Computing Pi with Bouncing Cubes

Why Does π Arise in the Number of Collisions Between Two Masses and a Wall?

Liam Ilan

An Extended Essay in Mathematics

Session: May 2023

Word Count: 3419

Table of Contents

Introduction

Two cubes exist on a frictionless plane, such that one cube (cube B) with a mass m_2, is placed between another cube (cube A) with a mass m_1, and a wall (Sanderson, 2019). m_1 and m_2 are defined as:

$$m_1 = m_2 100^n \{n \in \mathbb{Z} \,|\, n \geq 0\}$$

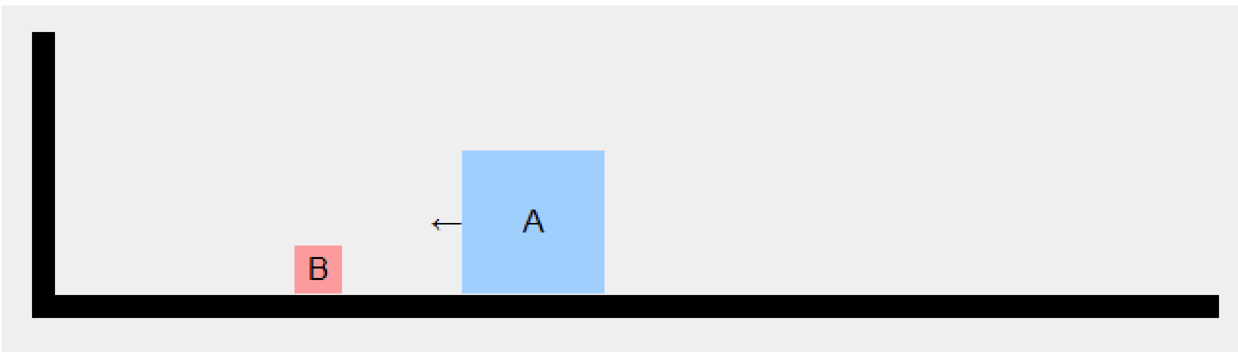

Figure 1. The Initial Setup. This figure is a screen capture of an animated simulation, built by the author of this paper. It can be found at https://bouncing-cubes.snowboardsheep.repl.co/.

Now send cube A with some velocity towards cube B. Assuming that collisions are completely elastic, the number of collisions (N) between cube B and cube A, or cube B and the wall, will be an integer whose digits are the first $n + 1$ digits of π. For example, if $m_2 = 1\,kg$, and $m_1 = m_2 \cdot 100^3 = 1000000\,kg$, then the number of collisions will be the first $3 + 1$ or 4 digits of pi (3141).

The objective of this essay is to explain why π arises from this seemingly arbitrary set of conditions.

Research Question

Why does π arise in the number of collisions between two masses and a wall?

Configuration Spaces

This paper relies on the concept of a configuration space. In physics, a configuration space is defined as a space that "gives you all the possible states [a system] can be in." (Freiberger, 2016, para. 1). Throughout this paper, configuration spaces will be used to model the possible displacements and velocities our bouncing cube system can be in.

Software

The software used to generate all graphs, visualizations, and figures, was written by the author of this paper. It can be downloaded from https://github.com/liam-ilan/extended-essay. See `README.MD` in the repo for details on software usage. The free body diagrams are screen captures from an interactive simulation built by the author for this paper. It can be found at https://bouncing-cubes.snowboardsheep.repl.co/.

A Kinematic Approach

To solve this problem, it is helpful to plot a configuration space of possible velocities. We will use this configuration space to derive π throughout this investigation (Sanderson, 2019).

Building a Configuration Space of Velocities

Let's name the velocity of cube A, v_1. Likewise, the velocity of cube B will be v_2 (Sanderson, 2019). We will set the y axis of our configuration space to be $v_2\sqrt{m_2}$, and the x axis to be $v_1\sqrt{m_1}$, such that $y = v_2\sqrt{m_2}$, and $x = v_1\sqrt{m_1}$. We can use the equation for the conservation of energy (Equation 1), to plot a configuration space of possible states of our velocities.

$$\frac{1}{2}m_1v_1^2 + \frac{1}{2}m_2v_2^2 = c\text{, where c is constant over time} \tag{1}$$

Substituting in x and y,

$$\frac{1}{2}x^2 + \frac{1}{2}y^2 = c \tag{2}$$

Since $kx^2 + ky^2 = c\ \{k \in \mathbb{R}\}$ graphs a circle, Equation 2 must do the same. This is the orange circle in figure 2, which is where π will ultimately arise from.

