# Cart on an Inclined Plane Lab

Report written by Joseph Farah
Partners: Georgia, Lincoln, Ning

*ABSTRACT*

The point of this lab was to determine the motion of a cart down an inclined plane. To collect the data necessary for the experiment, we constructed a ramp from a piece of (plywood?) on a low stack of books and rolled a cart down it. The data was collected by way of a ticker timer recording the position in time of the cart at 60 times a second (60 Hz). The result was the cart appeared to traverse the full distance of the ramp (109 cm) in little more than a second, while the rest of the data was possibly just garbage values from after the cart left the ramp. If this is not true, than the cart had a wildly varying acceleration and velocity, but somehow maintained a relatively linear distance-traveled graph.

*OBJECTIVE*

The point of this lab will be to determine the motion of a cart down an inclined plane, and using the data gathered to determine if the cart moves with constant/changing velocity/acceleration.

*HYPOTHESIS*

The cart will start off slowly, gathering speed and momentum as it heads down the ramp. It will have a positive acceleration and an increasing velocity.

*PROCEDURE*

The materials needed are as follows: several books, a cart, a wooden ramp, some tape, several meters of ticker tape, and a ticker timer. The procedure is as follows:

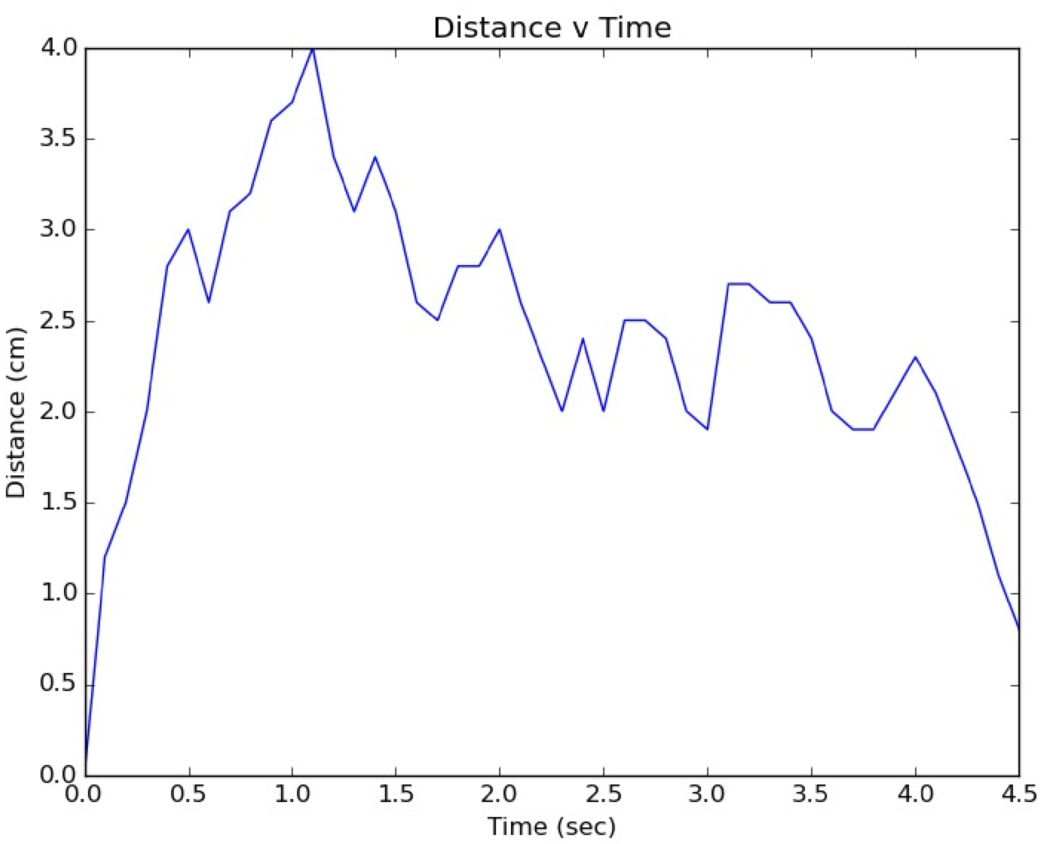
1. Place the wooden ramp on a stack comprised of several books to create a gradual incline.
2. Attach one end of the ticker tape to the back end of the cart with the tape.
3. Feed the other end of the ticker tape through the ticker timer, positioning both so that when the cart pulls the ticker tape, it passes through the timer.
4. Start the ticker timer.
5. Give the cart a gentle shove, enough to get it going but not enough to overly contribute to its original momentum.
6. Mark off every six dots created by the ticker timer.
7. Measure the distance between the dots.

