Cuboidal Dice Rolls

By Anthony The

Introduction:

Consider the following scenario. There is a cuboidal die with uniform density and dimensions *h* by 1 by 1. Of the six faces the die could land on, the die could either land on a face with an area of *h* or a face with an area of 1. Let each of the sides of the die with area *h* have a probability of being landed on p_{side} and let the sides of the die with area 1 have a probability of being landed on p_{end}. For my project, I wanted to determine the probability (p_{end}) of the die landing on one of the faces with an area of 1 as h varies from 0 to 2.

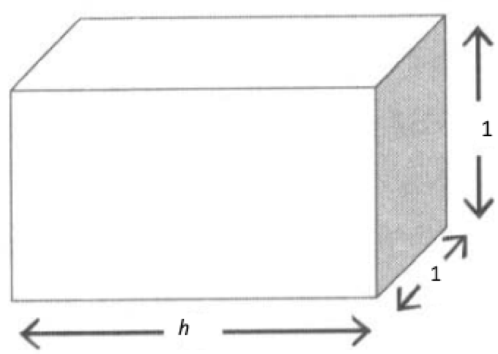
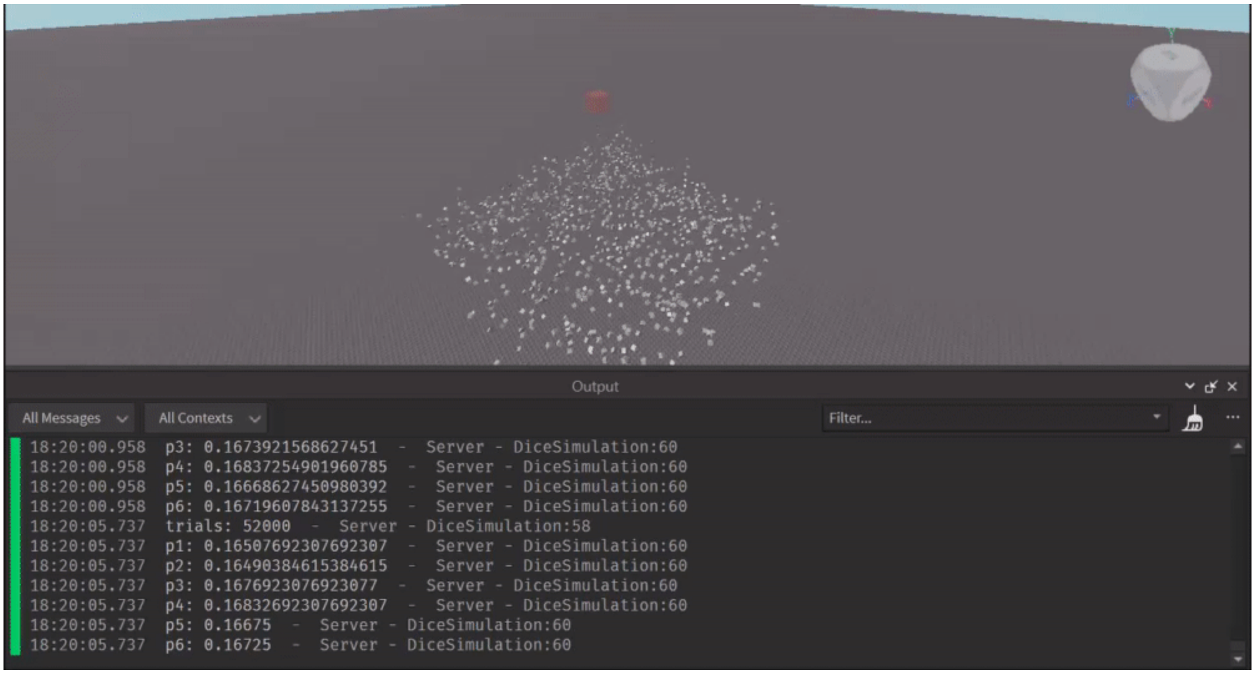


FIGURE 1

Simulation:

I simulated the die rolls in a program called Roblox Studio. Roblox Studio has a built in physics engine that can be customized fairly easily. For each trial, I gave the die a random angular velocity from (-10, 10) revolutions per second in each axis. I gave the die a random linear velocity from (-50, 50) studs per second in each axis. I used the default acceleration due to gravity which is 196.2 studs per second squared. I simulated 10000 trials for each value of h from [0.1, 2] with increments of 0.1 studs. To see the results, once a die stopped moving, I used a raycast to determine which face the die landed on.

