

Modeling a hanging slinky

Measured quantities:

(MKS units)

equilibrium data

```
In[*]:= mass = 212.0 / 1000;  
        length = 55.4 / 1000;  
        density = mass / length;  
        coils = 84;  
        width = length / coils;
```

height data

```
In[*]:= dataraw = {0, 3+5/8, 5+13/16, 8+5/8, 11, 13+3/8, 15+9/16,  
                  17+3/4, 19+7/8, 21+3/4, 23+3/4, 25+9/16, 27+3/8, 29+1/8, 30+3/4,  
                  32+1/2, 34, 35+1/2, 36+7/8, 37+9/16, 38+7/8, 39+7/8, 40+1/2,  
                  41+3/4, 42+7/8, 43+7/8, 44+13/16, 45+3/4, 46+1/2, 47+1/4,  
                  47+7/8, 48, 48+1/2, 49, 49+1/2, 50, 50+3/8, 50+3/4, 51, 51+5/16};
```

Notes on data collection:

1. The data is given in inches measured downward from the top point of the hanging slinky.
2. Each data point is the height of the even numbered coils starting with the top coil indexed at zero, i.e., the height of every other coil is measured.
3. The presumed accuracy is 1/16 of an inch.
4. The last measurement 51+5/16 is for the bottom point and there were approximately six coils, all essentially “bunched together” in almost equilibrium configuration between the bottom and the previously measured height at 51 inches.

5. The slinky winds tighter as it is extended.

QUESTION: What is the physical/mathematical explanation for this?

```
In[*]:= mksdataraw = -2.54 dataraw / 100;
```

Let's examine the measured data and see if we can put it in a more useable form.

```
In[*]:= Length[dataraw]
```

```
Out[*]= 40
```

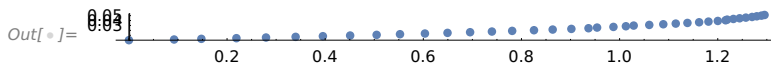
This seems slightly inconsistent with data collection notes 2 (that the height of every other coil is measured) and 4 (that there were six coils left out at the end). Given 84 coils, one might expect 42 possible data points. On the other hand, assuming the last data point gives the height of the bottom of the slinky, and counting this as one of the six “bunched” coils this would mean precisely two of the expected measurements were not made which is precisely what we have, and we should add two data points at the end. Let us proceed on this assumption.

A horizontal layout or “sideways plot” should be easier to fit on the page. I will also plot the data with the extension to the right.

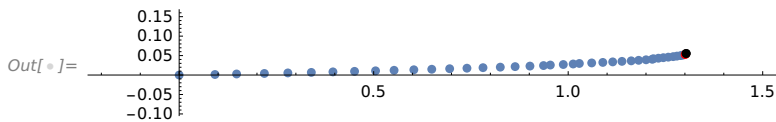
```
In[*]:= step[a_, b_, m_, j_] = a + j (b - a) / m;
```

```
In[*]:= startdata = Table[{-mksdataraw[[j + 1]], step[0, length, 41, j]}, {j, 0, 38}];
missingdata = {{-mksdataraw[[40]] - 4 width, step[0, length, 41, 39]},
               {-mksdataraw[[40]] - 2 width, step[0, length, 41, 40]}};
enddata = {{-mksdataraw[[40]], length}};
```

```
In[*]:= startplot = ListPlot[startdata, AspectRatio → Automatic, PlotRange → All]
```