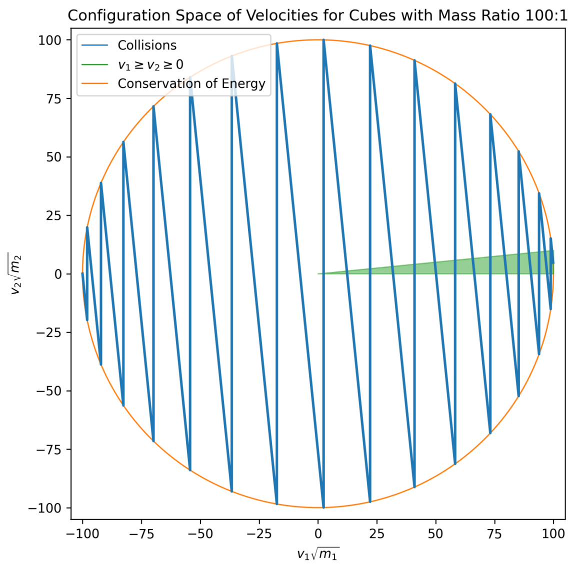

Figure 2. A configuration space of cube A and cube B, with a mass ratio of 100:1, generating 31 collisions (2 digits of π). This figure was created with software built by the author of this paper. The software can be found at https://github.com/liam-ilan/extended-essay.

Our state starts on the left of figure 2, where $v_2 = 0$ and v_1 is the initial velocity of cube A. Note that v_1 is negative since we assume that cube A starts moving left.

On each collision between two blocks, we can use the conservation of momentum (Equation 3), to find where our state moves to (Sanderson, 2019).

$$m_1v_1 + m_2v_2 = g, \text{ where g is constant over time} \tag{3}$$

Substituting x and y in,

$$\sqrt{m_1}x + \sqrt{m_2}y = g$$

Rearranging this into the form $y = mx + b$,

$$y = \frac{-\sqrt{m_1}}{\sqrt{m_2}}x + \frac{g}{\sqrt{m_2}}$$

On each collision, we can use the slope of our equation ($\frac{-\sqrt{m_1}}{\sqrt{m_2}}$) to graph a line of possible states where momentum is conserved. These lines are the diagonal blue lines found in Figure 2. This line must intersect the current state, and the state of the system post-collision. These two points must also obey the conservation of energy, thus, they must also intersect Equation 2.

When a collision between cube B and the wall occurs, v_2 is turned to the other direction. Thus,

$$v_{2_f} = -v_{2_i}$$

where v_{2_i} is the velocity of the block before the collision, and v_{2_f} is the velocity of the block after the collision (Sanderson, 2019). This is equivalent to jumping to the vertically opposite side of the circle in figure 2, and is represented by a vertical blue line.

We can stop counting collisions when $0 \leq v_2 \leq v_1$ (Sanderson, 2019). This corresponds to when Cube A is moving away from Cube B, faster than Cube B. This region is colored in green in figure 2.

Since each blue line in figure 2 represents a jump in the state post-collision, counting collisions is equivalent to counting lines in this circle (Sanderson, 2019). With each new jump, we create a new arc on the circumference of the circle, which is separated by two blue lines. Thus, counting collisions is also equivalent to counting arcs separated by said lines.

Connection to Pi

Let's name the angle between a block-on-block collision and a block-on-wall collision (a diagonal line, and a vertical line) θ (Sanderson, 2019). Figure 3 isolates one such pair for easier visualization. This angle will be the same for any given pair of lines, and thus any given arc, since all such angles are formed between a vertical line and a line of slope $\frac{-\sqrt{m_1}}{\sqrt{m_2}}$.

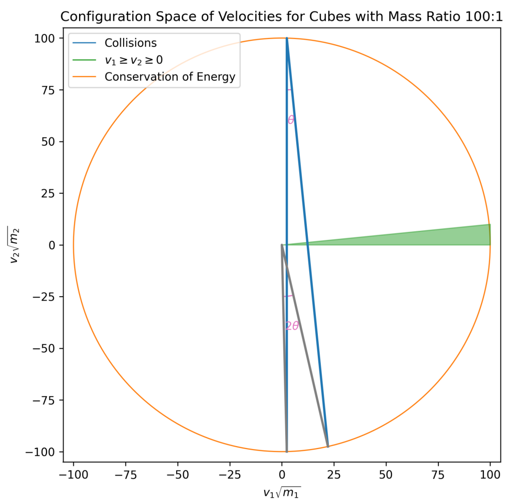

Figure 3. A configuration space of possible velocities, with a focus on two collisions. This figure was created with software built by the author of this paper. The software can be found at https://github.com/liam-ilan/extended-essay.

The inscribed angle theorem states that an angle subtending an arc on a circle from the center of said circle will be twice the angle subtending the same arc from another point on the circumference of that circle.

