Measuring Acceleration of Gravity

By Jax Lubkowitz with Lily Kendell

Introduction

Gravity is a force of attraction between all objects. All objects exert gravitational pull on all other objects relative to mass and proximity. However, due to the size of the planet, the gravitational pull of objects around us is indistinguishable and completely overpowered by Earth's gravity[1]. As all physical fields (flying planes to playing sports) are on Earth, the speed at which objects accelerate from Earth's gravity has major implications for all humans. In the absence of air resistance all objects regardless of mass accelerate at the same rate, 9.8 m/s^2, (though in the presence of air resistance their maximum speed will differ due to weight, drag and aerodynamics)[2]. Across the Earth's surface the rate at which objects accelerate can have slight variation depending on location and altitude[3].

In this experiment we will investigate the speed at which an object accelerates to calculate g, the value of gravity in meters per second squared (m/s^2). To achieve this, we used an ultrasonic sensor paired with an Arduino to both measure the distance an object (a flat piece of wood) has fallen and time sense it was dropped from ~1 meters. We plotted the time and distance traveled and then fitted an equation to our data. In doing so we were able to calculate an experimental g for each of our replicates.

Theory –

When an object is dropped it accelerates until it reaches terminal velocity. We operated under the assumption that the only force acting in our experiment was gravity. When gravity is the sole force, the rate at which an object has fallen can be calculated with equation 1.

$$\boldsymbol{y = \frac{1}{2}gt^2 + v_0t + y_0}$$

Equation 1. Distance object has fallen by gravity where y is the distance fallen, g is the acceleration due to gravity, v_0 is the initial velocity, y_0 is the initial height and t is elapsed time.

The theoretical value of g is 9.8 m/s^2. We used the time and distance an object has fallen to derive this equation experimentally. The distance the object has traveled from the dropping point was calculated using equation 2 with a known velocity (of ultrasonic sensor signal: speed of sound at 20C = 343 m/s) and time since drop. With equation 3 we measured our percent difference between our experimental g values and the known value of g.

$$\boldsymbol{Distance = Velocity \times Time}$$

Equation 2. Distance object has moved where velocity is the speed of the object and time is the time since object has started moving.

$$\boldsymbol{diff = \left| \frac{val1 - val2}{\frac{val1 + val2}{2}} \right| \times 100\%}$$

Equation 3. Percent difference formula between 2 values.

Procedure –

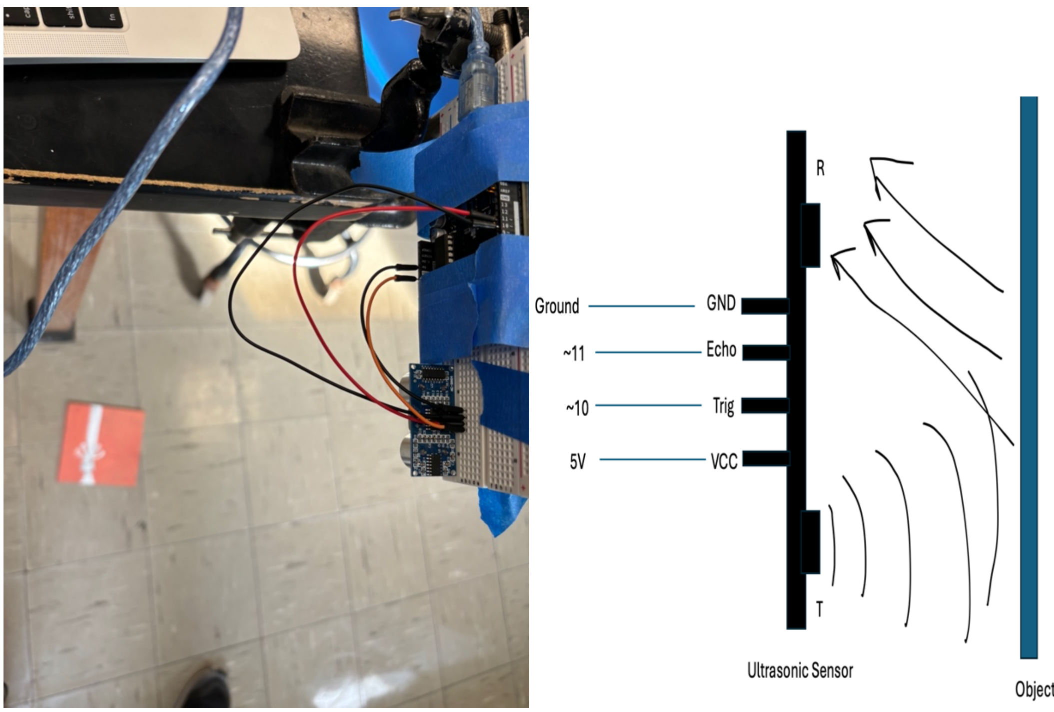

Figure 1. (A) Arduino with ultrasonic sensor set-up attached to table with wooden block below (B) Ultrasonic sensor circuit diagram

