How does g fluctuate around Bay Campus?

Kiran Mahesh Kumar 2331827

This report contains experiments conducted to determine how the values of g fluctuate based on different external conditions. The experiment was conducted with the same mass and controlled distances to ensure a fair test. The use of SUVAT equations and Excel graphing features aided my data analysis.

While some may argue that a stopwatch would be the cheapest and the most straightforward method, the use of a distance sensor instead not only gives us a much more accurate measurement but also can be used as a light gate to determine the exact moment the ball touches the ground which overall improves the accuracy of our readings. The distance sensor itself starts timing when the red button is pressed and stops when an object crosses the sensor. This required me to change the code of my sensor to measure time rather than distance

In summary, the experiment was conducted at sea level (Bay Campus beach), at the highest point on campus (apart from the clock tower), and finally in my room. Each reading of time from a certain height was taken 3 times to ensure maximum accuracy.

Introduction:

Any object of mass will experience a force pulling it toward the ground. The value of g tells us the acceleration of an object under the force of gravity. Free fall refers to when an object is descending from a certain height.

'G' is used in applications such as:

- Mechanics
- Fluid Dynamics
- Geophysics
- Astronomy
- Medical Devices

Why is it important that we have a good knowledge of this value? As you know 9.81 is the value of gravitational acceleration **in a vacuum.** Hence the value will fluctuate based on weather conditions, altitude, tides, etc.

The aim of this experiment is to determine the value of g at different locations on campus and compare how they vary based on the conditions. Bay Campus has a multitude of weather and wind conditions that will show us how the value of g is affected by external conditions

By doing this experiment I wish to determine the best conditions that will give us the closest value for g which can be useful for future projects. While the conditions will be purely qualitative however the data regarding the calculation of g itself will entail the use of quantitative data from the distance sensors and Excel spreadsheets

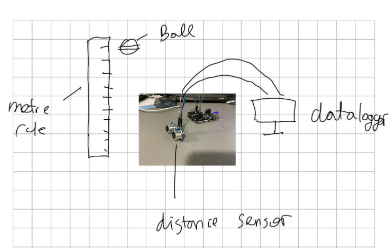
Methodology & Equipment:

- Cricket Ball (160 grams) - The test subject is an object of known weight. This will be dropped in free fall and will be timed to determine its speed and therefore the value of g

- Distance Sensor - Used as a light gate to determine when the ball will hit the ground. The red button will start the clock and it will stop when an object has been detected

- Excel Software - Used to record the data from the data sensor and produce quantitative data for analysis

- Tape Measure/Metre rule - measure the various heights the ball is dropped from



Method

1. Setup apparatus as shown above

2. Choose a sensible distance using your tape measure or metre rule. This distance is the distance you wish to drop the ball from

3. Turn on your Arduino distance sensor and drop the ball

4. A reading will emerge for the time it took for the ball to reach the ground. Type the reading from the serial monitor into Excel software.

5. Repeat this 3 times from the same height and adjust your readings accordingly. Remember to reset the distance sensor before you take each reading

6. Incrementally increase the height of the release of the ball. Repeat steps 2-6 for 5 more increments of 0.20 m

Analysis

As stated previously, the data analysis will be done by Excel via the graphing feature. The way to determine g is via the SUVAT equations. Below are my workings and the formula that will be inputted into Excel to determine the value for g