Thus, the angle subtending each arc between two lines is 2θ (Sanderson, 2019). This angle is marked with the gray lines in figure 3. Since a full rotation is $2\pi\, rad$, the number of collisions, will be the number of times 2θ fits into $2\pi\, rad$, or the number of times θ fits into $\pi\, rad$ (henceforth referred to as N). Thus, N is the largest integer such that

$$N\theta < \pi$$

If N was larger than π, we would exceed the number of arcs that could fit into the circle, and we would go past our stop condition ($0 \le v_2 \le v_1$). For cases where N is some number of digits of π, θ must be around 10^{-n} where $\{n \in \mathbb{Z} \mid n \ge 0\}$ and $n + 1$ is the number of computed digits of π. As an example, if $n = 1$, then $\theta = 10^{-1} = 0.1$, and $N = \lfloor \frac{\pi}{\theta} \rfloor = \lfloor \frac{\pi}{0.1} \rfloor = 31$. This is because, for cases such as this, N is the largest integer such that $N10^{-n} < \pi$ or $N < \pi 10^{n}$. In other words, when $\theta \approx 10^{-n}$, when N is $n + 1$ of π.

Solving for Theta

Our next step is to show that $\theta \approx 10^{-n}$ when $m_1 = m_2 100^{n}$ (Sanderson, 2019). To do this, we can find $tan(\theta)$. Recall that $tan(\theta) = \frac{opposite}{adjacent}$. Since $\frac{-\sqrt{m_1}}{\sqrt{m_2}}$, or $\frac{-adjacent}{opposite}$ is the slope of the diagonal lines in figure 3,

$$tan(\theta) = \frac{\sqrt{m_2}}{\sqrt{m_1}}$$

For cases where $m_1 = m_2 100^{n}$,

$$tan(\theta) = \frac{\sqrt{m_2}}{\sqrt{m_2 100^{n}}}$$

$$= \frac{\sqrt{m_2}}{\sqrt{m_2 100^{n}}}$$

$$= \sqrt{\frac{m_2}{m_2 100^{n}}}$$

$$= \sqrt{100^{-n}}$$

$$= 10^{-n}$$

Thus,

$$\theta = tan^{-1}(10^{-n})$$

We can expand $tan^{-1}(x)$ into a Maclaurin series,

$$tan^{-1}(x) = \sum_{k=0}^{\infty} \frac{tan^{(k)}(0)}{k!} x^k$$

$$= x - \frac{x^3}{3} + \frac{x^5}{5} - ...$$

Note: The maclaurin series expansion of $tan^{-1}(x)$ can be found in the IB Data Booklet.

Since all the following terms' exponents past $\frac{x^3}{3}$ increase, for values of x between 0 and 1, only the first term is significant (Sanderson, 2019). Thus, where $0 \leq x \leq 1$, $tan^{-1}(x) \approx x$. Because $n \geq 0$, $0 \leq 10^{-n} \leq 1$, thus, $\theta = tan^{-1}(10^{-n}) \approx 10^{-n}$ for cases where $m_1 = m_2 100^n$.

From Theta to Pi

Since $\theta \approx 10^{-n}$ when $m_1 = m_2 100^n$, and for N to be $n + 1$ of digits of π, $\theta \approx 10^{-n}$, we can conclude that when $m_1 = m_2 100^n$, N is $n + 1$ digits of π.

An Optical Approach

A kinematic approach using a configuration space of velocities is enough to explain why π appears in this system. With that being said, simulating these cubes by computing their velocities can introduce error, as the accuracy of the simulation is reliant on the size of the timestep in the simulation (Sanderson, 2019). This error is small enough such that the graphs and visuals in this essay were generated using this method, however, there is a method of explaining the occurrence of π, using a configuration space of displacements instead. Using displacements removes the need for a timestep, eliminating this source of error in simulation. It also provides more intuition as to why π arises in this system.

Building a Configuration Space of Displacements

Let's name the displacement of cube A measured from the left side of the cube, d_1. Likewise, the displacement of cube B measured from the right side of the cube will be d_2 (Sanderson, 2019).

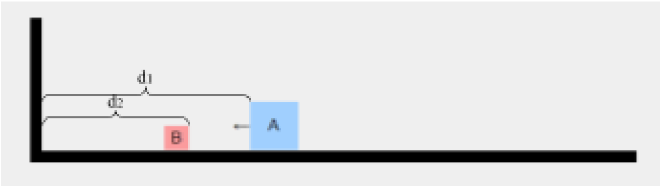

Figure 4. d_1 and d_2 illustrated in a modified version of Figure 1. This figure was created with software built by the author of this paper. The software can be found at https://github.com/liam-ilan/extended-essay.

A collision between cube B and cube A is equivalent to points where $d_1 = d_2$. Additionally, a collision with the wall is equivalent to points where $d_2 = the\ width\ of\ cube\ B$.