We used an ultrasonic sensor with an Arduino to measure the time since, and distance traveled by a dropped wood block object to calculate the value of gravity. The Arduino and sensor were attached to a vice, and a wood block was dropped below the sensor as seen in Figure 1. The ultrasonic sensor was used to measure the distance the object has fallen, and time was measured via the Arduino. The ultrasonic sensor was connected to the Arduino via 4 wires – GND to ground, VCC to 5V, echo to pin 11 and trig to pin 10 as seen Figure 1.B. The ultrasonic sensor emitted a 40 khz signal for 10 microseconds and the time is takes for the signal to bounce off the fallen object and return to the sensor again was collected. This measurement, ping time, is the time it takes for the signal to travel to the object *and back* and thus we divided ping time in half to calculate the time it took the signal to solely reach the object. As the velocity of the signal is known (speed of sound at 20c: 343 m/s) and the time sense the object has fallen (1/2 ultrasonic sensor ping time) the distance the object has traveled was calculated using Equation 2. At each read of the distance the object traveled, the time was printed to the serial monitor with the distance. 100 data points were collected using a for-loop to repeatedly collect the distance and time the object has fallen. The full code can be seen in the appendix. The serial monitor baud rate was set to 500000 for increased accuracy.

Data & Analysis

The data we collected is shown in figures 2 and 3 through two separate drops with time in seconds and distance fallen in meters (see appendix for data tables). Time was measured in microseconds (µs) and converted to seconds (s) by dividing by $1x10^6$. Distance was calculated from ping time to meters using Equation 2 using the speed of sound and ½ ping time.

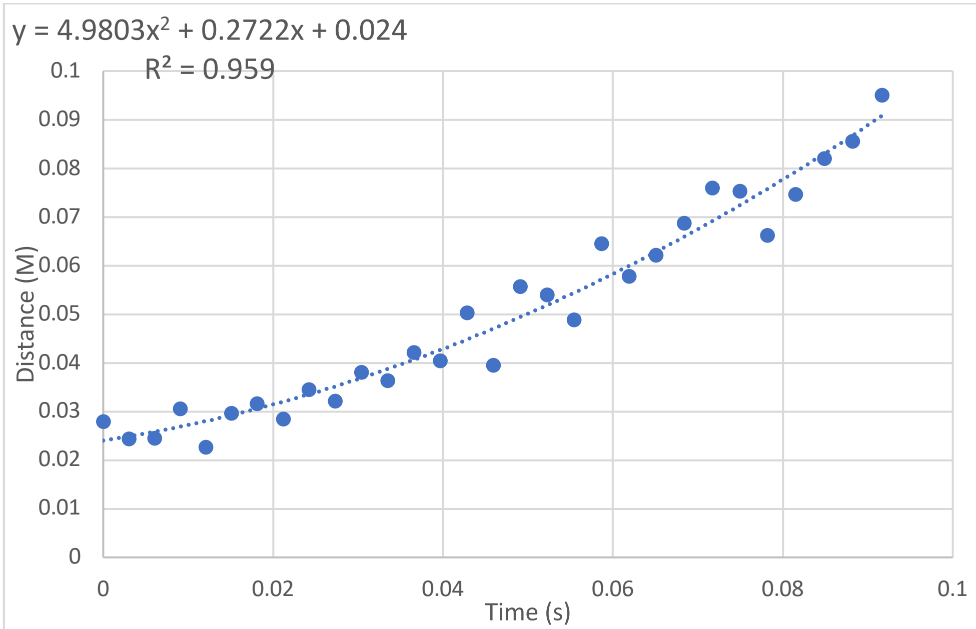

Figure 2. Acceleration of wood block through air as a function of meters per second with excessive noise in data due to misalignment of Arduino apparatus. Fitted polynomial equation of $Y = 4.9803X^2 + 0.2722X + 0.024$ ($R^2 = 0.959$).