$$s = ut + \frac{1}{2}at^2$$

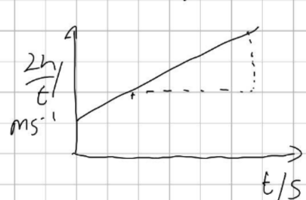
$$2s = 2ut + at^2$$

$$\frac{2s}{t} = 2u + at$$

a in this case will be equivalent to g

$$\therefore \frac{2s}{t} = gt + 2u$$

$$y = mx + c$$

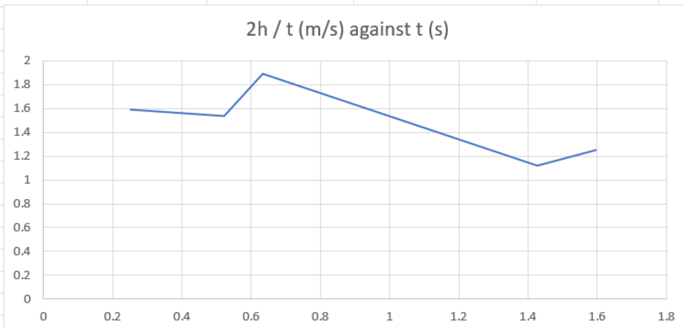


gradient (m) = g

Results

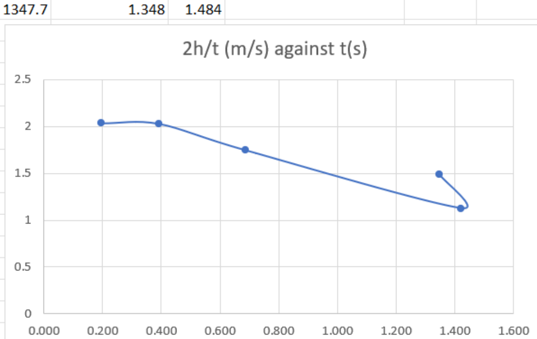
This is in my room where there are no external conditions
G = -0.394

Height (m)	Time 1 (ms)	Time 2 (ms)	Time 3 (ms)	Average Time (ms)	Average Time (s)	2h/t	G value via slope function
0.2	226	232	295	251.00	0.251	1.593625498	-0.394169961
0.4	449	536	579	521.3	0.521	1.534526854	
0.6	676	628	600	634.7	0.635	1.890756303	
0.8	1467	1413	1400	1426.7	1.427	1.121495327	
1	1573	1622	1593	1596.00	1.596	1.253132832	



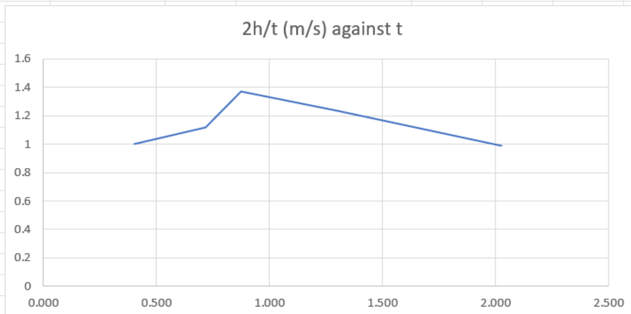
This is at the beach (sea level) on a windy day
G = -0.667

Height (m)	Time 1 (ms)	Time 2 (ms)	Time 3 (ms)	Average Time (ms)	Average Time (s)	2h/t	G value via slope function
0.2	193	185	211	196.3	0.196	2.0374	
0.4	421	445	316	394.0	0.394	2.0305	
0.6	659	611	789	686.3	0.686	1.7484	-0.66636022
0.8	1581	1323	1365	1423.0	1.423	1.1244	
1	1444	1518	1081	1347.7	1.348	1.484	



These results are from the highest flats on campus (Rod Jones Hall)
G = -0.044

Height (m)	Time 1 (ms)	Time 2 (ms)	Time 3 (ms)	Average Time (ms)	Average Time (s)	2h/t	G value via slope function
0.2	326	570	306	400.7	0.401	0.998336	
0.4	761	734	656	717.0	0.717	1.11576	
0.6	838	855	928	873.7	0.874	1.373522	-0.043566095
0.8	1453	1193	1249	1298.3	1.298	1.232349	
1	1917	2010	2158	2028.3	2.028	0.986031	



Discussions & Conclusions

The results themselves are very far from the original value of g which is 9.81. Shown below is a table of all the possible errors that may have occurred during the course of the experiment. Overall, while taking the readings the value seemed close to the true value if the method were to be reproduced elsewhere.

As demonstrated through the readings, the time increases as the height of the drop was increased which makes sense as the ball takes longer to reach the ground. This is the trend for all of my environments where the experiment was conducted.