*DATA AND OBSERVATIONS*

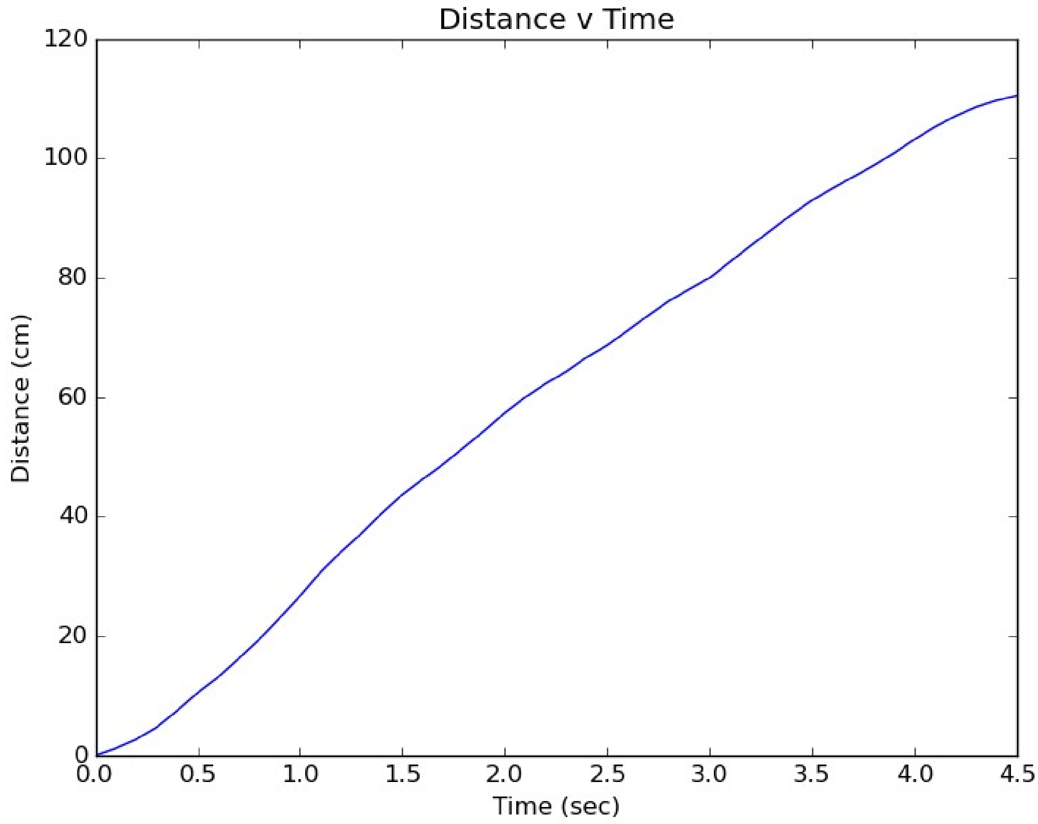
Distance Traveled vs. Time

| Time (sec) | Length (cm) |
|---|---|
| 0.0 | 0.0 |
| 0.1 | 1.2 |
| 0.2 | 1.5 |
| 0.3 | 2.0 |
| 0.4 | 2.8 |
| 0.5 | 3.0 |
| 0.6 | 2.6 |
| 0.7 | 3.1 |
| 0.8 | 3.2 |
| 0.9 | 3.6 |
| 1.0 | 3.7 |
| 1.1 | 4.0 |
| 1.2 | 3.4 |
| 1.3 | 3.1 |
| 1.4 | 3.4 |
| 1.5 | 3.1 |
| 1.6 | 2.6 |
| 1.7 | 2.5 |
| 1.8 | 2.8 |
| 1.9 | 2.8 |
| 2.0 | 3.0 |