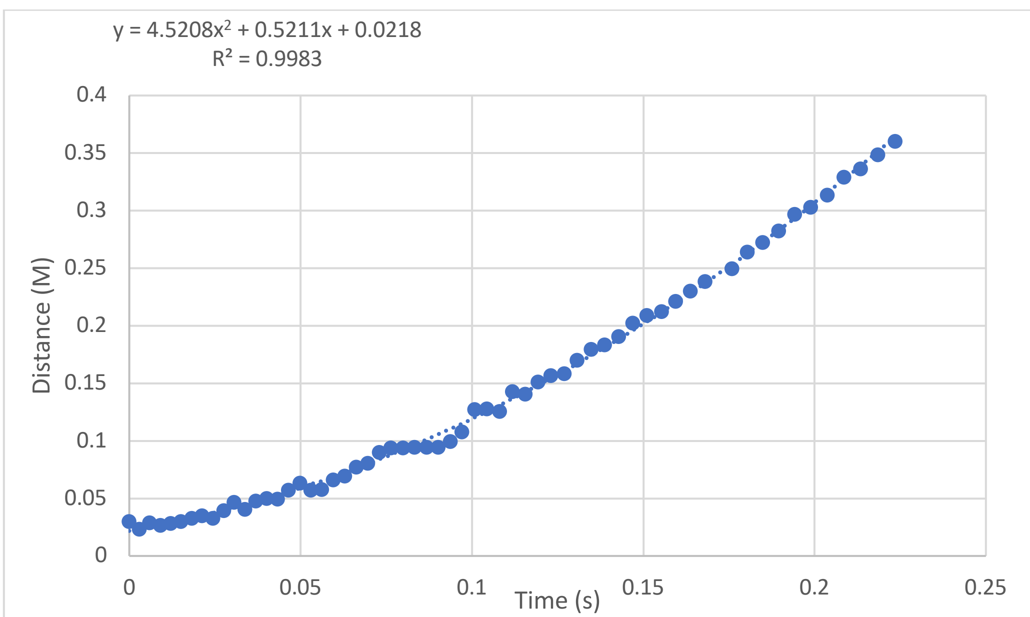

Figure 3. Acceleration of wood block through air as a function of meters per second using Arduino apparatus. Fitted polynomial equation of $Y = 4.5208X^2 + 0.5211X + 0.0218$ ($R^2 = 0.9983$).

With the distance the object has fallen and time since it has fallen, we used a polynomial model to measure the coefficients at which the object accelerated to calculate an experimentally measured g-value. In both figures we fitted an equation to each relationship and as we plotted time on the horizontal axis and distance on the horizontal access, we rewrote each trendline in terms of g. Then we substituted the fitted coefficients into Equation 1 and then calculated g for the Figure 2 data.

$$y = 4.9803t^2 + 0.2722t + 0.024$$

$$4.9803t^2 = \frac{1}{2}gt^2$$

$$g = 9.9606 \text{ m}/s^2$$

Using the known value of g, 9.8 m/s^2 we calculated our values precent difference using Equation 3.

$$diff = \left| \frac{9.8 \text{ m}/s^2 - 9.9606 \text{ m}/s^2}{\frac{9.8 \text{ m}/s^2 + 9.9606 \text{ m}/s^2}{2}} \right| x\ 100$$

$$diff = 1.625\%$$

This same process was performed on the results from Figure 3 (see below).

$$g = 9.0416 \text{ m}/s^2$$

$$diff = 8.05\%$$

Each of our 2 drops collected 100 data points but as the object is no longer accelerating after hitting the ground we excluded the later points from our models to prevent their influence in coefficients. Furthermore, as the object got farther from the ultrasonic sensor the accuracy decreases further discouraging the use of later data points. In addition, we increased the baud rate to 500000 to increase the accuracy of the timing for the most precision. In both cases the V_o values of our experimental equations were close to 0 which is to be expected as the objects initial speed was 0.

One of our experimental values was above the expected value for gravity and the other below. This could be explained by interference of our sound waves hitting other objects like the side of the table causing increased noise in readings (as seen in the jitter of Figure 2). Furthermore, human error could have been introduced by the way in which the dropped wood object was released causing a non-level surface to measure on the descent resulting in less accurate readings. The human release of the object made this experiment hard to reproduce. In addition, our calculated values could be different from the expected 9.8 m/s^2 due to our altitude and location though this much variation is unlikely.

Conclusions

We measured the value of gravity to be 9.9606 m/s^2 and 9.0416 m/s^2 which is 1.625% and 8.05% difference, respectively, then the expected value of 9.8 m/s^2. Together our results have an average difference of 4.83% which is reasonably close to, and thus supports, the theoretical value. The slight differences in our experiment values could have arisen from various factors.

The theoretical value of acceleration, 9.8 m/s^2, and equation we used to calculate our experimental values assumes there is no other forces at play. However, we dropped a wide flat wood block that has a reasonable amount of drag (an outside force) which could lead to skewed results. In addition, as we only used a subset of the data points measured (the points where the object was still accelerating and close enough to measure accurately) accidently inclusion of too much or too little of our data could have influenced our experimental values. Due to the large human component of this experiment when dropping the wood object and starting the code simultaneously this experiment is hard to reproduce. There was a wide spread of drop quality from user error of mistiming the drop, to object rotating or falling out of sensors path to other interference in sensor readings. In Figure 2 we see an increased amount of variability in the points which is not present in Figure 3 where we had shifted the Arduino farther away from the table. This indicates some of the signal was being picked up off the table and not just the dropped object. In addition, the use of an ultrasonic sensor to measure the distance introduces some variability into our experiment as the way in which the object falls could impact where we measure and thus our distance calculations.