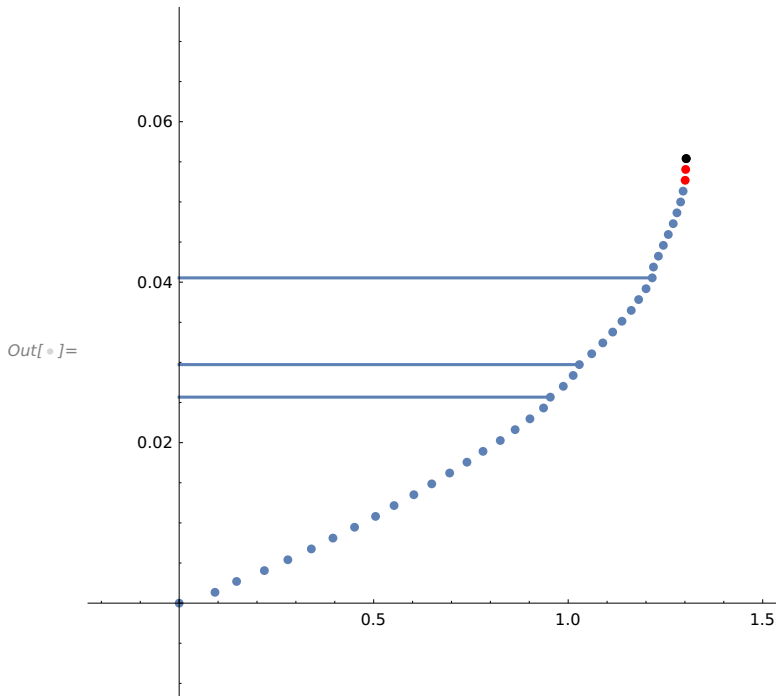
```
In[*]:= Show[startplot, ListPlot[missingdata, PlotStyle → Red],
              ListPlot[enddata, PlotStyle → Black],
              PlotRange → {{-0.2, -mksdataraw[[40]] + 0.2}, {-0.1, length + 0.1}}]
```



```

In[ ]:= Show[ListPlot[startdata], ListPlot[missingdata, PlotStyle -> Red],
  ListPlot[enddata, PlotStyle -> Black],
  Plot[step[0, length, 41, 19], {x, 0, -mksdataraw[[20]]},
  Plot[step[0, length, 41, 22], {x, 0, -mksdataraw[[23]]},
  Plot[step[0, length, 41, 30], {x, 0, -mksdataraw[[31]]},
  PlotRange -> {{-0.2, -mksdataraw[[40]] + 0.2}, {-0.01, 0.07}}, AspectRatio -> 1]

```



Corrections and Data Manipulations

Most of the data looks pretty good at this point. There are three points marked above that look a bit off. The first two look like they are more or less within tolerance but perhaps were the result of an oscillation in the position of the slinky or just a slightly misread measurement. The last one (point number 31) looks like a copying error. Looking at that section of the raw data I see

..., $47 + 7/8$, 48, $48 + 1/2$, 49, ...

I'll guess that actually point number 31 is correct, but the previous point corresponding to 48 in the raw data is a miscopy/miscommunication. So the error is actually in point number 30 rather than the one which is marked. Point number 30 should not be there (I guess).

This leaves us with an apparently missing third point.

Here is an explanation for that: Of the enumerated and marked 84 coils, there were two coils at the top of the slinky which are tied together and consequently prevented from spreading by a ferule.

Thus points coils number 1 and 2 correspond to the same measured point.

According to these hypotheses corrected data for the effective 83 coils with 42 expected data points is as follows:

```
In[*]:= correcteddataraw =
```

```
{0, 3 + 5 / 8, 5 + 13 / 16, 8 + 5 / 8, 11, 13 + 3 / 8, 15 + 9 / 16, 17 + 3 / 4, 19 + 7 / 8, 21 + 3 / 4,
 23 + 3 / 4, 25 + 9 / 16, 27 + 3 / 8, 29 + 1 / 8, 30 + 3 / 4, 32 + 1 / 2, 34, 35 + 1 / 2, 36 + 7 / 8,
 37 + 9 / 16, 38 + 7 / 8, 39 + 7 / 8, 40 + 1 / 2, 41 + 3 / 4, 42 + 7 / 8, 43 + 7 / 8, 44 + 13 / 16, 45 + 3 / 4,
 46 + 1 / 2, 47 + 1 / 4, 47 + 7 / 8, 48 + 1 / 2, 49, 49 + 1 / 2, 50, 50 + 3 / 8, 50 + 3 / 4, 51, 51 + 5 / 16};
```

```
In[*]:= correctedmks = -2.54 correcteddataraw / 100;
```