| 2.1 | 2.6 |
| 2.2 | 2.3 |
| 2.3 | 2.0 |
| 2.4 | 2.4 |
| 2.5 | 2.0 |
| 2.6 | 2.5 |
| 2.7 | 2.5 |
| 2.8 | 2.4 |
| 2.9 | 2.0 |
| 3.0 | 1.9 |
| 3.1 | 2.7 |
| 3.2 | 2.7 |
| 3.3 | 2.6 |
| 3.4 | 2.6 |
| 3.5 | 2.4 |
| 3.6 | 2.0 |
| 3.7 | 1.9 |
| 3.8 | 1.9 |
| 3.9 | 2.1 |
| 4.0 | 2.3 |
| 4.1 | 2.1 |
| 4.2 | 1.8 |
| 4.3 | 1.5 |
| 4.4 | 1.1 |

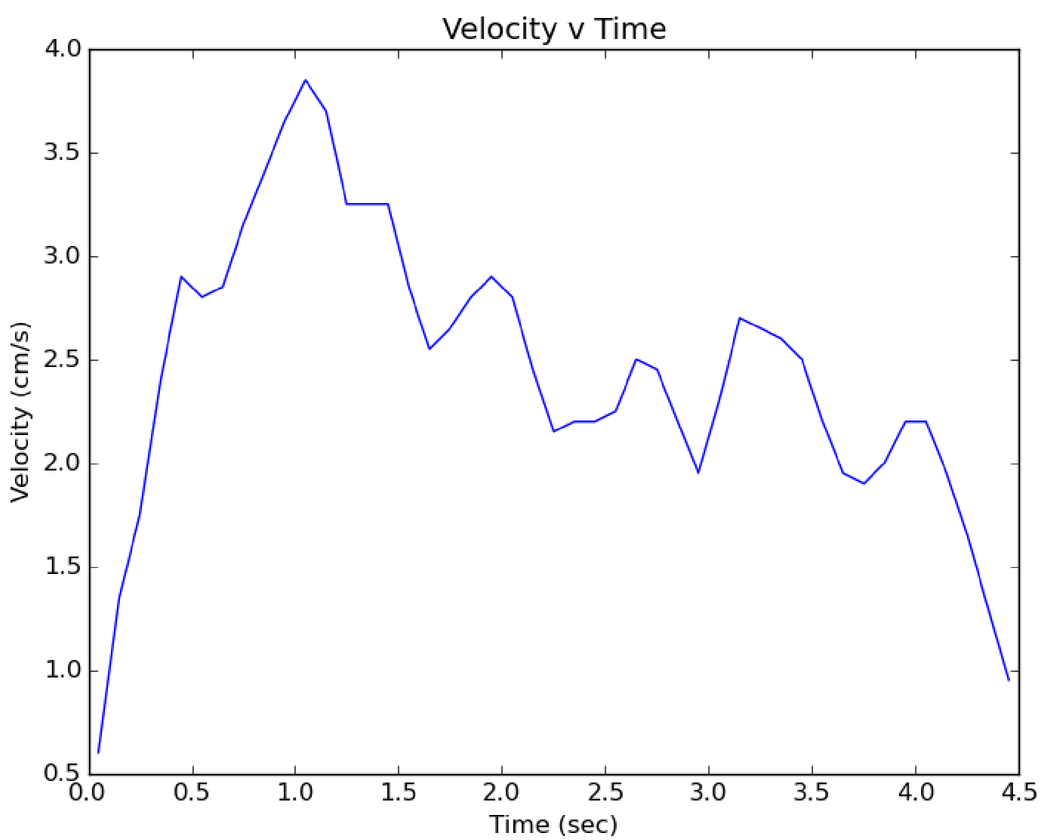
DISTANCE v TIME (distance from dot to dot vs time)



