2714}{7848}$. The coefficients of restitution, χ are kept constant as 0.49 (aluminum) and 0.55 (steel). The height, h_0 follows a normal distribution with an expected value of 1 m and a variance of 0.2 m. For each translational speed, 10,000 groups of initial variables are calculated.

5.7.2 Results of Simulation

The probabilities of each die's six outcomes are presented in the following table and graph:

x-velocity	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$
0.5	0.1265	0.1573	0.18125	0.16115	0.16095	0.21285
1	0.1321	0.158	0.1816	0.1594	0.1574	0.2115
1.5	0.1329	0.1575	0.1828	0.163	0.1568	0.207
2	0.134	0.1554	0.1797	0.1599	0.1622	0.2088
2.5	0.1222	0.1597	0.175	0.1607	0.1623	0.2201
3	0.1254	0.1618	0.187	0.1609	0.1529	0.212
3.5	0.1278	0.1616	0.1783	0.1586	0.1633	0.2104
4	0.1287	0.158	0.1829	0.1582	0.165	0.2072
4.5	0.1324	0.1608	0.1744	0.1636	0.1588	0.21
5	0.122	0.1653	0.1805	0.1567	0.1653	0.2102
5.5	0.1249	0.1592	0.1775	0.164	0.1624	0.212
6	0.133	0.1581	0.1817	0.1561	0.1607	0.2104
6.5	0.1298	0.1702	0.1806	0.1557	0.1572	0.2065
7	0.1339	0.1645	0.1778	0.157	0.1535	0.2133
7.5	0.1318	0.1594	0.1764	0.1597	0.1587	0.214
8	0.1319	0.1598	0.1781	0.1615	0.1551	0.2136
8.5	0.1256	0.1623	0.1767	0.1628	0.1595	0.2131
9	0.1234	0.1629	0.172	0.1644	0.164	0.2133
9.5	0.1204	0.1699	0.179	0.1566	0.1644	0.2097
10	0.1253	0.1602	0.1762	0.1609	0.1653	0.2121
10.5	0.1258	0.1607	0.1819	0.166	0.1569	0.2087
11	0.1295	0.163	0.1744	0.1575	0.1629	0.2127
11.5	0.1267	0.1672	0.1723	0.1604	0.1596	0.2138
12	0.128	0.1618	0.1853	0.1578	0.1608	0.2063
12.5	0.1281	0.1583	0.1795	0.171	0.1619	0.2012
13	0.1281	0.1551	0.1788	0.1639	0.1563	0.2178
13.5	0.1282	0.1629	0.1786	0.1616	0.1579	0.2108
14	0.1275	0.1567	0.1841	0.164	0.1616	0.2061
14.5	0.1307	0.1651	0.1766	0.1609	0.1557	0.211
15	0.1332	0.1521	0.1755	0.1603	0.1626	0.2163

Table 12: The Probabilities of Dice with Different x-velocities

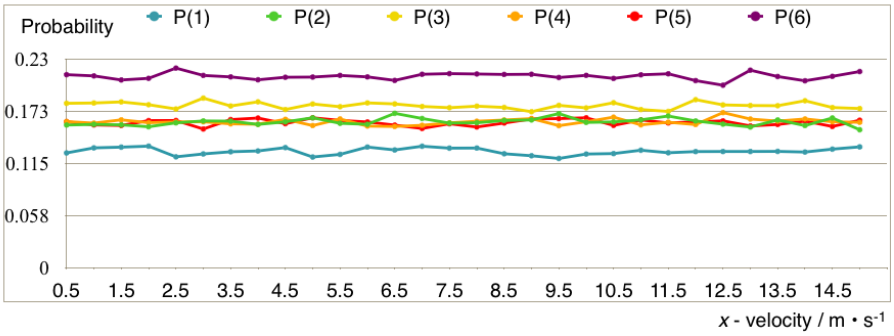

Figure 30: The Probabilities of Dice with Different x-velocities

In the graph above, x-coordinates are the x-component speeds (from 0.5 m/s to 15 m/s). Same as other simulations, the six lines are the graphs of possibilities for each one of the six sides. How each number corresponds to each side of the cube is the same as the experiment in Section 5.2.

The pattern shown in the graph is very similar to the graph for initial heights. Line 6 is above all other lines, while line 1 the lowest for every initial speed in the range. Line 2, 3, 4, and 5 are all in the space between line 6 and line 1. Line 2, 4, 5 coincide with each other a lot while line 3 is a little bit higher, but in compared to other line 6 and line 1, line 3 is much closer to other three lines. There are some subtle rises and falls on the function, but these small changes are super random, and there is no obvious patterns. All six lines are very close to constant functions, and the average value of each side's probability calculated from this graph is very similar to the results calculated from the various heights simulation.

5.8 Numerical Simulation of Die Toss from Different Coefficients of Restitution

Purpose: To observe how coefficient of restitution with the ground influence the probability of each outcome of a die toss.

5.9 Data Sampling

Instead of considering different material's different coefficient, this simulation assumes coefficient of restitution, χ, is same for all eight corners in order to find coefficient of restitutions direct influence on the probability for each outcome. The value of χ varies from 0.04 to 0.96. In total, 24 coefficients of restitution is discussed in the simulation, all the values are products of 0.04. For other variables: the volume ratio is kept constant at $\frac{1}{2}$ the three initial Euler angles, $\psi, \theta,$ andϕ

are chosen from random within the range of $[0, 2\pi]$. The magnitude of angular velocities, $\dot{\alpha}$ follows a normal distribution with an expected value of 0 rad and a variance of 18 rad/s. Three components of orientational axis, v^1, v^2, and v^3 are random numbers chosen from a range of [0,1], after being chosen, those three components are normalized. Three components of translational velocity, v_{x_0}, v_{y_0}, and v_{z_0} follow normal distributions with an expected value of 0 and variance of 0.5 m/s. The density ratio is kept constant as $\frac{2714}{7848}$. The height, h_0 follows a normal distribution with an expected value of 1 m and a variance of 0.2 m. For each coefficient of restitution, 10,000 groups of initial variables are calculated.