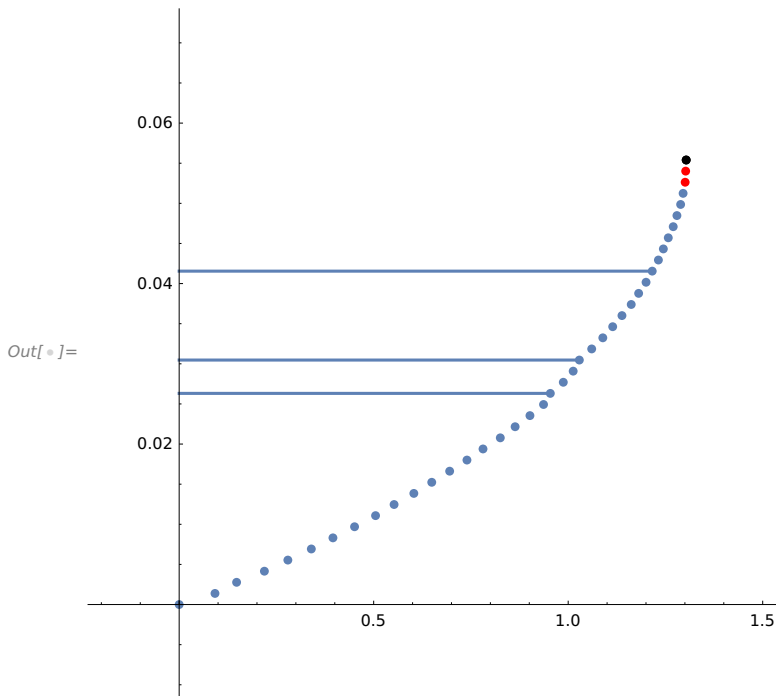
```
In[*]:= Length[correctedmks]
```

```
Out[*]= 39
```

```
In[*]:= correctedstartdata = Table[{-correctedmks[[j + 1]], step[0, length, 40, j]}, {j, 0, 37}];
correctedmissingdata = {{-correctedmks[[39]] - 4 width, step[0, length, 40, 38]},
  {-correctedmks[[39]] - 2 width, step[0, length, 40, 39]}};
correctedenddata = {{-correctedmks[[39]], length}};
```

```
In[*]:= correctedplot =
```

```
Show[ListPlot[correctedstartdata], ListPlot[correctedmissingdata, PlotStyle → Red],
  ListPlot[correctedenddata, PlotStyle → Black],
  Plot[step[0, length, 40, 19], {x, 0, -correctedmks[[20]]},
  Plot[step[0, length, 40, 22], {x, 0, -correctedmks[[23]]},
  Plot[step[0, length, 40, 30], {x, 0, -correctedmks[[31]]},
  PlotRange → {{-0.2, -correctedmks[[39]] + 0.2}, {-0.01, 0.07}}, AspectRatio → 1]
```



Model Prediction

Taking the gravitational acceleration constant as

```
In[*]:= gravity = 9.8
```

```
Out[*]= 9.8
```

our model gives

$$w(x) = - \text{density gravity } x^2 / (2 \text{ elasticity}) + [\text{density gravity length} / \text{elasticity} + 1] x$$

One possible choice for the elasticity constant is to choose it so that the end height is attained exactly. An alternative would be obtain the best overall fit of the measured heights, e.g., using least squares. Let's start with the first possibility. That is,

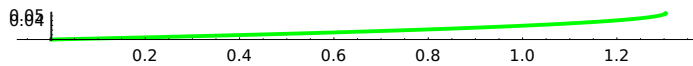
```
In[*]:= elasticityone = density gravity length^2 / (2 (-correctedmks[[39]] - length))
```

```
Out[*]= 0.0461157
```

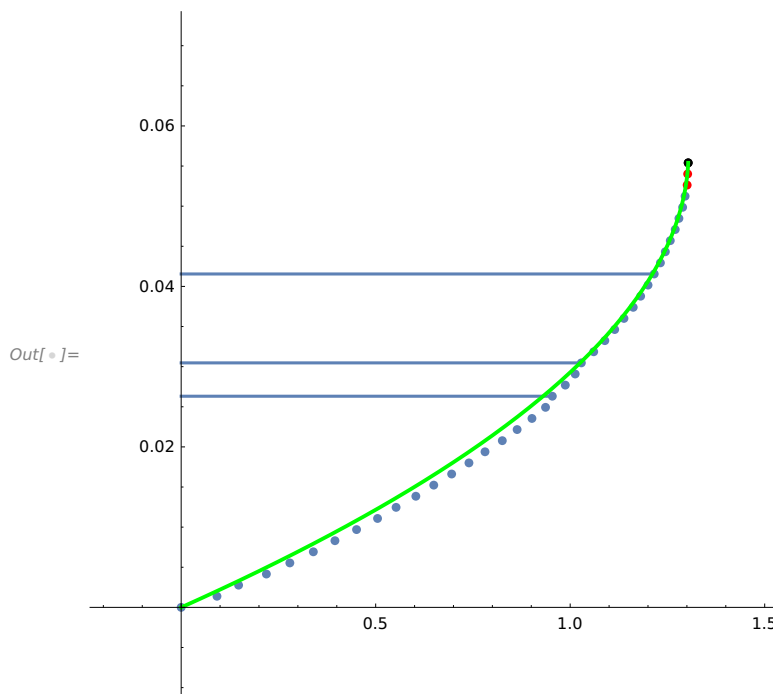
```
In[*]:= wone[x_] =
```

$$- \text{density gravity } x^2 / (2 \text{ elasticityone}) + (\text{density gravity length} / \text{elasticityone} + 1) x;$$

```
In[*]:= curveone = ParametricPlot[{wone[x], x}, {x, 0, length}, PlotStyle -> {Thick, Green}]
```

```
Out[*]= 
```

```
In[*]:= Show[correctedplot, curveone]
```



