

## CSCI 3010U Assignment 2 Report

### Question 1

$n$  point masses  $m_i$ , where  $i = [1, n]$ . The uniform force is  $F_g = mg$  at these point masses.

$\therefore$  Net Force for a rigid body of  $n$  point masses is:

$$F_{net} = \sum_{i=1}^n F_i = g \sum_{i=1}^n m_i$$

Let  $d_{ik} = \begin{cases} 0, & i = k \\ \text{distance from } m_i \text{ to } m_k, & \text{otherwise} \end{cases}$

where  $i, k = [1, n]$

$\therefore$  Net Torque for a rigid body of  $n$  point masses is:

$$\tau_{net} = \sum_{i=1}^n (F_i \sum_{k=1}^n d_{ik})$$

### Question 2

We care about random processes because sometimes certain simulations can be too costly or very difficult to describe deterministically. Random processes also help us estimate errors and determine a range of results.

### Question 3

#### **Generating Random Initial Variables**

The requirements asked for a radius in  $[0.1, 0.2]$ m, velocity magnitude in  $(0, 10]$ m/s, positions constrained by the box dimensions of  $5 \times 5$ m, and mass in  $[1, 5]$  kg.

Since these dimensions are too small to display in python directly, I multiplied all (except mass) by **100** to display it and set the time step to **0.01** since the velocity would be high for the box dimensions if the time step was 0.1 or higher. More details on how I generated the random numbers are shown below.

**Box Width = 5**

**Box Height = 5**

**Radius in  $[0.1, 0.2]$**

**Position in  $[0, 5]$ :**

1. Position X-Component: Random uniform in  $[0 + \text{Radius}, \text{Box Width} - \text{Radius}]$
2. Position Y-Component: Random uniform in  $[0 + \text{Radius}, \text{Box Height} - \text{Radius}]$

**Velocity in (0, 10]:**

1. Velocity Magnitude:

Random uniform in (0, 10]

2. Velocity Component Sign:

$xComponentSign = (\text{Random choice between } -1 \text{ and } 1)$

$yComponentSign = (\text{Random choice between } -1 \text{ and } 1)$

3. Velocity X-Component:

$xComponentSign \times (\text{Random uniform value in } [0, 1])$

4. Velocity Y-Component:

$yComponentSign \times \sqrt{(Velocity\ Magnitude)^2 - (Velocity\ xComponent)^2}$

**Disk-Wall Collision:**

1. For each disk, a collision is checked between with all 4 walls at each time step. The wall positions to check for collision were set as follows:
  - Left Wall = (0, y-position of disk)
  - Right Wall = (**Box Width**, y-position of disk)
  - Top Wall = (x-position of disk, **Box Height**)
  - Bottom Wall = (x-position of disk, 0)
2. Get the distance between the disk and the wall and check if its less than or equal to the radius of the disk. If so, we get the direction vector of the displacement. If the distance between the wall is greater than the radius of the disk, then we get no collision.
3. Get the relative velocity which would just be the velocity of the disk since the walls are not moving.
4. Get the dot product of the relative velocity and the displacement direction vector. If this value is greater than or equal to 0 we know the disk is moving parallel to the wall or moving away from it.
5. Calculate impulse using the coefficient of restitution, 1 (fully elastic collision), multiplied by the dot product calculated in step 4 above and divided by 1 over mass of disk only. Using impulse, we calculate the velocity after collision.
6. The above steps are repeated for each wall.