5.9.1 Results of Simulation

The probabilities of each die's six outcomes are presented in the following table and graph:

CoR	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$
0.04	0.1109	0.1587	0.1765	0.1658	0.1592	0.2289
0.08	0.1111	0.1637	0.1741	0.1685	0.1638	0.2188
0.12	0.1182	0.1588	0.1727	0.1572	0.1644	0.2287
0.16	0.1141	0.1583	0.1864	0.1604	0.1572	0.2236
0.2	0.1199	0.1595	0.1823	0.1616	0.1565	0.2202
0.24	0.1282	0.1577	0.173	0.1644	0.1657	0.211
0.28	0.1192	0.158	0.1747	0.167	0.1631	0.218
0.32	0.1205	0.1618	0.1776	0.162	0.1599	0.2182
0.36	0.1281	0.1651	0.1781	0.1577	0.1615	0.2095
0.4	0.1245	0.1597	0.1771	0.1651	0.1652	0.2084
0.44	0.1287	0.1652	0.1801	0.1619	0.1558	0.2083
0.48	0.137	0.16	0.1799	0.1538	0.1657	0.2036
0.52	0.1319	0.1647	0.1794	0.1633	0.1533	0.2074
0.56	0.1335	0.1621	0.1769	0.16	0.1558	0.2117
0.6	0.1328	0.1645	0.1748	0.1626	0.1592	0.2061
0.64	0.1444	0.1624	0.1796	0.155	0.1611	0.1975
0.68	0.1371	0.1615	0.1755	0.1618	0.1563	0.2078
0.72	0.1417	0.1615	0.1736	0.1632	0.1607	0.1993
0.76	0.1352	0.1684	0.1804	0.1622	0.1511	0.2027
0.8	0.1395	0.1579	0.1789	0.1597	0.1655	0.1985
0.84	0.1337	0.1642	0.1869	0.1609	0.158	0.1963
0.88	0.1415	0.1624	0.1772	0.1568	0.1658	0.1963
0.92	0.1463	0.1634	0.1702	0.1625	0.1661	0.1915
0.96	0.149	0.1595	0.1772	0.164	0.1657	0.1846

Table 13: The Probabilities of Dice with Different Coefficients of Restitution

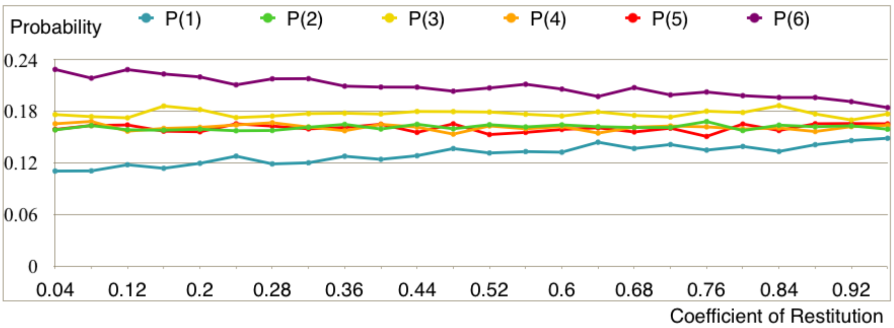

Figure 31: The Probabilities of Dice with Different Coefficients of Restitution

In the graph above, x-coordinates are coefficients of restitution (from 0.04 to 0.96). Same as other simulations, the six lines are the graphs of possibilities for each one of the six sides. How each number corresponds to each side of the cube is the same as the experiment in Section 5.2.

In the graph, line 6 is always higher than other lines, and line 1 is always lower than other lines. Line 2, 4, and 5 coincide with each other a lot, while line 3 is a little bit higher. All lines show a tendency to approach probability to approach 0.16 as coefficient of restitution increases. As coefficient changes from 0.04 to 0.96, line 6 linearly decreases to 0.18 from 0.23, and line 1 linearly increases to 0.15 from 0.11. Other lines for mixed sides don't have an obvious positive or negative derivative, and the probability for these mixed sides is always around 0.16.

5.10 Numerical Simulation of Die Toss with Different Initial Orientation

Purpose: To observe how different initial orientations influence the possibilities for each outcome of a die toss.

5.10.1 Data Sampling

During this stimulation, six special set of initial Euler angles, ψ, θ, ϕ, are chosen. The orientation of die shown in the figure 8 (The Density Distribution Example) is assumed to be $(0, 0, 0)$. Based on that, the initial orientations are:

$(0, 0, 0)$: the orientation when stainless steel side is at the top;

$(0, \frac{\pi}{2}, 0)$: the orientation when aluminum alloy side is at the top;

$(0, \frac{3\pi}{4}, 0), (0, \frac{\pi}{4}, 0), (0, \frac{\pi}{4}, \frac{\pi}{4}), (0, \frac{\pi}{4}, \frac{3\pi}{4})$: the orientation when a mixed side is at the top.

For other variables: The volume ratio is kept constant at $\frac{1}{2}$. The magnitude of angular velocities, $\dot{\alpha}$ follows a normal distribution with an expected value of 0 rad and a variance of 18 rad/s. Three components of orientational axis, v^1, v^2, and v^3 are random numbers chosen from a range of [0,1], after being chosen, those three components are normalized. Three components of translational velocity, $v_{x_0}, v_{y_0}, and v_{z_0}$ follow normal distributions with an expected value of 0 and variance of 0.5 m/s. The density ratio is kept constant as $\frac{2714}{7848}$. The coefficients of restitution, χ are kept constant as 0.49 (aluminum) and 0.55 (steel). The height, h_0 follows a normal distribution with an expected value of 1 m and a variance of 0.2 m. For each orientation, 20,000 groups of initial variables are calculated.