Initial Observations

1. Noting that the model is normalized to match the data exactly at the endpoints, the following seem pretty clear:

(a) The model matches the data extremely well when there is relatively small extension (and consequently relatively small tightening of the coils---this was not specifically observed above, but it is indeed the case: The tightening of the coils increases in magnitude with increasing extension).

(b) The model does not match the data for relatively large extensions very well. The mismatch is clearly coherent and significant.

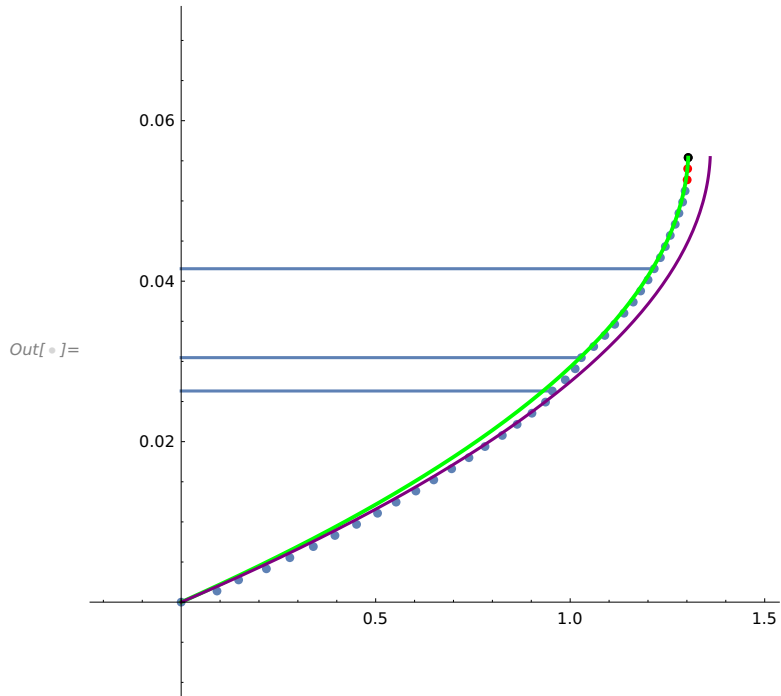
2. It is natural to assume the elasticity “constant” is not constant but depends on the magnitude of the extension, i.e.,

$$\text{elasticity} = \text{elasticity}(w'-1) \quad \text{or equivalently} \quad \text{elasticity} = \text{elasticity}(w').$$

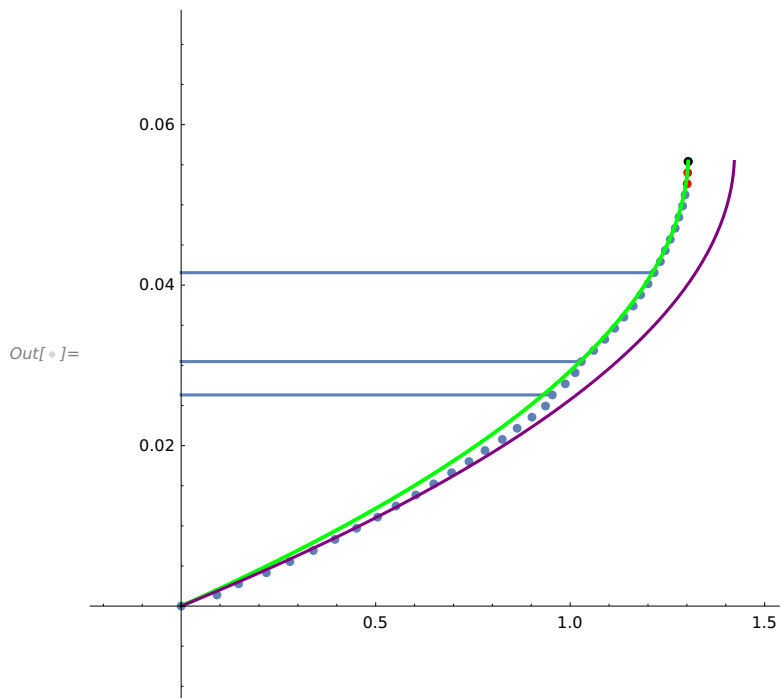
3. The fact that the model predicts an extension of smaller magnitude than that which is observed (when the extension is of large magnitude) suggests the elasticity becomes smaller with increasing w' . We can give a rough test of this hypothesis by simply decreasing the elasticity constant and seeing if our current model gives a better fit for the portion with large extension. This, however, does not give us an improved model, which is what we would really like.

```
In[*]:= wtest[x_, eps_] = -density gravity x^2 / (2 (elasticityone - eps)) +
  (density gravity length / (elasticityone - eps) + 1) x;
```