TOTAL DISTANCE v TIME (net distance, increasing from dot to dot, assuming positive motion)




VELOCITY v. TIME (instantaneous velocity at each point versus time)



*DATA ANALYSIS*

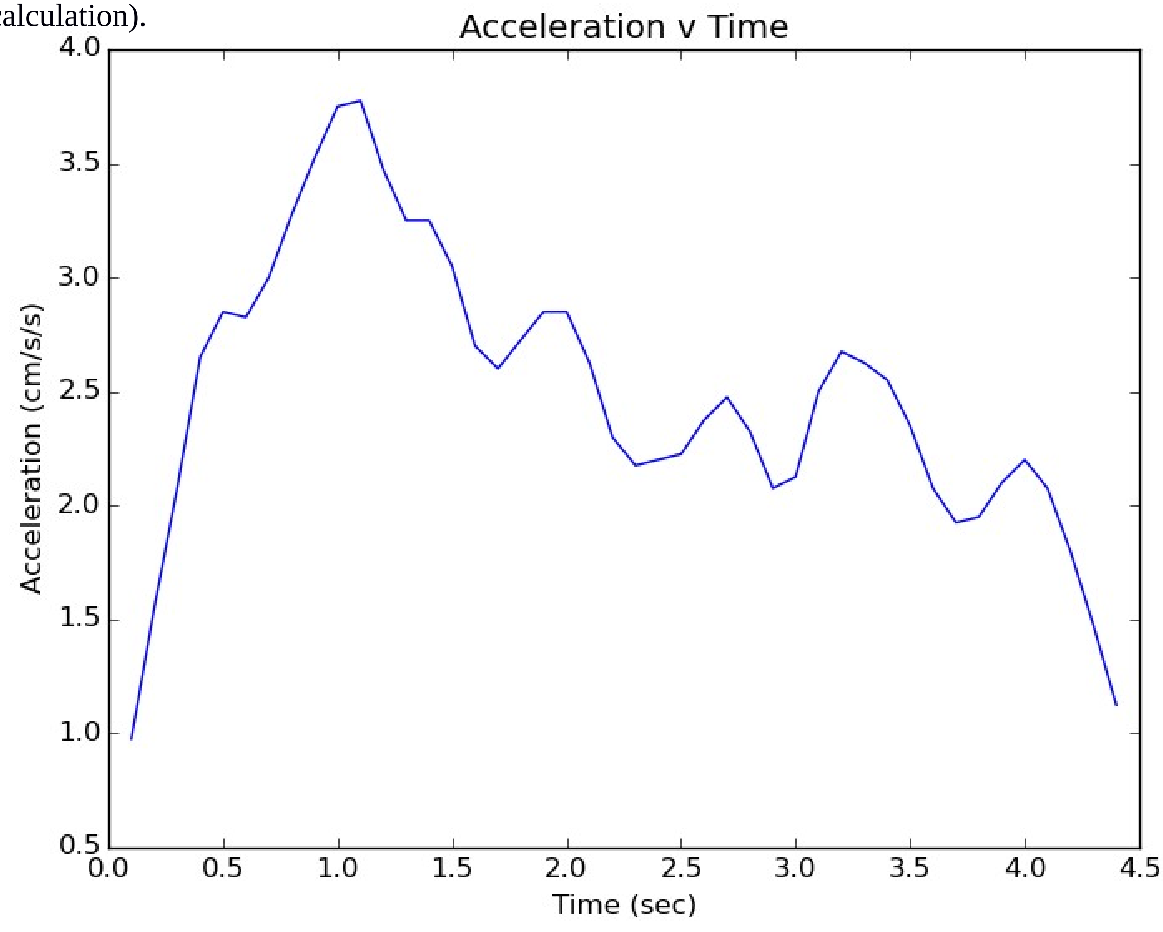
The table has 45 data points, from 0 to float(44/10) in increments of 1, labeled with the corresponding distance value. As we measured every six ticker timer dots, we measured the distance from point to point, and not from the beginning.