5.10.2 Results of Simulation

The probabilities of each dies six outcomes are presented in the following table

top face	$P(1)$	$P(2)$	$P(3)$	$P(4)$	$P(5)$	$P(6)$
1	0.1030	0.1787	0.1831	0.1576	0.1328	0.2449
2	0.0929	0.1697	0.2005	0.1574	0.1522	0.2275
3	0.1120	0.1559	0.2106	0.1575	0.1555	0.2087
4	0.1184	0.1756	0.1717	0.1687	0.1450	0.2207
5	0.1228	0.1659	0.1937	0.1573	0.1533	0.2072
6	0.1225	0.1772	0.1790	0.1662	0.1568	0.1984

Table 14: The Probabilities of Dice with Different Coefficients

In the table above, number 1-6 on the left of the table represents the initial orientation: for example, 1 means the side with number 1 is at top, which is the side of stainless steel. No matter which side is at top at the beginning, the probability of side 6 is always the higher, and the probability for side 1 is the lowest. Two initial orientations have some subtle influence on the possibilities. When initial orientation is side 1 at top, the probability for side 6 as final outcome is greater than the scenarios when initial orientation is side 6 at top; when initial orientation is side 6 at top, the probability for side 1 as final outcome is greater than the scenarios when initial orientation is side 1 at top. However, when mixed side is at top, there are no patterns of changes that can be found.

6 Conclusion

6.1 Correlation Between Probability and Volume Ratio

Die 1	Simulation	Experiment
$P(1)$	0.128 ± 0.0065	0.086 ± 0.0246
$P(2)$	0.1627 ± 0.0072	0.166 ± 0.0326
$P(3)$	0.18 ± 0.0075	0.148 ± 0.0311
$P(4)$	0.1632 ± 0.0072	0.17 ± 0.03299
$P(5)$	0.1562 ± 0.0071	0.162 ± 0.0323
$P(6)$	0.2099 ± 0.0080	0.27 ± 0.0389

Table 15: Comparison between Simulation and Experiment for Volume Ratio-Die 1

Die 2	Simulation	Experiment
$P(1)$	0.1341 ± 0.0067	0.114 ± 0.0279
$P(2)$	0.1604 ± 0.0072	0.17 ± 0.0329
$P(3)$	0.1771 ± 0.0075	0.16 ± 0.0321
$P(4)$	0.1603 ± 0.0072	0.182 ± 0.0338
$P(5)$	0.1628 ± 0.0072	0.166 ± 0.0326
$P(6)$	0.2053 ± 0.0079	0.208 ± 0.0356

Table 16: Comparison between Simulation and Experiment for Volume Ratio-Die 2

Die 3	Simulation	Experiment
$P(1)$	0.2066 ± 0.0079	0.246 ± 0.0378
$P(2)$	0.1990 ± 0.0078	0.158 ± 0.0320
$P(3)$	0.1876 ± 0.0077	0.156 ± 0.0318
$P(4)$	0.1479 ± 0.0070	0.164 ± 0.0325
$P(5)$	0.1562 ± 0.0071	0.172 ± 0.0331
$P(6)$	0.1027 ± 0.0059	0.108 ± 0.0272

Table 17: Comparison between Simulation and Experiment for Volume Ratio-Die 3

From the three tables shown above, there are some obvious differences between the data from simulation and data from experiments. For Die-1, which is the die with half stainless steel and half aluminum alloy, the probabilities from two sources for mixed sides are very similar, but the probabilities for two full metal sides varies a lot.The reason may due to the limited number of experiments, or the die is not thrown randomly during the experiments. However, most of the data matches good, and the correctness of the program can be proved.

As the volume ratio between aluminum alloy and stainless steel increases, the probability for the aluminum alloy side increases, and the probability for the stainless steel side decreases. When the volume ratio reaches a critical value, like $ratio = 19/30$ in the simulation of this research, probability for aluminum alloy side reaches its maximum, the probability for the stainless steel side reaches its minimum. After this critical value, aluminum alloy side's probability begins to decrease and the stainless steel side's probability begins to increase. The probabilities for mixed sides are not influenced by volume ratio. The possible theory behind this phenomenon is, as the volume ratio increases, the position of COM shifts. Before reaching the critical value, the COM is shifting to the heavier side as volume ratio increases, so the lighter side has larger and larger probability to appear on top; when the volume ratio passes that critical value, the COM begins to be shifted back to the middle, so the difference between two full metal sides' probability begins to shrink and finally they coincide again when the dice turn homogeneous when ratio $= \frac{30}{30}$. For four mixed sides, they are symmetric to each other, so from a geometric perspective, those sides' probabilities is very similar to each others. Because the sum of probability for stainless steel and probability for aluminum alloy doesn't change a lot, so the probability for each one of the four mixed side doesn't change much.

6.2 Correlation Between Probability and Height

1.0 m	Simulation	Experiment
$P(1)$	0.1295 $\pm$ 0.0066	0.086 $\pm$ 0.025
$P(2)$	0.1571$\pm$ 0.0071	0.166 $\pm$ 0.033
$P(3)$	0.1823$\pm$ 0.0076	0.148 $\pm$ 0.031
$P(4)$	0.1655$\pm$ 0.0073	0.17 $\pm$ 0.033
$P(5)$	0.1598$\pm$ 0.0072	0.162 $\pm$ 0.032
$P(6)$	0.2058$\pm$ 0.0079	0.27 $\pm$ 0.039

Table 18: Comparison between Simulation and Experiment for Height 1.0 m

0.8 m	Simulation	Experiment
$P(1)$	0.1341 ± 0.0067	0.208 ± 0.0356
$P(2)$	0.1604 ± 0.0072	0.17 ± 0.0329
$P(3)$	0.1771 ± 0.0075	0.16 ± 0.0321
$P(4)$	0.1603 ± 0.0072	0.182 ± 0.0338
$P(5)$	0.1628 ± 0.0072	0.166 ± 0.0326
$P(6)$	0.2053 ± 0.0079	0.416 ± 0.092

Table 19: Comparison between Simulation and Experiment for Height 0.8 m

0.4 m	Simulation	Experiment
$P(1)$	0.1312 ± 0.0066	0.108 ± 0.0272
$P(2)$	0.162 ± 0.0072	0.158 ± 0.0320
$P(3)$	0.1801 ± 0.0075	0.156 ± 0.0318
$P(4)$	0.1574 ± 0.0071	0.164 ± 0.0325
$P(5)$	0.1606 ± 0.0072	0.172 ± 0.0331
$P(6)$	0.2087 ± 0.0080	0.246 ± 0.0378

Table 20: Comparison between Simulation and Experiment for Height 0.4 m

The probability for each side to appear and the height at which the die is released from are independent. The possible theory behind may be this: greater height increases the time for a die's motion in air, but the increase of the height doesn't decline the randomness for die's motion in air: angular velocities are randomly chosen from a normal distribution with a large variance, so the value of angular velocities are very random in this simulation. A little change of angular velocities may bring much more randomness that overweight the decline of randomness achieved by a constant height. To see the height's influence on die's probability, the initial magnitude of angular velocity may need to be set as zero first.