To obtain more accurate results we could perform more drop rounds and average our experimental g-values to reduce the effects of outliers and bad drops. Further steps could be taken to ensure less noise in the sensor readings by eliminating near obstacles such as the side table. In addition, automation of the dropping process could achieve more consistent and level drops further eliminating any influence of human error.

(1) Thompson, Hobie, and Sarah Havern. *Gravity*, Stanford, web.stanford.edu/~buzzt/gravity.html. Accessed 10 Apr. 2024.
(2) “Free Fall.” *Wikipedia*, Wikimedia Foundation, 25 Feb. 2024, en.wikipedia.org/wiki/Free_fall.
(3) Carolina.Biological. “Earth’s Gravitational Field.” *Carolina Knowledge Center*, 27 Nov. 2023, knowledge.carolina.com/carolina-essentials/earths-gravitational-field/.

Appendix –

Free Fall Lab Code

```
FreeFallLab
unsigned long ping_time; // value to read in pulse
int trigpin = 10; // trig pin to arduino
int echopin = 11; // echo pin to arduino
unsigned long falltime; // unsigned long to get precise microseconds

void setup() {
  pinMode(trigpin, OUTPUT); // initialize trig pin
  pinMode(echopin, INPUT); // initialize echo pin
  Serial.begin(500000); // set high baud rate for increased accuracy
}

void loop() {
  Serial.println("Ready?"); // Prompt user
  while (Serial.available() < 1){} // Wait for user to enter something

  for(int i =0; i<100 ; i++){ // capture 100 data points of object falling
    digitalWrite(trigpin, HIGH); // send out ultrasonic sensor
    delayMicroseconds(10); // delay for 10
    digitalWrite(trigpin,LOW); // turn off out signal
    ping_time = pulseIn(echopin, HIGH); // read in how long it takes for signal to return
    falltime = micros(); // get time measurement
    // print tab seperated results to output
    Serial.print("\t");
    Serial.print(falltime);
    Serial.print("\t");
    Serial.println(ping_time);
    Serial.read();
  }
}
```

Figure 2 Data.

Time (s)	Distance (M)
0	0.0279545
0.003004	0.024353
0.006012	0.0245245
0.009052	0.030527
0.012048	0.022638
0.015084	0.0296695
0.018128	0.031556
0.021156	0.028469
0.02422	0.0344715

0.027272	0.0320705
0.03036	0.038073
0.033436	0.036358
0.036548	0.042189
0.039648	0.040474
0.042808	0.0502495
0.045904	0.039445
0.049092	0.0557375
0.052268	0.0540225
0.055412	0.0488775
0.058652	0.064484
0.061856	0.0577955
0.065084	0.062083
0.068356	0.0687715
0.07166	0.0759745
0.074956	0.0752885
0.078208	0.066199
0.081508	0.0746025
0.084856	0.081977
0.088216	0.0855785
0.091636	0.095011

Figure 3. Data

Time (s)	Distance (M)
0	0.030184
0.003024	0.0231525
0.006076	0.0289835
0.009116	0.026754
0.012168	0.0286405

0.015228	0.030184
0.0183	0.0327565
0.021388	0.0351575
0.024464	0.0327565
0.02758	0.039788
0.030736	0.0464765
0.03386	0.040474
0.037028	0.047677
0.040208	0.050078
0.043388	0.049392
0.046612	0.057281
0.049876	0.063455
0.053104	0.0574525
0.056324	0.0577955
0.0596	0.066199
0.062896	0.0698005
0.06624	0.077175
0.0696	0.0807765
0.07302	0.090209
0.07646	0.0938105
0.0799	0.0938105
0.083344	0.0948395
0.086792	0.0944965
0.090232	0.0944965
0.093708	0.099813
0.097228	0.107702
0.100856	0.127253
0.104496	0.1281105

0.108116	0.125538
0.111848	0.143031
0.115564	0.14063
0.119328	0.151263
0.123136	0.156751
0.126956	0.158466
0.130844	0.1699565
0.134788	0.1795605
0.138796	0.1836765
0.142856	0.1908795
0.146984	0.2021985
0.151156	0.2090585
0.155344	0.21266
0.15958	0.2210635
0.163864	0.230496
0.168204	0.238728
0.17594	0.249704
0.18042	0.2639385
0.18496	0.2725135
0.189556	0.282632
0.194236	0.297038
0.198948	0.302869
0.203728	0.313845
0.2086	0.32928
0.213508	0.336483
0.218492	0.348488
0.223544	0.360493