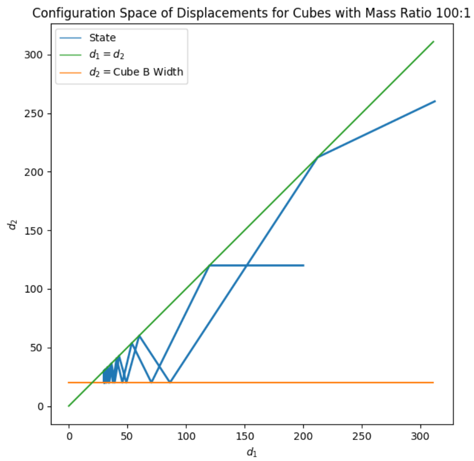

Figure 5. A configuration space of displacements. The blue line is the path of the state that the cubes follow over time. This figure was created with software built by the author of this paper. The software can be found at https://github.com/liam-ilan/extended-essay.

The blue line in figure 5 resembles a beam of light reflecting between two mirrors. However, it does not follow the law of reflection.

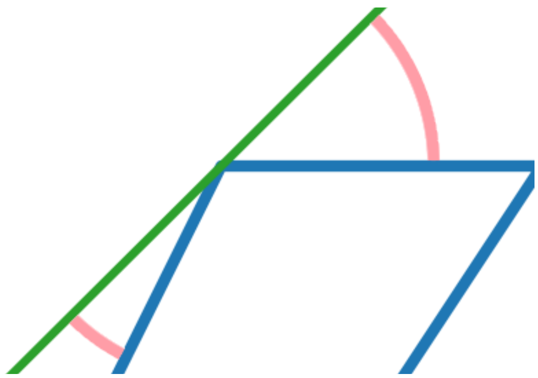

Figure 6. Zooming in on figure 5 to one collision, we can see that the angle of incidence ≠ the angle of reflection, thus breaking the law of reflection. This figure was created with software built by the author of this paper. The software can be found at https://github.com/liam-ilan/extended-essay.

Additionally, the velocity of the state on the configuration space is not constant. Beams of light in the real world have a constant speed. It is helpful to meet these two rules, so that we can use them to determine the path of our state, thus determining the number of collisions.

Adapting the Conservation of Energy

Since the state of our configuration space moves with respect to time, we can think of it as having some velocity u. $u_x = \frac{dx}{dt}$ where t is time (Sanderson, 2019). Likewise, $u_y = \frac{dy}{dt}$. Since $x = d_1$, and $y = d_2$, $u_x = v_1$, and $u_y = v_2$. Thus,

$$|u| = \sqrt{u_x^2 + u_y^2}$$

$$= \sqrt{(\frac{dx}{dt})^2 + (\frac{dy}{dt})^2}$$

$$= \sqrt{v_1^{\ 2} + v_2^{\ 2}}$$

Scaling y by a factor of $\sqrt{m_2}$, and x by by a factor of $\sqrt{m_1}$,

$$|u| = \sqrt{(v_1\sqrt{m_1})^2 + (v_2\sqrt{m_2})^2}$$

$$= \sqrt{m_1 v_1^2 + m_2 v_2^2}$$

$$= \sqrt{2(\frac{1}{2}m_1 v_1^2 + \frac{1}{2}m_2 v_2^2)}$$

Recall that the conservation of energy states that $\frac{1}{2}m_1 v_1^2 + \frac{1}{2}m_2 v_2^2 = c$, where c is constant over time, thus,

$$|u| = \sqrt{2c}$$

Thus $|u|$ is constant over time when $x = d_1\sqrt{m_1}$ and $y = d_2\sqrt{m_2}$. In other words by scaling x and y by $\sqrt{m_1}$ and $\sqrt{m_2}$, the speed of the state in the configuration space, much like the speed of light, is constant.

Since we scale our space by a factor of $\sqrt{m_1}$ on the x axis, and $\sqrt{m_2}$ on the y axis, the line that represents a collision between two blocks is no longer $y = x$, rather, it is, the line $y\sqrt{m_1} = x\sqrt{m_2}$, or $y = x\frac{\sqrt{m_2}}{\sqrt{m_1}}$ (Sanderson, 2019).

Adapting the Conservation of Momentum

Recall the conservation of momentum, $m_1 v_1 + m_2 v_2 = g$, where g is constant over time. This is equivalent to

$$\begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \cdot \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = g$$

Thus,

$$\begin{pmatrix}\sqrt{m_1}\\ \sqrt{m_2}\end{pmatrix}\cdot u = g$$

$$\left|\begin{pmatrix}\sqrt{m_1}\\ \sqrt{m_2}\end{pmatrix}\right||u|\cos(\alpha) = g$$

Where h is constant over time,

$$\cos(\alpha) = h$$

Since the slope of the collision line is $\frac{\sqrt{m_2}}{\sqrt{m_1}}$, α is the angle formed between the said line and u (Sanderson, 2019).