6.3 Correlation Between Probability and Angular Speed

There is no experiments due to the difficulty to control a small cube's initial angular velocity. However, based on two comparisons between experiments and simulations in last two sub-sections, the simulating program is proved to be trustworthy.

Angular speed changes the possibility for each side drastically. As the magnitude of angular velocity increases, the possibility for each side oscillates up and down: Two full metal sides' and two of the mixed sides' functions go up and down like wave equations, while the other two mixed sides oscillates in a special way, which is carefully described in Section 5.6. The possible reason behind that may be about the angular velocity largely decides which point is the point of contact A and which side will project the largest shadow before each bouncing. Even though the time in the air may also decides which vertex to contact with ground, the height has a relatively small variance in this simulation, and there are 10,000 groups of calculation, so the determination of contact point A is dominated by angular velocity. Because the angular velocity moves in a circular way, so the function shows a circular pattern as well.

6.4 Correlation Between Probability and Translational Speed

The probability for each side to appear is independent of the initial translational speed. As the initial speed increases from 0.5 m/s to 15 m/s, there is no big difference among any two groups of data. All function are approximately constant function throughout the range. The possible reason behind this phenomenon is: the translational speed and the angular speed are independent during a object's motion in air. The air resistance is neglected, so the translational speed will not produce a resistance force and add a torque on the cube. The translational speed discussed in this simulation is the speed in x-direction, so this speed keeps at constant throughout the motion because collision doesn't influence the speed in x-directoin.

6.5 Correlation Between Probability and Coefficient of Restitution

As the coefficient of restitution increases, the difference between the probability for aluminum alloy side and the stainless steel side decreases. As the coefficient of restitution approaches to 1, two metal side's probability approaches to 0.17, and the unfair die behaves more and more like a fair die when the coefficient of restitution is greater. The possible reason is, greater coefficient of restitution means less energy dissipated during the collision, which implies a minor force exerted by the surface when collided. Because the value of momentum doesn't change its value much, a minor force implies a longer time. When the coefficient of restitution is small, it implies a longer time of contact with the surface. When the cube is contacts with the ground, the gravitational force has more time to pull down the cube towards the heavier side, so the aluminum alloy side has greater possibility to appear when coefficient of restitution is small

6.6 Correlation Between Probability and Orientation

The initial orientation influences the probabilities in a minor way. Comparing the volume ratio, coefficient of restitution, and angular velocities, the change of possibilities due to the different initial orientation is small. An initial orientation can determine the coordinate of contact point and the side projecting the largest shadow, but the effect of initial orientation is often over-weighed by angular velocity because a small change in angular velocity may lead to a big change in orientation.

6.7 Error Analysis

6.7.1 In Experiments

The errors for an experiment may occur when we tossed the dice non-randomly. For example, the initial orientation should be random, but some times we set the initial orientation upright and fixed.

The errors may also occur if we misestimate the range of the initial values.

6.7.2 In Simulations

The algorithm ignores the friction and air resistance.

The experiment uses a random number generator, which could not generate truly random numbers.

Again, like in experiments, the errors may occur if we misestimate the range of the initial values.

References

[1] Diaconis, P., Keller, J.B.: Fair dice. Amer. Math. Monthly 96, 337339 (1989) 7, 9

[2] Liber de Ludo Aleac (The Book of Games of Chance) Gerolamo Cardano 1663

[3] G. Galilei, Sopra le Scoparte dei Dadi [Analysis of Dice Games] (1612) [reprinted in G. Galileo, "Sopra le scoperte dei dadi," in Opere: A cura di Ferdinando Flora (Ricciardi, Milan, 1952)]

[4] J. B. Keller, "The probability of heads," The American Mathematical Monthly, Vol. 93, No. 3. (Mar., 1986), pp. 191-197

[5] The Three-Dimensional Dynamics of a Dice Throw: M. Kapitaniak,1,2 J. Strzalko,1 J. Grabski,1 and T. Kapitaniak1 2012

[6] Shi Yongfang, Zhao Gengsheng, Zhao Gengsheng, Qu Hongbin, Lei Guilin, "Research about the Probability of a Special Kind of Unfair Dice (Yi Lei Fei Jun Yun Tou Zi De Gai Lv Wen Ti)," 2009.

[7] J. Strzalko, J. Grabski, A. Stefanski, and T. Kapitaniak, "Can the dice be fair by dynamics?," Int. J. Bifurcation Chaos 20, 1175 (2010).

[8] J. Strzalko, J. Grabski, P. Perlikowski, A. Stefanski, and T. Kapitaniak, Dynamics of Gambling: Origins of Randomness in Mechanical Systems, Lecture Notes in Physics Vol. 792 (Springer, Berlin, 2010).

[9] ME 6590 Multibody Dynamics Moments and Products of Inertia and the Inertia Matrix http://homepages.wmich.edu/ kamman/Me659InertiaMatrix.pdf

[10] Tijms H.: Understanding Probability. Cambridge University Press, Cambridge (2007) 1, 13

[11] Maclin, O.H., Dixon, M.R.: A computerized simulation for investigating gambling behavior during roulette play. Behav. Res. Methods, Instrum. Comput. 36 (1), 96100 (2004) 2

[12] Arnold, V.I.: Mathematical Methods of Classical Mechanics. Springer, New York (1989)

[13] Wittaker, E.T.: Analytical Dynamics of Particles and Rigid Bodies. Cambridge University Press, Cambridge (1937) 23, 31

[14] Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors James Diebel Stanford University (2006)

[15] Altmann, S.L.: Rotations, Quaternions, and Double Groups. Oxford University Press, Oxford (1986) 34

[16] Goldstein, H.: Classical Mechanics. Addison-Wesley, Reading (1950) 23, 31, 37

[17] Neimark, J.I., Fufaev, N.A.: Dynamics of nonholonomic systems Translations of Mathematical Monographs, vol. 33. American Mathematical Society, Providence, Rhode Island, 518 (1972) 65, 70

[18] United Kingdom Office of Public Sector Information: Definition as gaming. http://www.opsi.gov.uk/acts1990/Ukpga-19900026-en1.htm (2009). Cited 10 Apr 2009 1

[19] P. Diaconis, S. Holmes, and R. Montgomery, "Dynamical bias in the coin toss," SIAM Rev. 49, 211 (2007).