**Disk-Disk Collision:**

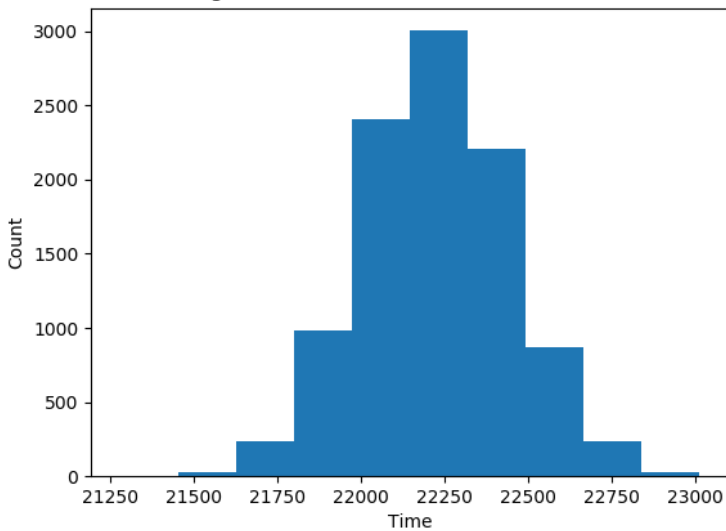
1. For each disk, a collision is checked between it and every other disk. Only two disks are checked at once. This is done after checking wall collisions for each disk.
2. The steps are the same as the Disk-Wall Collision, except the relative velocity will be the difference between the two disks' velocity vectors. The resulting velocity for the second disk is also calculated along with the first disk's. The impulse will also be different because now the divisor will be the sum of the inverses of both disk masses being checked.
3. The above steps are repeated for every disk with every other disk.

**Question 4**

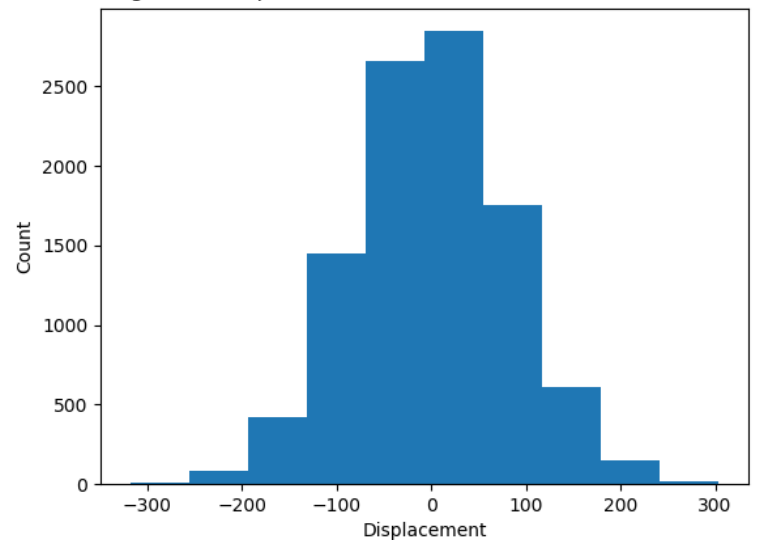
Simulation Ran 10000 times				
Height	Time		Displacement	
	$\mu$	$\sigma$	$\mu$	$\sigma$
10	22.1416	49.60574944	0.0365	6.45316775
100	222.1463	493.89029631	-0.0444	66.74362864
1000	2223.5223	4962.85130271	0.2082	669.80705276
10000	22223.6677	49291.3878767	0.666	6745.305044

The histograms below are for height 10000 where the simulation was run 10000 times.

Histogram of Time Leaf Takes to Reach Ground



Histogram of Displacement of Leaf When it Reaches the Ground



From my observations we get the following correlations:

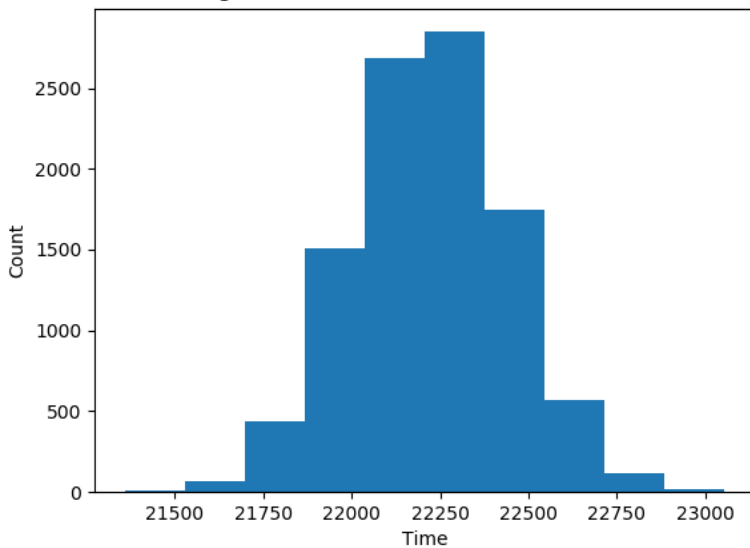
- $time(\mu) \approx 2.22 \times \text{height}$
- $time(\sigma) \approx 4.95 \times \text{height}$
- $displacement(\mu)$  has no correlation to height, but from the histogram and mean value, we see the displacement is mostly near 0. This makes sense because the left and right probabilities are equal.
- $displacement(\sigma) \approx 0.67 \times \text{height}$

We also observe from running the simulation a greater number of times, the correlation of height with time and displacement becomes more accurate.

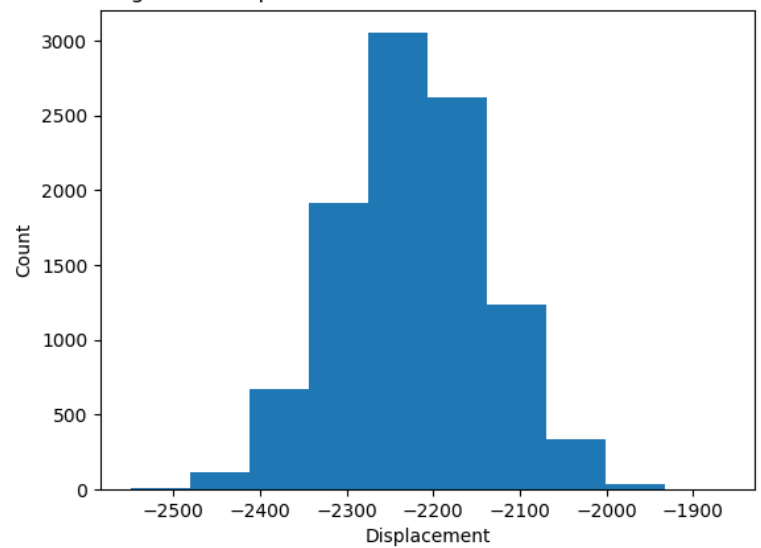
Simulation Ran 10000 times with left P(0.2) and right P(0.1)				
Height	Time		Displacement	
	$\mu$	$\sigma$	$\mu$	$\sigma$
<b>10</b>	22.2103	49.23207391	-2.2645	7.50173975
<b>100</b>	222.4407	490.17748351	-22.1786	76.06370204
<b>1000</b>	2222.5526	4915.37323324	-222.3862	741.77544956
<b>10000</b>	22223.7591	48349.2538672	-2222.6953	7413.05105791

The histograms below are for height 10000 where the simulation was run 10000 times with left and right probabilities of 0.2 and 0.1, respectively.

Histogram of Time Leaf Takes to Reach Ground



Histogram of Displacement of Leaf When it Reaches the Ground



From my observations we get the following correlations which appear to be the similar as left and right probabilities being equal, except for the displacement:

- $time(\mu) \approx 2.22 \times \text{height}$
- $time(\sigma) \approx 4.9 \times \text{height}$
- $displacement(\mu) \approx -0.22 \times \text{height}$
- $displacement(\sigma) \approx 0.75 \times \text{height}$

Increasing left probability to 0.20 and lowering the right one to 0.1 has increased the displacement variance factor by 0.1 and gives a correlation to the mean displacement.