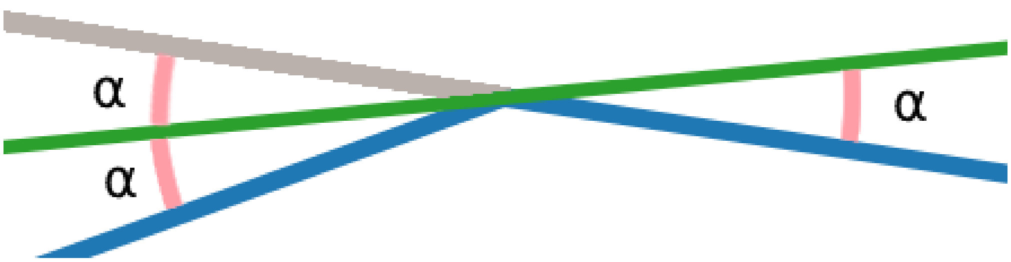

Figure 7. A configuration space, scaled by the roots of the masses, zoomed in on one collision. This figure was created with software built by the author of this paper. The software can be found at https://github.com/liam-ilan/extended-essay.

Since $cos(\alpha)$ must be constant, the angle u made with the collision line before a collision must be equal to the angle after the collision. In other words, the angle of incidence must be equal to the angle of reflection. On a collision with $d_2 = width\, of\ cube\, B$, the y component of u will reflect vertically, thus also maintaining the law of reflection.

Finding Pi Using Mirrors

The number of collisions (N) between two cubes is equal to the number of times our state collides with $y = x\frac{\sqrt{m_2}}{\sqrt{m_1}}$ or $d_2 = width\, of\, cube\, B$. Since our state obeys the law of reflection, we can treat our state like a ray of light, and our lines of collision as mirrors (Sanderson, 2019).

Let θ be the angle between $y = x\frac{\sqrt{m_2}}{\sqrt{m_1}}$ and $d_2 = width\, of\, cube\, B$, thus $\theta = tan^{-1}(\frac{\sqrt{m_2}}{\sqrt{m_1}})$. We know that when $m_1 = m_2 100^n$, $\theta \approx 10^{-n}$ (see a Kinematic Approach - Solving for Theta for more details).

Counting the number of collisions of the ray is equivalent to reflecting the configuration space as seen in Figure 8 (Sanderson, 2019). Thus, we can compute N by counting the number of times a straight ray intersects all the mirrors.

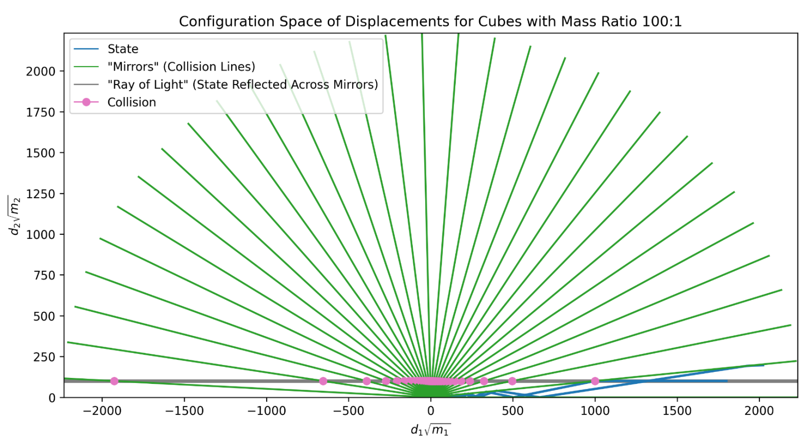

Figure 8. The configuration space is repeatedly reflected, such that the number of times a straight ray intersects all the mirrors is the number of collisions. This figure was created with software built by the author of this paper. The software can be found at https://github.com/liam-ilan/extended-essay.

Our ray will intersect any mirror whose angle of rotation is less than $\pi\, rad$, thus the number of collisions (N) is the largest integer such that $N\theta < \pi\, rad$. Since $\theta \approx 10^{-n}$,

$$N < \frac{\pi}{10^{-n}}$$

$$N = \lfloor \frac{\pi}{10^{-n}} \rfloor$$

In other words, when $m_1 = m_2 100^n$, N is $n + 1$ of digits of π.

Simulation and Computation

As mentioned before (See An Optical Approach), simulating our bouncing cubes using velocities and timesteps may introduce errors due to the size of the timestep. With that being said, we can simulate the cubes with precise calculations, as opposed to estimations utilizing timesteps. The goal of this section will be to produce an algorithm that accurately predicts the position of the two cubes at any point in time.

This algorithm can help us observe and reason with our phenomena experimentally, providing the opportunity to test different initial conditions, different masses, bases, etc. This is important in order to gain a deeper understanding of our problem, and potentially can act as a starting point for future work. Since such simulations are the only way to observe the impractical conditions of our setup, they are critical to gaining an understanding of the problem.

Defining Collisions

There are two types of collisions in our system. The first is a collision between Block B and the wall. Previously we have defined d_2 as the displacement of cube B, measured from the wall to the right edge. For ease of use, in this section, we will redefine d_2 as the displacement of cube B, measured from the wall to the left edge. We will keep d_1 as the displacement of cube A, measured from the wall to the left edge.