[20] Weisstein, Eric W. "Euler Angles." From *MathWorld*–A Wolfram Web Resource. http://mathworld.wolfram.com/EulerAngles.html

A Programs

The language is Mathematica

```
(* Constants *)
g = 10;
orientationType = {3, 1, 3};

(* Accuracy parameters *)
errorBoundForTrueHeightBaoli = 0.0000000000000000001;
minimumUpwardSpeedToStop=0.1;

(* Die information *)
{rho1,rho2}={1500  10050};
{l1,l2}={0.015,0.015};
compo=(l1^2*rho1/2+(l1+l2/2)*rho2*l2)/
(rho1*l1+rho2*l2);
(*Print[size]*)
density[x_,y_,z_]:=If[x<l1-compo,rho1,rho2];
size={{compo,l1+l2-compo},{compo,l1+l2-compo},
{compo,l1+l2-compo}};
m=Integrate[density[x,y,z],Flatten[{x,size[[1]]}],
Flatten[{y,size[[2]]}],Flatten[{z,size[[3]]}]];
CoM={0,0,0}(* it must be *);
(*sixVectors={{1,0,0},{0,1,0},{0,0,1},{-1,0,0},
{0,-1,0},{0,0,-1}};*)
sixVectors={{-1,0,0},{0,-1,0},{0,0,-1},{0,0,1},
{0,1,0},{1,0,0}};
eightVectors={{-l1, -0.0155, -0.015}, {-l1, 0.015,
0.015}, {-l1, 0.015, -0.015},{-l1, -0.015, 0.015},
{l2, -0.015, -0.015}, {l2, -0.015, 0.015}, {l2,
0.015, -0.015}, {l2, 0.015, 0.015}};
vertices=eightVectors;
CoR={0.49,0.49,0.49,0.49,0.55,0.55,0.55,0.55};

(* Conversion between Euler rate and angular velocity *)
eulerRateToAngularVelocity[eulerRates_,eulerAngles_]
:=Flatten[{{Sin[eulerAngles[[2]]] Sin[eulerAngles[[3]]
],Cos[eulerAngles[[3]]],0},
{Sin[eulerAngles[[2]]] Cos[eulerAngles[[3]]],
-Sin[eulerAngles[[3]]],0},{Cos[eulerAngles[[2]]],0,1}}
```

```
.{{eulerRates[[1]]},{eulerRates[[2]]},
{eulerRates[[3]]}}];
angularVelocityToEulerRate[angularVelocity_,
eulerAngles_]:=
Flatten[Inverse[{{Sin[eulerAngles[[2]]]
Sin[eulerAngles[[3]]],Cos[eulerAngles[[3]]],0},
{Sin[eulerAngles[[2]]]Cos[eulerAngles[[3]]],
-Sin[eulerAngles[[3]]],0},{Cos[eulerAngles[[2]]]
,0,1}}]
.{{angularVelocity[[1]]},{angularVelocity[[2]]},
{angularVelocity[[3]]}}];

(* Conversion between Euler rate and angular velocity *)
eulerRateToAngularVelocity[eulerRates_,eulerAngles_]
:=Flatten[{{Sin[eulerAngles[[2]]]
Sin[eulerAngles[[3]]],Cos[eulerAngles[[3]]],0},
{Sin[eulerAngles[[2]]]Cos[eulerAngles[[3]]],
-Sin[eulerAngles[[3]]],0},{Cos[eulerAngles[[2]]],0,1}}
.{{eulerRates[[1]]},{eulerRates[[2]]},
{eulerRates[[3]]}}];
angularVelocityToEulerRate[angularVelocity_,
eulerAngles_]:=Flatten[Inverse[{{Sin[eulerAngles[[2]]
]Sin[eulerAngles[[3]]],Cos[eulerAngles[[3]]],0},
{Sin[eulerAngles[[2]]]Cos[eulerAngles[[3]]],
-Sin[eulerAngles[[3]]],0},{Cos[eulerAngles[[2]]]
,0,1}}].{{angularVelocity[[1]]},{angularVelocity[[2]]
},{angularVelocity[[3]]}}];

(* Moment of inertia *)
MoI[axis_]:=NIntegrate(* or use Integrate instead
*)[(distance[{x,y,z},axis,CoM])^2*density[x,y,z],
{x,size[[1]][[1]],size[[1]][[2]]},{y,size[[2]][[1]],
size[[2]][[2]]},{z,size[[3]][[1]],size[[3]][[2]]}];

(* Find Top Face *)
getTopFace[sixFinalVectors_] := Module[
{sixAngles, minimumAngle, result},
sixAngles = Table[VectorAngle[sixFinalVectors[[i]],
{0, 0, 1}], {i, 1, 6}];
minimumAngle = Min[sixAngles];
```

```
For[i = 1, i <= 6, i++, If[sixAngles[[i]] ==
minimumAngle, result = i]];
result
];

(* Find Contact Point *)
contactPoint[eightFinalVectors_] := Module[
{eightAngles, minimumAngle, result},
eightAngles = Table[VectorAngle[eightFinalVectors[[i]], {0, 0, -1}], {i,
minimumAngle = Min[eightAngles];
For[i = 1, i <= 8, i++, If[eightAngles[[i]] == minimumAngle, result = i]
result
]

(* Vector rotates about a single axis *)
rotateAboutAxis[vector_, axis_, theta_] := Module[
{R, normalizedAxis, vectorMatrix, finalVectorMatrix,
finalVector},
normalizedAxis = Normalize[axis];
R = {{Cos[theta] + normalizedAxis[[1]]^2*(1 -