However, the aim of this experiment was to show how the value of g fluctuates based on different conditions. For the first experiment in my room, the readings seem to have good precision and they don't change much between the first, second, and third drop of the ball. They were precise readings but obviously not accurate. Finally, the result was a value of -0.394 ms^-2

The next experiment was carried out at sea level in windy conditions. In terms of the readings, it is clear that the ball landed on the ground quicker than the experiment in my room as gravity is stronger as you go closer to the earth's core. However, there are a few readings that don't follow the trend for that specific height such as drop 3 from 0.6 m, drop 1 from 0.8m and drop 3 from 1m. This discrepancy may possibly be due to the wind moving the distance sensor.

Furthermore, the flight of the ball is also affected by the wind hence it may have not properly detected the whole ball. Hence we ended up with the result of G being -0.667 which is lower than the value obtained from my room. I was expecting this value to be higher however it seems the discrepancies in the readings did affect the final result. Additionally, this discrepancy is also visible in the graph for this environment

The final experiment was conducted at the highest flats on campus. The readings were what I expected from this height. The ball took longer to reach the ground and this makes sense as gravity gets weaker as you move away from the earth's core. There were a few discrepancies such as drop 1 at 0.8m which may just be due to the distance sensor not detecting the ball properly. Overall the readings were precise however again, not accurate. The value we gained from the readings was -0.044. This is not near the value for g however it is lower than the other 2 values which is expected as stated previously gravity is weaker as you ascend.

Another trend imminent in the readings is there seems to be a big jump between the height of 0.6m and 0.8m. This is due to the human delay in setting the ball from that height and pressing the button on my Arduino. Overall the experiment was conducted as fairly as possible and all the possible errors that may have occurred are shown in the next pages.

Error	Type	How to improve
Uncertainty in the measurement of the initial height of the ball	Parallax	Reve the code for the distance sensor to provide a more accurate reading. Alternatively, I ensured the measurement was taken at eye level and from the centre of the ball
The distance sensor did not pick up the exact time the ball crossed it.	Random	The range of the sensor is 10 cm so I ensured the ball was dropped as close to the sensor as possible
The wind has caused discrepancies and anomalies in the readings	Random	Ensure all equipment is tightly placed and use a heavier ball that will not be affected by the sway of the wind

Time discrepancies. The time between resetting Arduino and dropping the ball introduces human error (Steps 3 and 4)	Systematic / Human	Introduce a quantifiable buffer such as a stopwatch before you release the ball. Additionally, we could introduce a button that is attached externally to the Arduino so the reset is quicker
Calculation and Analysis errors in Excel	Human	Bizarre results indeed. Ensure all values are adjusted properly before analysis and ensure all equations are correct.

References

Garrett, Emily and Duncan, Lindsay. *CGP OCR A Level Physics Revision Guide.* Publisher,. Page 18.

Save My Exams. "Development of Practical Skills in Physics: Precision, Accuracy & Experimental Limitations." Save My Exams, n.d. Web. https://www.savemyexams.com/a-level/physics/ocr/17/revision-notes/1-development-of-practical-skills-in-physics/1-2-handling-data/1-2-9-precision-accuracy--experimental-limitations/

Appendix 1

```
const int trigPin = 3;
const int echoPin = 2;

unsigned long startTime = 0;  // To store the start time when an object is detected
boolean objectDetected = false;  // Flag to indicate if an object is detected

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  long duration;
  int distance;

  // Trigger the ultrasonic sensor to send a pulse
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
```

```
digitalWrite(trigPin, LOW);

// Read the echo pulse duration
duration = pulseIn(echoPin, HIGH);

// Calculate the distance in centimeters
distance = duration / 58;

// Check if an object is detected and record the time
if (distance < 10 && !objectDetected) {
  startTime = millis();  // Record the time when the object is detected
  objectDetected = true;
}

// Check if the object has moved away from the sensor
if (objectDetected && distance > 10) {
  // Print the timestamp when the object crosses the sensor
  Serial.print("Object detected at: ");
  Serial.println(startTime);
  objectDetected = false;
}
```