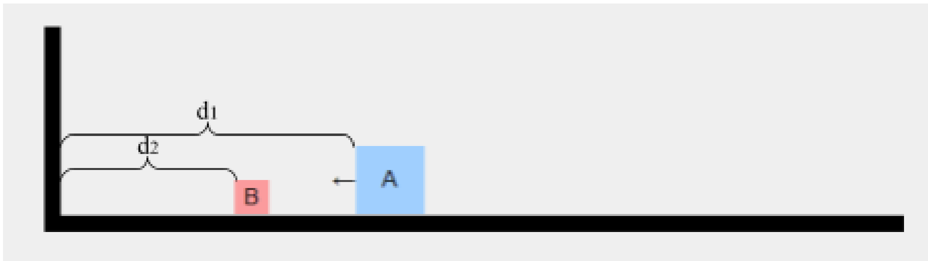

Figure 9. The new definitions for d_1 and d_2. This figure is a modified screen capture of an animated simulation, built by the author of this paper. It can be found at https://bouncing-cubes.snowboardsheep.repl.co/

A collision between the wall and block B occurs when

$$d_2 = 0 \tag{4}$$

Following references to this type of collision will be called Block-on-Wall collisions. Additionally, a collision between block B and block A occurs when

$$d_1 = d_2 + w_2 \tag{5}$$

where w_2 is the width of block B. These collisions will be referred to as Block-on-Block collisions.

Time to Collisions

We can express d_2 in terms of the initial displacement of cube B (or d_{2_i}), v_2, and a change in time (or t_c).

$$d_2 = d_{2_i} + v_2 t_c \tag{6}$$

Likewise, d_1 can be expressed in a similar way.

$$d_1 = d_{1_i} + v_1 t_c \tag{7}$$

A Block-on-Wall collision occurs when $d_2 = 0$ (Equation 4). Thus, such a collision also occurs when

$$d_{2_i} + v_2 t_c = 0$$

where t_c is now the time from a given displacement to a block-on-wall collision. Rearranging,

$$t_c = -\frac{d_{2_i}}{v_2}$$

A Block-on-Block collision occurs when $d_1 = d_2 + w_2$ (Equation 5). Thus, such a collision also occurs when

$$d_{1_i} + v_1 t_c = d_{2_i} + v_2 t_c + w_2$$

Solving for t_c

$$v_1 t_c - v_2 t_c = d_{2_i} - d_{1_i} + w_2$$

$$t_c(v_1 - v_2) = d_{2_i} - d_{1_i} + w_2$$

$$t_c = \frac{d_{2_i} - d_{1_i} + w_2}{v_1 - v_2}$$

To summarize, for block-on-wall collisions,

$$t_c = -\frac{d_{2_i}}{v_2} \tag{8}$$

For block-on-block collisions,

$$t_c = \frac{d_{2_i} - d_{1_i} + w_2}{v_1 - v_2} \tag{9}$$

We will use this to bypass the need for a timestep in our final algorithm.

Computing Velocities After a Block-on-Block Collision

Recall that on a Block-on-Block collision, both energy (Equation 1) and momentum (Equation 3) must be conserved. We can rearrange the conservation of energy,

$$\frac{1}{2}m_1v_{1_i}^{\ 2} + \frac{1}{2}m_2v_{2_i}^{\ 2} = \frac{1}{2}m_1v_{1_f}^{\ 2} + \frac{1}{2}m_2v_{2_f}^{\ 2}$$

$$m_1v_{1_i}^{\ 2} + m_2v_{2_i}^{\ 2} = m_1v_{1_f}^{\ 2} + m_2v_{2_f}^{\ 2}$$

$$m_1v_{1_i}^{\ 2} - m_1v_{1_f}^{\ 2} = m_2v_{2_f}^{\ 2} - m_2v_{2_i}^{\ 2}$$

$$m_1(v_{1_i}^{\ 2} - v_{1_f}^{\ 2}) = m_2(v_{2_f}^{\ 2} - m_2v_{2_i}^{\ 2})$$

$$m_1(v_{1_i} - v_{1_f})(v_{1_i} + v_{1_f}) = m_2(v_{2_i} - v_{2_f})(v_{2_i} + v_{2_f}) \tag{10}$$

We can rearrange the conservation of momentum,

$$m_1v_{1_i} + m_2v_{2_i} = m_1v_{1_f} + m_2v_{2_f}$$

$$m_1v_{1_i} - m_1v_{1_f} = m_2v_{2_f} - m_2v_{2_i}$$

$$m_1(v_{1_i} - v_{1_f}) = m_2(v_{2_f} - v_{2_i})$$

Using Equation 10,

$$v_{1_i} + v_{1_f} = v_{2_i} + v_{2_f} \tag{11}$$

We can also rearrange the conservation of momentum to

$$v_{2_f} = \frac{m_1v_{1_i} + m_2v_{2_i} - m_1v_{1_f}}{m_2}$$

Using Equation 11,

$$v_{1_i} + v_{1_f} = \frac{m_1v_{1_i} + m_2v_{2_i} - m_1v_{1_f}}{m_2} + v_{2_i}$$