Cos[theta]), normalizedAxis[[1]]*normalizedAxis[[2]]
*(1 - Cos[theta]) - normalizedAxis[[3]]
*Sin[theta], normalizedAxis[[1]]*normalizedAxis[[3]]
*(1 - Cos[theta]) + normalizedAxis[[2]]*Sin[theta]},
{normalizedAxis[[2]]*normalizedAxis[[1]]*(1 -
Cos[theta]) + normalizedAxis[[3]]*Sin[theta],
Cos[theta] +
normalizedAxis[[2]]^2*(1 - Cos[theta]),
normalizedAxis[[2]]*normalizedAxis[[3]]*(1 -
Cos[theta]) - normalizedAxis[[1]]*Sin[theta]},
{normalizedAxis[[3]]*normalizedAxis[[1]]*(1 -
Cos[theta]) - normalizedAxis[[2]]*Sin[theta],
normalizedAxis[[3]]
*normalizedAxis[[2]]*(1 - Cos[theta]) +
normalizedAxis[[1]]*Sin[theta], Cos[theta] +
normalizedAxis[[3]]^2*(1 - Cos[theta])}};
R = N[R];
vectorMatrix = vectorToMatrix[vector];
finalVectorMatrix = R.vectorMatrix;
finalVector = matrixToVector[finalVectorMatrix]
```

```
    ]

(* Euler Angles Conversion *)
xi[angle_] := {{1, 0, 0}, {0, Cos[angle],
-Sin[angle]}, {0, Sin[angle], Cos[angle]}};
eta[angle_] := {{Cos[angle], 0, Sin[angle]},
{0, 1, 0}, {-Sin[angle], 0, Cos[angle]}};
zeta[angle_] := {{Cos[angle], -Sin[angle], 0},
{Sin[angle], Cos[angle], 0}, {0, 0, 1}};
XiEtaZeta[angle_, type_] := {xi[angle], eta[angle],
zeta[angle]}[[type]];
orientationMatrix[orientation_] :=
XiEtaZeta[orientation[[1]],
orientationType[[1]]].XiEtaZeta[orientation[[2]],
orientationType[[2]]]
.XiEtaZeta[orientation[[3]], orientationType[[3]]];
vectorAfterOrientation[vector_, orientation_] :=
matrixToVector[
Inverse[orientationMatrix[orientation]].
vectorToMatrix[vector]];

(* Free Fall *)
freeFallTimeAndAngleAndFinalVelocity[velocity_,
height_,angularSpeed_] := Module[
{finalDownwardSpeed,finalVelocity,angle, time},
finalDownwardSpeed = Sqrt[velocity[[3]]^2 +
2*g*height];
finalVelocity = {velocity[[1]], velocity[[2]],
-1*finalDownwardSpeed};
time = (velocity[[3]] - finalVelocity[[3]])/g;
angle = time*angularSpeed;
{time, angle,finalVelocity}
];

(* Free fall calculation *)
freeFallTimeAndAngleAndFinalVelocity[velocity_,
height_,angularSpeed_] := Module[
{finalDownwardSpeed,finalVelocity,angle, time},
finalDownwardSpeed = Sqrt[velocity[[3]]^2 +
2*g*height];
```

```
finalVelocity = {velocity[[1]], velocity[[2]],
-1*finalDownwardSpeed};
time = (velocity[[3]] - finalVelocity[[3]])/g;
angle = time*angularSpeed;
{time, angle,finalVelocity}
];

(* Free fall *)
freeFall[orientation_,sixVectors_, eightVectors_,
initialVelocity_, angularVelocity_,height_] :=
Module[{exactHeight, timeAndAngleAndFinalVelocity,
orientationChange,orientationAfter,
initialAngularVelocity,angularSpeed,
axis, sixVectorsAfter, eightVectorsAfter},
angularSpeed=Norm[angularVelocity];
axis=Normalize[initialAngularVelocity];
exactHeight = trueHeightBaoli[orientation,
sixVectors,eightVectors, initialVelocity,
angularVelocity, height,
errorBoundForTrueHeightBaoli];
timeAndAngleAndFinalVelocity = freeFallTimeAndAngleAndFinalVelocity
[initialVelocity,
exactHeight,angularSpeed];
orientationChange=calcEulerAnglesFromAngularVelocity
[angularVelocity];
orientationAfter=orientation+orientationChange;
sixVectorsAfter =
Table[vectorAfterOrientation[sixVectors[[i]],
orientationChange], {i, 1, 6}];
eightVectorsAfter =
Table[vectorAfterOrientation[eightVectors[[i]],
orientationChange], {i, 1, 8}];
{orientationAfter,sixVectorsAfter, eightVectorsAfter,
timeAndAngleAndFinalVelocity[[3]],angularVelocity,
CoMHeight[eightVectorsAfter],
angularVelocityToEulerRate[angularVelocity,
orientationAfter],
timeAndAngleAndFinalVelocity[[1]]}
]
```

```
(* From free fall to bounce *)
calcFoure[angle_, axis_]:={Cos[angle/2],