The first graph is non-net distance versus time. It directly reflects the data given in the table, without adding up the values along the way. While this makes it easier to calculate velocity versus time, it does not give us much useful information.

The second graph is a net distance versus time plot. It adds up the distance as the graph progresses, which shows the total distance covered by the cart, not just from point to point, but over the whole time traveled. This shows us plenty of useful information, allowing us to infer velocity and acceleration just by looking at the graph.

The final graph shows velocity versus time. The data points were calculated by taking the averages of the distances (non net) and time values between each two points. The result is a mountain-shaped graph, remarkably similar in appearance to the first plot. If we had used larger increments of time, the graph would have undoubtedly had a less jagged persona, but a trend is somewhat clear—in the first second, the velocity rockets up to its maximum of about $3.8 m.s^{-1}$, but then gradually slows for the rest of the time period. Despite the fact that the cart was on an negatively inclined plane (relative to the horizon), the cart slows down as it travels farther. This suggests several things. First, I obviously delivered a much harder shove that I thought to get the cart going. Second, friction between the carts wheels and the not-so-smooth ramp may have led to the rapid decrease in speed. Finally, a more logical conclusion to draw would be that the graph does not in fact accurately represent the cart's motion. It is bordering on unbelievable that a cart would take more than *four seconds* to traverse the distance of the ramp; a little more than one second seems more likely. This conclusion is, in fact, supported by the graph—the cart's velocity increases as expected in the first 1.1 seconds before inexplicably slowing down.

The cart's acceleration throughout the four seconds is somewhat wild, but measurable. A graph of the cart's acceleration based off of the velocity versus time graph is shown below (see Appendix A for calculation).



Wherever the acceleration increases, the velocity has a corresponding increase, along with a spike in the distance. Whenever the acceleration decreases, the velocity has a corresponding decrease, along with a dip in the traversed distance.

Using the formula $(delta\, d) = v_1 t + \frac{1}{2} at^2$ we can figure that the figure that the approximate distance traveled was 14.4 cm (in the first four seconds, the only time with relatively constant acceleration). According to the net distance graph, the actual distance is a little more than 20 cm. This gives a percent difference of 42%, which is significantly different.

The total area under the velocity versus time graph is 109.45 cm. The actual traveled distance (as measured) is 109 centimeters, which yields a percent difference of 0%, with no significant difference (see Appendix A for calculation).

The more accurate method for calculating the distance of the cart is plainly the integral method. This takes into account the subtle—or sometimes drastic—changes in the velocity, whereas the distance formula cannot. The result is less accuracy for the formula and pinpoint precision for the graph calculation.

If we assume that the actual cart-traveling-down-the-ramp took place in the first second, then the hypothesis is supported. The cart did gather speed, but then that speed changed drastically after it left the ramp, leading to the weird shapes in the latter three-quarters of the velocity versus time graph.

*CONCLUSION*

The cart (which moved with a swiftly changing velocity/acceleration) traveled about 109 cm down a wooden ramp. The data yields velocity, distance, and acceleration graphs with mountainous shapes, showing the wildly varying accelerations and velocities. The hypothesis was somewhat supported. This has interesting contexts, especially for topics like alternative methods for space travel —a good algorithm (see Appendix A) for calculating and comparing velocities and accelerations could come in handy when, for example, comparing the potentials of solar sails versus chemical propulsion.

*ERROR ANALYSIS*

There were many limitations to this lab. I do not trust the data we collected for several reasons. First of all, the ticker timers were old and no doubt were not vibrating at the proper frequencies. Secondly, after measuring about 40 tiny dots on a slim piece of paper, one's eyes begin to get blurry and their hands shaky. Finally, I measured and recorded a good three quarters of the ticker tape, which means that it would NOT hold up in a court of law, thanks to my famously terrible estimation skills and horrifyingly hideous handwriting.

APPENDIX A

This section contains all the code that I used to complete the calculations and graphs.