Solving for v_{1_f},

$$v_{1_f} = \frac{m_1 v_{1_i} + m_2 v_{2_i} - m_1 v_{1_f}}{m_2} + v_{2_i} - v_{1_i}$$

$$= \frac{m_1 v_{1_i} + m_2 v_{2_i}}{m_2} - \frac{m_1}{m_2} v_{1_f} + v_{2_i} - v_{1_i}$$

$$(1 + \frac{m_1}{m_2}) v_{1_f} = \frac{m_1 v_{1_i} + m_2 v_{2_i}}{m_2} + v_{2_i} - v_{1_i}$$

$$v_{1_f} = \frac{\frac{m_1 v_{1_i} + m_2 v_{2_i}}{m_2} + v_{2_i} - v_{1_i}}{\frac{m_2 + m_1}{m_2}}$$

$$= \frac{m_2}{m_2 + m_1} (\frac{m_1 v_{1_i} + m_2 v_{2_i}}{m_2} + v_{2_i} - v_{1_i})$$

$$= \frac{m_2}{m_2 + m_1} (\frac{m_1 v_{1_i} + m_2 v_{2_i}}{m_2} + v_{2_i} - v_{1_i})$$

$$= \frac{m_1 v_{1_i} + m_2 v_{2_i} + m_2 v_{2_i} - m_2 v_{1_i}}{m_2 + m_1}$$

$$v_{1_f} = \frac{(m_1 - m_2) v_{1_i} + 2 m_2 v_{2_i}}{m_2 + m_1} \tag{12}$$

Since both cubes must follow this rule, it is also true that

$$v_{2_f} = \frac{(m_2 - m_1) v_{2_i} + 2 m_1 v_{1_i}}{m_1 + m_2} \tag{13}$$

Constructing the Algorithm

There are a couple important observations we can take advantage of, when constructing our algorithm. The first collision will always be a Block-on-Block collision. This is because cube B is stationary, and cube A is moving towards cube B at some velocity v_1. Additionally, after any

Block-on-Block collision, the next collision will be a Block-on-Wall collision. This will help us determine the next type of collision in our algorithm.

1. Let d_1 be the displacement of Cube A, d_2 be the displacement of Cube B, v_1 be the velocity of Cube A, v_2 be the velocity of Cube B, m_1 be the mass of Cube A, m_2 be the mass of Cube B, and w_2 be the width of Cube B. Let t be the total time in the system, and t_c be the time until the next collision. Record the initial conditions of d_1, d_2, v_1, v_2, and t.
2. The first collision is always a Block-on-Block collision. Thus, we can use Equation 9, $t_c = \frac{d_{2_i} - d_{1_i} + w_2}{v_1 - v_2}$, to solve for the time to a block on block collision.
3. Advance t by t_c
4. Advance d_1 and d_2 using equations 6 and 7 ($d_2 = d_{2_i} + v_2 t_c$ and $d_1 = d_{1_i} + v_1 t_c$)
5. In the case of a block-on-block collision, update v_1 and v_2 using equations 12 and 13. In the case of a block-on-wall collision, multiply v_2 by -1, to reflect all momentum.
6. Record d_1, d_2, v_1, v_2, and t.
7. If the last collision evaluated was a Block-on-Block collision, calculate t_c according to $t_c =- \frac{d_{2_i}}{v_2}$ (Equation 8, preparation for computing the next Block-on-Wall collision). If the last collision evaluated was a Block-on-Wall Collision, calculate t_c according to $t_c = \frac{d_{2_i} - d_{1_i} + w_2}{v_1 - v_2}$ (Equation 9, preparation for computing the next Block-on-Block collision).
8. Repeat steps 3-7, until $0 \leq v_2 \leq v_1$.

The result of this algorithm will be a list of displacements and velocities, starting with the initial conditions, followed by the displacements, velocities, and time of each collision. We can interpolate to produce data between the recorded points.

Using such an algorithm allows us to experimentally observe the collision count, displacements, and velocities of our cubes, rather than solving for them analytically.