Normalize[axis][[1]]*Sin[angle/2],
Normalize[axis][[2]]*Sin[angle/2],
Normalize[axis][[3]]*Sin[angle/2]};
calcEulerAnglesFromAngularVelocity[angularVelocity_]
:=Module[
{e0,e1,e2,e3,s1,s2,s3,s4,s5,s6,s7,s8,
ph,the,ps,al,su,sm,nm,re},
{e0,e1,e2,e3}=calcFoure[Norm[angularVelocity],
Normalize[angularVelocity]];
s1=Re[{ArcSin[Sqrt[-1+e0^2+e1^2+e2^2]/
Sqrt[-1+e1^2+e2^2]]+ArcSin[e2/Sqrt[e1^2+e2^2]],
2 ArcSin[Sqrt[e1^2+e2^2]],
ArcSin[Sqrt[-1+e0^2+e1^2+e2^2]/Sqrt[-1+e1^2+e2^2]]
- ArcSin[e2/Sqrt[e1^2+e2^2]]}];
s2=Re[{- ArcSin[Sqrt[-1+e0^2+e1^2+e2^2]/
Sqrt[-1+e1^2+e2^2]]+ArcSin[e2/Sqrt[e1^2+e2^2]],
2ArcSin[Sqrt[e1^2+e2^2]],-ArcSin[
Sqrt[-1+e0^2+e1^2+e2^2]/Sqrt[-1+e1^2+e2^2]]
-ArcSin[e2/Sqrt[e1^2+e2^2]]}];
s3=Re[{ArcSin[Sqrt[-1+e0^2+e1^2+e2^2]/
Sqrt[-1+e1^2+e2^2]]-
ArcSin[ e2/Sqrt[e1^2+e2^2]],2ArcSin[
Sqrt[e1^2+e2^2]], ArcSin[Sqrt[-1+e0^2+e1^2+e2^2]/
Sqrt[-1+e1^2+e2^2]]+ ArcSin[ e2/Sqrt[e1^2+e2^2]]}];
s4=Re[{ArcSin[Sqrt[-1+e0^2+e1^2+e2^2]/
Sqrt[-1+e1^2+e2^2]]+ArcSin[e2/Sqrt[e1^2+e2^2]],
-2ArcSin[ Sqrt[e1^2+e2^2]],
ArcSin[Sqrt[-1+e0^2+e1^2+e2^2]/
Sqrt[-1+e1^2+e2^2]]- ArcSin[e2/Sqrt[e1^2+e2^2]]}];
s5=Re[{- ArcSin[
Sqrt[-1+e0^2+e1^2+e2^2]/Sqrt[-1+e1^2+e2^2]]
- ArcSin[ e2/Sqrt[e1^2+e2^2]],2ArcSin[
Sqrt[e1^2+e2^2]],- ArcSin[ Sqrt[-1+e0^2+e1^2+e2^2]/
Sqrt[-1+e1^2+e2^2]]+ ArcSin[ e2/Sqrt[e1^2+e2^2]]}];
s6=Re[{- ArcSin[
Sqrt[-1+e0^2+e1^2+e2^2]/Sqrt[-1+e1^2+e2^2]]
+ArcSin[e2/Sqrt[e1^2+e2^2]],-2 ArcSin[
Sqrt[e1^2+e2^2]],-
```

```
ArcSin[ Sqrt[-1+e0^2+e1^2+e2^2]/Sqrt[
-1+e1^2+e2^2]]- ArcSin[e2/Sqrt[e1^2+e2^2]]}];
s7=Re[{ArcSin[Sqrt[-1+e0^2+e1^2+e2^2]/Sqrt[
-1+e1^2+e2^2]]- ArcSin[ e2/Sqrt[e1^2+e2^2]],
-2ArcSin[ Sqrt[e1^2+e2^2]],ArcSin[Sqrt[
-1+e0^2+e1^2+e2^2]/Sqrt[-1+e1^2+e2^2]]
+ ArcSin[ e2/Sqrt[e1^2+e2^2]]}];
s8=Re[{- ArcSin[ Sqrt[
-1+e0^2+e1^2+e2^2]/Sqrt[-1+e1^2+e2^2]]-
ArcSin[ e2/Sqrt[e1^2+e2^2]],-2 ArcSin[
Sqrt[e1^2+e2^2]],-ArcSin[
Sqrt[-1+e0^2+e1^2+e2^2]/Sqrt[-1+e1^2+e2^2]]
+ArcSin[e2/Sqrt[e1^2+e2^2]]}];
al=Table[ph={s1,s2,s3,s4,s5,s6,s7,s8}[[i]][[1]];
the={s1,s2,s3,s4,s5,s6,s7,s8}[[i]][[2]];
ps={s1,s2,s3,s4,s5,s6,s7,s8}[[i]][[3]];
{e0+Cos[0.5*(ph+ps)]*Cos[0.5the],
e1-Cos[0.5*(ph-ps)]*Sin[0.5the],
e2-Sin[0.5*(ph-ps)]*Sin[0.5the],
e3-Sin[0.5*(ph+ps)]*Cos[0.5the]},{i,1,8}];
su=Table[al[[j]][[1]]^2+al[[j]][[2]]^2+
al[[j]][[3]]^3+al[[j]][[4]]^2,{j,1,8}];
sm=Min[su];
For[k=1,k<=8,k++,If[su[[k]]==sm,
re={s1,s2,s3,s4,s5,s6,s7,s8}[[k]]]];
N[re]
]

(* Bouncing *)
rebounce[contactPointEmbeddedCoordinate_,chi_,
initialVelocity_,initialOrientationChangeRate_,
orientation_,m_,J_]:=Module[
{zInitial,psi0,theta0,phi0,psi,theta,phi,xi,
eta,zeta,a,b,c,d,e,f,g,h,i,j,k,l,solution,n
,o,p,q,zFinalPrint,angularVelocity},
psi=orientation[[1]];
theta=orientation[[2]];
phi=orientation[[3]];
xi=contactPointEmbeddedCoordinate[[1]];
eta=contactPointEmbeddedCoordinate[[2]];
```

```
zeta=contactPointEmbeddedCoordinate[[3]];
psi0=initialOrientationChangeRate[[1]];
theta0=initialOrientationChangeRate[[2]];
phi0=initialOrientationChangeRate[[3]];

zInitial=initialVelocity[[3]];

b=J*Cos[psi];
c=J*Sin[theta]*Sin[psi];
d=J(theta0*Cos[psi]+phi0*Sin[theta]*Sin[psi]);
a=(eta*Cos[theta]-zeta*Cos[phi]*Sin[theta]);
f=J*Sin[psi];
g=-J*Cos[psi]*Sin[theta];
h=J(theta0*Sin[psi]-phi0*Cos[psi]*Sin[theta]);
e=(zeta*Sin[theta]*Sin[phi]-xi*Cos[theta]);
i=J;
k=J*Cos[theta];
l=J(phi0*Cos[theta]+a);
j=(xi*Cos[phi]*Sin[theta]-eta*Sin[theta]*Sin[phi]);

n=1;
o=-Sin[theta]*(xi*Cos[theta]*Cos[psi]*(Sin[phi])^2
+eta*(Cos[theta/2])^2*Cos[psi]*Sin[2*phi]-
eta*(Sin[phi])^2*Sin[psi]+xi*(Cos[theta/2])^2*
Sin[2*phi]*Sin[psi]+