*Creating the Distance V. Time graphs*

```
#!/usr/bin/python

import matplotlib.pyplot as plt
import os

try:
	os.system("rm -f table.png")
except:
	pass

data = 'data.txt'
target = open(data, 'r+w')
lines = target.readlines()
x_values = []
y_values = []
labels = []
values = []
for line in lines:
	print line
	if line[0]=="#":
		labels.append(line)
		pass
	else:
		if line[0]=='=':
			y_value_begins = lines.index(line) + 1
			break
		else:
			num = line.rstrip()
			print num
			x_values.append(float(num))
for i in range(y_value_begins,len(lines)-1):
	print line
	if lines[i][0]=='=':
		break
	else:
		num = lines[i].rstrip()
		print num
		y_values.append(float(num))

print x_values
print '\n'
print y_values
print len(x_values), len(y_values)
final_x_values = []
```

```
final_y_values = []
for i in range(len(x_values)-1):
      values.append([x_values[i],y_values[i]])
colLabels=('Time (sec)', 'Length (cm)')
hcell, wcell = 0.3, 4.
hpad, wpad = 0, 0
nrows, ncols = len(values)+1, len(colLabels)
fig=plt.figure(figsize=(ncols*wcell+wpad, nrows*hcell+hpad))
ax = fig.add_subplot(111)
ax.xaxis.set_visible(False)
ax.yaxis.set_visible(False)
the_table = ax.table(cellText=values,
          colLabels=colLabels,
                  loc='center')
plt.title("Distance Traveled vs. Time")
plt.savefig("table.png")
```

*Creating the Velocity and Acceleration v. Time graphs*

```
import matplotlib.pyplot as plt
import os

try:
      os.system("rm -f dvt_graph.png")
except:
      pass

def integrate(y_vals, h):
    i=1
    total=y_vals[0]+y_vals[-1]
    for y in y_vals[1:-1]:
        if i%2 == 0:
            total+=2*y
        else:
            total+=4*y
        i+=1
    return total*(h/3.0)

data = 'data.txt'
target = open(data, 'r+w')
lines = target.readlines()
x_values = []
y_values = []
velx = []
vely = []
accely = []
accelx = []
labels = []
values = []
```

```
for line in lines:
      print line
      if line[0]=="#":
            labels.append(line)
            pass
      else:
            if line[0]=='=':
                  y_value_begins = lines.index(line) + 1
                  break
            else:
                  num = line.rstrip()
                  print num
                  x_values.append(float(num))
for i in range(y_value_begins,len(lines)-1):
      print line
      if lines[i][0]=='=':
            break
      else:
            num = lines[i].rstrip()
            print num
            y_values.append(float(num))
y_values2 = []
y_values2.append(y_values[0])
for i in range(1, len(y_values)):
      y_values2.append(y_values2[i-1]+y_values[i])
print len(x_values), len(y_values2)
for i in range(len(x_values)-1):
      velx.append((x_values[i]+x_values[i+1])/2)
for i in range(len(y_values)-1):
      vely.append((y_values[i]+y_values[i+1])/2)
for i in range(len(velx)-1):
      accelx.append((velx[i]+velx[i+1])/2)
for i in range(len(vely)-1):
      accely.append((vely[i]+vely[i+1])/2)
x = accelx
y = accely
area = integrate(vely,1)
print area
plt.plot(x,y)
plt.xlabel('Time (sec)')
plt.ylabel('Acceleration (cm/s/s)')
plt.title('Acceleration v Time')
# plt.show()
plt.savefig("accelvtime_graph.png")
```

*Raw Data, text file format, in file* “data.txt”

```
#TIME(SEC)#
#LENGTH(CM)#
=
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2.0
2.1
2.2
2.3
2.4
2.5
2.6
2.7
2.8
2.9
3.0
3.1
3.2
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4.0
4.1
4.2
4.3
4.4
=
0.0
1.2
1.5
2.0
2.8
3
2.6
3.1
3.2
3.6
3.7
4
3.4
3.1
3.4
3.1
2.6
2.5
2.8
2.8
3
2.6
2.3
2
2.4
2
2.5
2.5
2.4
2.0
1.9
2.7
2.7
2.6
2.6
2.4
2.0
1.9
1.9
2.1
2.3
2.1
1.8
1.5
1.1
.8
=
```