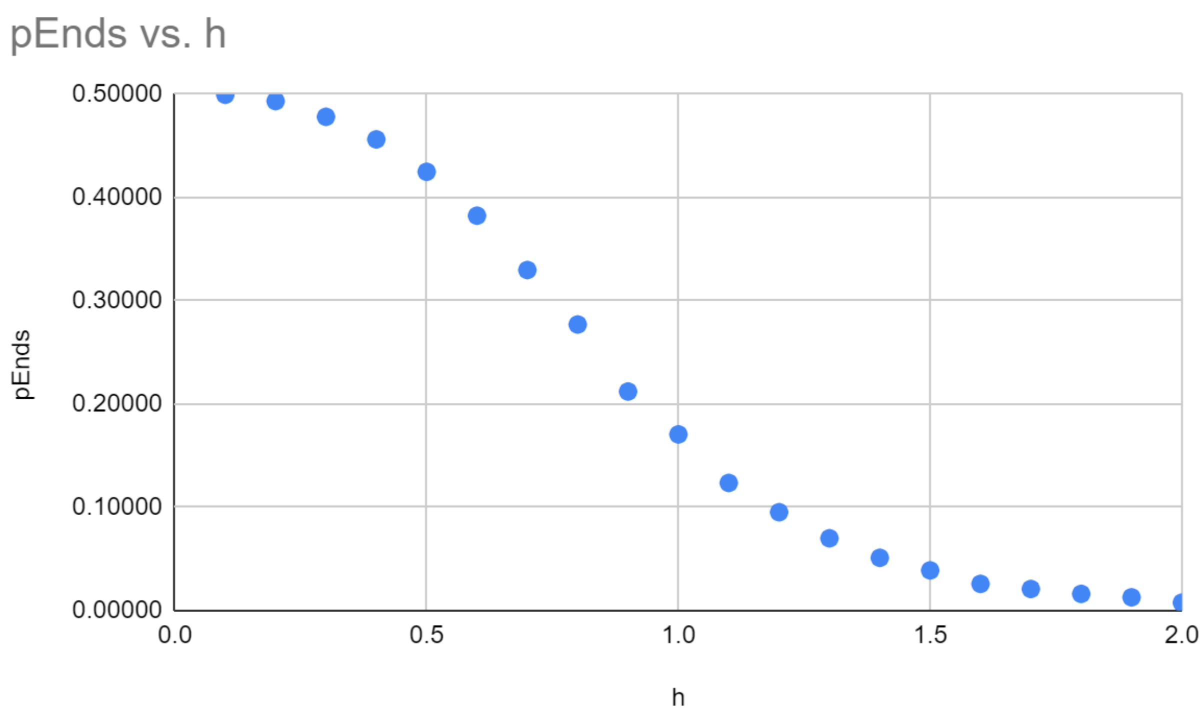


Below is a table of my data:

h	p_{end}	p_{side}
0.1	0.49950	0.00025
0.2	0.49335	0.00333
0.3	0.47815	0.01093
0.4	0.45630	0.02185
0.5	0.42490	0.03755
0.6	0.38240	0.05880
0.7	0.32980	0.08510
0.8	0.27705	0.11148
0.9	0.21215	0.14393
1.0	0.17065	0.16468
1.1	0.12365	0.18818
1.2	0.09530	0.20235
1.3	0.07005	0.21498
1.4	0.05115	0.22443
1.5	0.03895	0.23053
1.6	0.02590	0.23705

1.7	0.02095	0.23953
1.8	0.01620	0.24190
1.9	0.01290	0.24355
2.0	0.00775	0.24613

Below is a graph of the data:



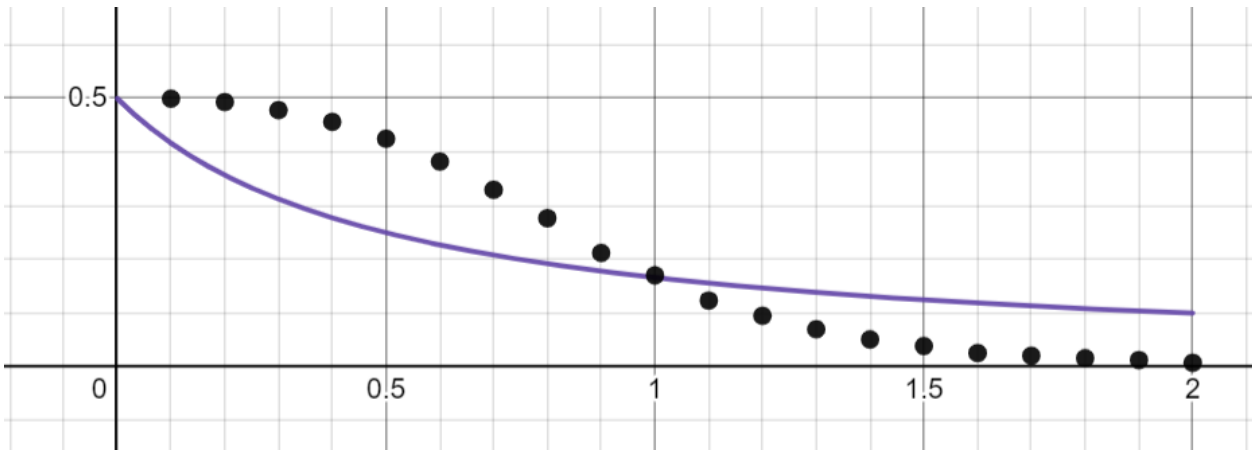
Additionally, I briefly tried running the simulation with different values for initial angular velocity, initial linear velocity, friction, elasticity and acceleration due to gravity. From my observations:

- p_{end} increased when gravity increased
- p_{end} decreased when initial angular velocity increased
- p_{end} increased initially when there was more initial linear velocity and then decreased
- p_{end} varied when using different values for the friction and elasticity coefficients

This suggests *h* is not the only factor that matters when predicting the outcome of the die roll.

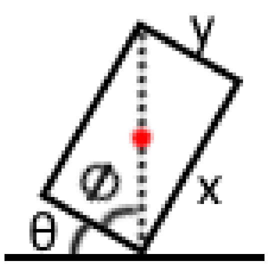
Deriving an Equation:

The papers I read on this topic confirmed that it is difficult to predict the outcome of a die roll beforehand as many factors impact the probability of the outcomes. A naive approach would be to model the probability based on the area of each face relative to the surface area. However, you will see that it doesn't fit the data very well.



Another approximation could be made by analyzing the center of mass. Imagine you choose an arbitrary orientation for the die, hold the die to a surface, and release it and see what side it lands on.

This can be simplified by looking at the 2D scenario for this first. Let $x = h$ and $y = 1$. Let p_{2D} be the probability of landing on a given side with length y. Assume the die is now a rectangle with dimensions x by y. Since we have assumed uniform density, the center of mass is right in the center of the rectangle. Given the image below, if you draw a line straight down from the center of mass, the face it intersects with will be the face the die lands on.



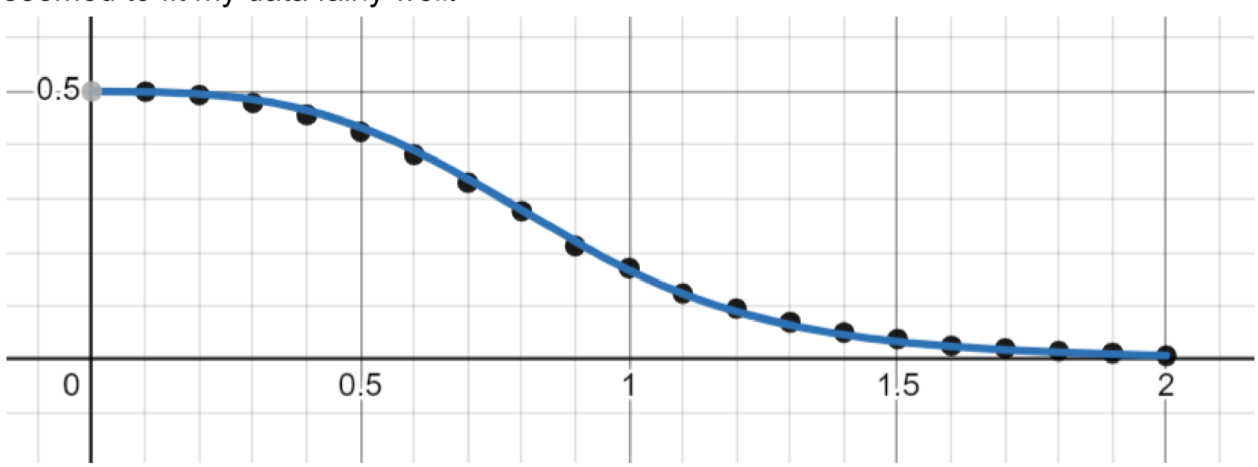
In the image above, $\Phi = \tan^{-1}(x/y)$. Let $\theta_0 = \pi/2 - \Phi$. At this angle, the die will be perfectly balanced. If $0 < \theta < \theta_0$ then the die will land on y. Note θ ranges from $(0, \pi/2)$. Thus $p_{2D} = \theta_0/(\pi/2) = (\pi/2 - \tan^{-1}(x/y))/(\pi/2) =$ $\mathbf{1 - 2/\pi * \tan^{-1}(x/y)}$.