(Cos[phi])^2*(-xi*Cos[psi]+eta*Cos[theta]*
Sin[psi])-zeta*Sin[theta]*Sin[phi+psi]);
p=(-zeta*Cos[phi+psi]*Sin[theta]+Cos[theta]*
Cos[phi+psi]*
(eta*Cos[phi]+xi*Sin[phi])+(xi*Cos[phi]-eta*
Sin[phi])*Sin[phi+psi]);
q=-chi(zInitial+theta0(-zeta*Cos[phi+psi]*
Sin[theta]+Cos[theta]*Cos[phi+psi](eta*
Cos[phi]+xi*Sin[phi])+
(xi*Cos[phi]-eta*Sin[phi])Sin[phi+psi])+psi0*
Sin[theta](xi*Cos[theta]*Cos[psi]*(Sin[phi])^2+
eta*(Cos[theta/2])^2*Cos[psi]*Sin[2phi]-eta*
(Sin[phi])^2*Sin[psi]+xi*(Cos[theta/2])^2*
Sin[2*phi]*Sin[psi]+
(Cos[phi])^2*(-xi*Cos[psi]+eta*Cos[theta]*
```

```
Sin[psi])-zeta*Sin[theta]*Sin[phi+psi]));

solution=NSolve[b*theta1+c*phi1==d+a*S&&f*
theta1+g*phi1==h+e*S&&i*psi1+k*phi1==l+j*S
&&n*zFinal+o*psi1+p*theta1==q&&S==m*zFinal-
m*zInitial,{psi1,theta1,phi1,zFinal,S}];
solution={psi1,theta1,phi1,zFinal,S}/. solution;
solution=Flatten[solution];
zFinalPrint=-chi(zInitial+theta0(-zeta*
Cos[phi+psi]*Sin[theta]+Cos[theta]*Cos[phi+psi]
(eta*Cos[phi]+xi*Sin[phi])+(xi*Cos[phi]
-eta*Sin[phi])Sin[phi+psi])+psi0*Sin[theta]
(xi*Cos[theta]*Cos[psi]*(Sin[phi])^2+
eta*(Cos[theta/2])^2*Cos[psi]*Sin[2phi]-eta*(Sin[phi])^2*
Sin[psi]+xi*(Cos[theta/2])^2*Sin[2*phi]
*Sin[psi]+(Cos[phi])^2*
(-xi*Cos[psi]+eta*Cos[theta]*Sin[psi])-
zeta*Sin[theta]*Sin[phi+psi]))-(* LHS: *) solution[[2]](-zeta*Cos[phi+ps
+Cos[theta]*Cos[phi+psi]*(eta*Cos[phi]+xi*Sin[phi])+
(xi*Cos[phi]-eta*Sin[phi])*Sin[phi+psi])
+solution[[1]]*Sin[theta]*(xi*Cos[theta]
*Cos[psi]*(Sin[phi])^2
+eta*(Cos[theta/2])^2*Cos[psi]*Sin[2*phi]
-eta*(Sin[phi])^2*Sin[psi]+xi*(Cos[theta/2])^2*
Sin[2*phi]*Sin[psi]+(Cos[phi])^2*(-xi*Cos[psi]+
eta*Cos[theta]*Sin[psi])-zeta*Sin[theta]*
Sin[phi+psi]);
angularVelocity=eulerRateToAngularVelocity
[{solution[[1]],solution[[2]],solution[[3]]},
orientation];
{{initialVelocity[[1]],initialVelocity[[2]]
,zFinalPrint}
,{solution[[1]],solution[[2]],solution[[3]]}
,angularVelocity}
]

(* Free fall and bouncing *)
combined[orientation_,velocity_,
angularVelocity_,height_]:=Module[
{one,contact,two,eulerRates,moi},
```

```
If[Norm[{angularVelocity[[1]],
angularVelocity[[2]]}]==0,Return[7]];
one=freeFall[orientation,sixVectors,
eightVectors,velocity,angularVelocity,height];
one=N[Re[one]];
two={{1,1,1}};
While[two[[1]][[3]]>0,
contact=contactPoint[one[[3]]];
moi=MoI[Normalize[angularVelocity]];
eulerRates=angularVelocityToEulerRate
[angularVelocity,one[[1]]];
two=rebounce[vertices[[contact]],
CoR[[contact]],one[[4]],eulerRates,
one[[1]],m,moi];
two=N[Re[two]];
one=freeFall[one[[1]],one[[2]],one[[3]]
,two[[1]],two[[3]],one[[6]]];
one=N[one];];
getTopFace[one[[2]]]
]

(* Final step! *)
calcProbability[orientationRange_,
velocityRange_,angularVelocityRange_,
heightRange_]:=Module[
{allValues,a,b,c,d,e,f},
allValues=Flatten[Table[combined[
{i1,i2,i3},{j1,j2,j3},{k1,k2,k3},l],
Flatten[{i1,orientationRange[[1]]}],
Flatten[{i2,orientationRange[[2]]}],
Flatten[{i3,orientationRange[[3]]}],
Flatten[{j1,velocityRange[[1]]}],
Flatten[{j2,velocityRange[[2]]}],
Flatten[{j3,velocityRange[[3]]}],
Flatten[{k1,angularVelocityRange[[1]]}],
Flatten[{k2,angularVelocityRange[[2]]}],
Flatten[{k3,angularVelocityRange[[3]]}],
Flatten[{l,heightRange}]]]];
allValues=Flatten[allValues];
Print[allValues];
```

```
a=Count[allValues,1];
b=Count[allValues,2];
c=Count[allValues,3];
d=Count[allValues,4];
e=Count[allValues,5];
f=Count[allValues,6];
{a,b,c,d,e,f}/(a+b+c+d+e+f)
]


Print[calcProbability[{{0,2Pi,0.5},
{0,2Pi,0.5},{0,2Pi,0.5}},{{-0.9,1,0.5},
{0,0,0},{-0.9,1,0.5}},{{-2,2,1},{-2,2,1}
,{-2,2,1}},{0.2,1,0.4}]]
```