Pseudocode for the Algorithm

The algorithm described in the section above can be expressed formally as IB pseudocode. This representation allows us to easily translate our algorithm to other programming languages.

```
// DATA array stores output
// each element in DATA is an array of form [T, D1, D2, V1, V2]
// T, D1, D2, V1, V2, M1, M2, and W2 are their initial values

// record initial observations
DATAINDEX = 0
DATA[DATAINDEX][0] = T
DATA[DATAINDEX][1] = D1
DATA[DATAINDEX][2] = D2
DATA[DATAINDEX][3] = V1
DATA[DATAINDEX][4] = V2
DATAINDEX = DATAINDEX + 1

// find the time to next block-on-block collision
TC = (D2 + W2 - D1) / (V1 - V2)

// if current simulated collision is Block-on-Block, set to true
ISBLOCKONBLOCK = true

loop until 0 <= V2 <= V1

  // update T
  T = T + TC

  // update displacements
  D1 = D1 + V1 * TC
  D2 = D2 + V2 * TC

  // update V1 and V2 appropriately
  if ISBLOCKONBLOCK then
    V1 = ((M1 - M2) * V1 + 2 * M2 * V2) / (M2 + M1)
    V2 = ((M2 - M1) * V2 + 2 * M1 * V1) / (M1 + M2)
  else
    V2 = -1 * V2
  end if

  // push into DATA
  DATA[DATAINDEX][0] = T
  DATA[DATAINDEX][1] = D1
  DATA[DATAINDEX][2] = D2
  DATA[DATAINDEX][3] = V1
  DATA[DATAINDEX][4] = V2
  DATAINDEX = DATAINDEX + 1

  // calculate TC
```

```
  if ISBLOCKONBLOCK then
    TC = -D2 / V2
  else
    TC = (D2 + W2 - D1) / (V1 - V2)
  end if

  // record next type of collision
  ISBLOCKONBLOCK = NOT ISBLOCKONBLOCK
end loop

output DATA
```

Conclusion

This paper provides two methods of reasoning as to why π arises in the number of collisions between two masses and a wall. It demonstrates the usefulness of configuration spaces in converting numerical problems to geometric ones. Additionally, it provides an algorithmic description of the problem, useful in building simulations and observing and understanding the phenomena itself, through an experimental method, rather than an analytical one.

In a summarized form, there are two answers as to why π arises in the number of collisions between two masses and a wall.

The first answer, outlined in A Kinematic Approach, utilizes the fact that the conservation of energy, when considered with two variable velocities, graphed on a configuration space, is elliptical in nature. Scaling this ellipse to a circle, and subdividing its segments using the conservation of momentum, shows us that for masses m_2, and $m_2 100^n \{n \in \mathbb{Z} \mid n \geq 0\}$, the number of segments on the circle will be equal to $n + 1$ digits of π.

The second answer, outlined in An Optical Approach, utilizes a configuration space, considering two variable displacements, scaled by the roots of the masses of each respective cube. In this system, we can then graph the path of a point, representing the state of our system. This point appears to reflect on a line representing any state where the two masses are colliding, as well as a line where one mass and the wall collides. We can treat these lines as imaginary mirrors, and the path traced by the state of our system as a ray of light. Since a reflection of our ray of light, and the imaginary mirrors is equivalent to a collision, we can count the number of reflections to count the number of collisions. From there, we can find that for masses m_2, and $m_2 100^n \{n \in \mathbb{Z} \mid n \geq 0\}$, the number of reflections is equal to $n + 1$ digits of π.

Computing pi using bouncing cubes is by no means an efficient method. With that being said, it serves as an elegant example of the use of configuration spaces to solve non-trivial problems. This problem also demonstrates the use of simulation to visualize and find bizarre, elegant, and unexpected patterns. While simulation and visualization let us deduce the recurring patterns in the number of collisions in the system above, configuration spaces help us find intuition behind said patterns. Both work in conjunction to build this paper.

Further Research

There are a couple extensions to this problem not covered in this paper. The first is the issue of counting collisions in different bases. The original paper by Galperin addresses this under a footnote, mentioning that for a base b, the mass ratio between cube A and cube B must be b^{2N}, thus for a base of 10, the mass ratio is 100^N.

The second is the issue of accuracy of the estimation of $tan^{-1}(x) = x$ used throughout this essay. Galperin's paper addresses this issue as well. For certain numbers of collisions, the inaccuracy of this estimation leads to errors. This issue is outside of the scope of this essay, however may be used as a starting point for future work.

Works Cited

Galperin, G. (2003). Playing Pool With π (the Number π From a Billiard Point of View). *Regular and Chaotic Dynamics*, *8*(4), 375. https://doi.org/10.1070/rd2003v008n04abeh000252

Freiberger M. (2016). *Physics in a minute: Configuration space.* Plus Magazine. Retrieved August 9, 2022, from https://plus.maths.org/content/physics-minute-configuration-space

Sanderson, G. (2019, February 3). *How colliding blocks act like a beam of light...to Compute Pi.* YouTube. Retrieved March 9, 2022, from https://youtu.be/brU5yLm9DZMS

Sanderson, G. (2019, January 13). *The most unexpected answer to a counting puzzle*. YouTube. Retrieved March 9, 2022, from https://youtu.be/HEfHFsfGXjs

Sanderson, G. (2019, January 20). *Why do colliding blocks compute pi?* YouTube. Retrieved March 9, 2022, from https://youtu.be/jsYwFizhncE