Bringing this into 3D, assume we now have a cuboid with dimensions *x* by *y* by *y*. Let p_{3D} be the probability of the die landing on an end face (a face with dimensions *y* by *y*). We know that the cuboid will most likely land on a corner because there is likely to be some rotation on all three axes. However, this can be thought of as landing on whichever edge is closest to the floor because the cuboid will fall to that edge first. If the cuboid lands on an edge of length *y*, we can then use p_{2D} to model the probability of it landing on an end. Since we are only looking at one face, the die has to land on one of four edges length *y*. We can calculate p_{3D} by weighing the length of the edges that could be landed on. Thus p_{3D} = 4y/(8y + 4x)*p_{2D} = **2/(2 + x/y)*(1 - 2/π*$\tan^{-1}$(x/y))**

However, one problem with this approximation is it assumes the die lands on the corner with equal probability of all orientations. When a die is rolled, it is more likely to land on a larger face because

- More elastic vibrations can be excited over a larger area so the die slows more
- It has a higher impact speed since the center of mass is lower and thus loses more speed

To account for this, we can weigh the terms x/y in p_{3D} by some positive power *n*. This effectively makes larger sides larger and smaller sides smaller since the ratio x/y will increase if x/y > 1 and decrease if x/y < 1. Thus, p_{end} = 1/(2 +$(x/y)^n$)*(1 - 2/π*$\tan^{-1}$($(x/y)^n$)) = **1/(2 + h^n)*(1- 2/π*$\tan^{-1}$(h^n))**. In the real world, a value of *n* between 2.5 to 3.5 seemed to model dice rolls of various materials and throwing methods well. A value of *n* = 3 seemed to fit my data fairly well.



Another Model:

After thinking about it some more. I had a problem with the previous model. On the second step when the model is brought into 3D, it assumes the probability of landing on an edge is based on the length of an edge which was the same issue with the surface

area model and defeats the purpose of trying to use the center of mass to determine which face the die will fall on.
Here is a new idea, assume we are holding the die at a corner and are about to release it. If you draw a line straight down from the center of mass, whichever face this line passes through, that's what face the die will land on.

A way of calculating this is through finding the solid angle of each of the faces. A solid angle of a surface is defined as the surface area of a unit sphere covered by the surface's projection onto the sphere.

This can be written as:

$$\Omega = \iint_S \frac{\hat{r} \cdot \mathrm{d}A}{r^2}$$

Where r^ is a unit vector from the origin, dA is the differential area of a surface patch, and r is the distance from the origin to the patch.

Take one of the faces and let it lie in the plane z = Z. Let the face have dimensions 2X by 2Y. Let Φ be the angle between z^ and r^.

By definition of dot product:

$$\hat{r} \cdot dA = cos\phi dxdy$$

By definition of cos:

$$cos\phi = \frac{Z}{r}$$

$$r^2 = x^2 + y^2 + z^2$$

Putting it all together:

$$\Omega = \int_{-Y}^{Y} \int_{-X}^{X} \frac{Z dxdy}{(x^2+y^2+Z^2)^{\frac{3}{2}}}$$

I believe this simplifies to:

$$4\tan^{-1}\left(\frac{XY}{Z\sqrt{X^2+Y^2+Z^2}}\right)$$

For my die specifically, for one of the square faces we have Z = h/2, X = 1/2, Y = 1/2.

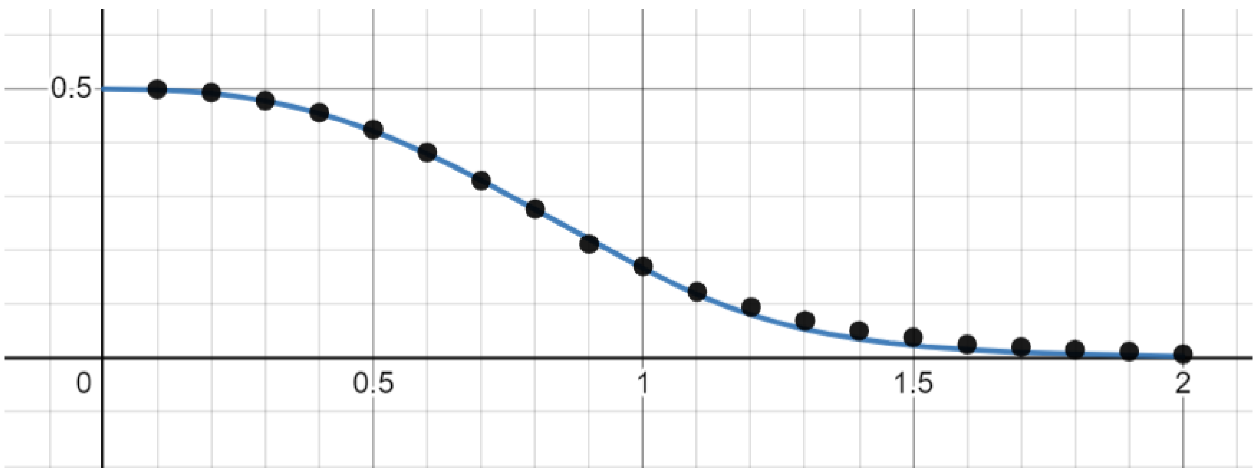
Thus,

$$\Omega = 4tan^{-1}(\frac{1}{h\sqrt{2+h^2}})$$

Divide it by 4pi, the surface area of a unit sphere, to get the p_{end}.

$$p_{end} = \frac{1}{\pi}tan^{-1}(\frac{1}{h\sqrt{2+h^2}})$$

Graph using the new function with the same technique in the first model of raising h to a power of *n* in this case I did *n* = 2.5.



Exercises:

1. Model the probability for a cuboidal die of dimensions 2 by 1 by 1 to land on an end by varying initial angular velocity instead of height.
2. Model the tendency for a cuboidal die of dimensions 2 by 1 by 1 to be in certain ranges of “close” orientations before it hits the ground given a random initial

angular velocity of fixed magnitude. See if there is a specific axis the block naturally prefers to rotate around.

3. Create a simulation of cuboidal dice rolls with varying heights using your favorite physics engine and programming language with a “hollow” block i.e. a box with thin sides. Compare your results to a simulation of dice rolls with a solid block under the same conditions.
4. Create a simulation of dice rolls using your favorite physics engine and programming language with tetrahedral dice of varying height.
5. Derive an equation to find what side a tetrahedral die would land on if put on a table at a random orientation and released.

Additional Links:

[1] My simulation code:
https://gist.github.com/athe27/15d88ada057268660204f9308cf53445

[2] A thread describing another approach to modeling the probability of the die using the center of mass and a solid angle approach:
https://physics.stackexchange.com/questions/41297/how-to-determine-the-probabilities-for-a-cuboid-die

[3] A paper describing the approach I wrote about for modeling the probability:
https://www.cambridge.org/core/services/aop-cambridge-core/content/view/7B4316554D9BC23B9212F73472E1FD92/S0025557200005635a.pdf/9716-probability-analysis-for-rolls-of-a-square-cuboidal-die.pdf

[4] A paper describing modeling the probability with Gibbs distribution:
https://arxiv.org/pdf/1302.3925.pdf

[5] My slide presentation:
https://docs.google.com/presentation/d/1pZ6xpMDtreb8qi1KNIp8sc3ztgabv5nUaVWtwehNL8A/edit?usp=sharing