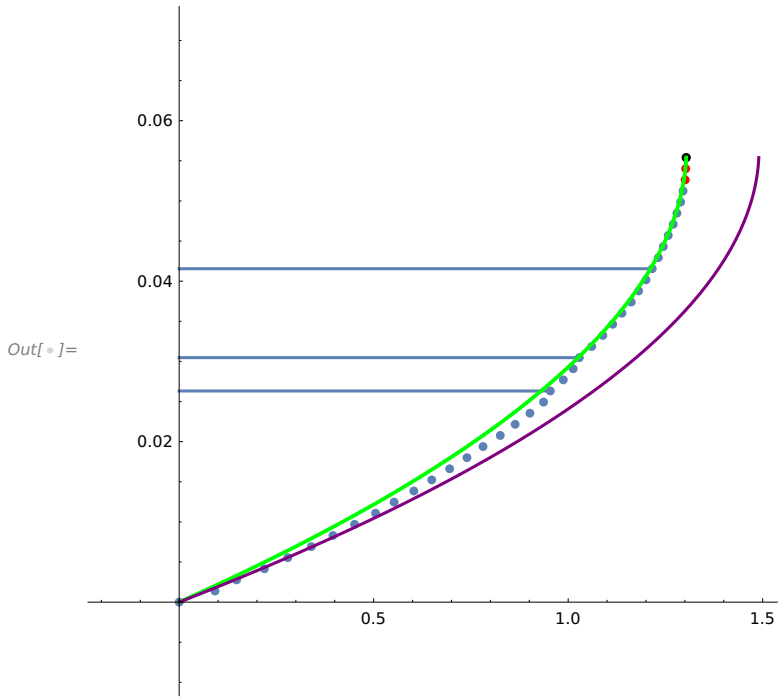
```
In[ ]:= Show[correctedplot, curveone,  
  ParametricPlot[{wtest[x, 0.002], x}, {x, 0, length}, PlotStyle -> Purple]]
```



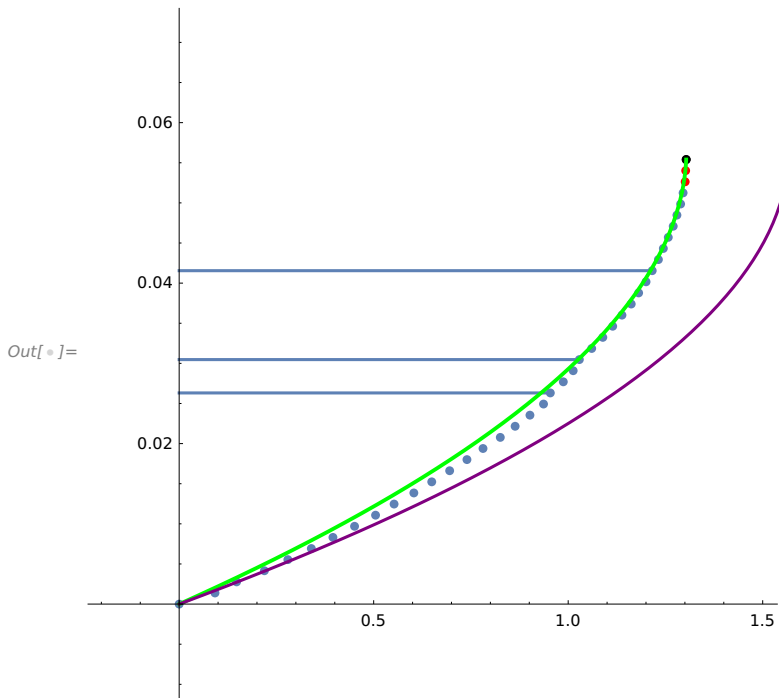
```
In[ ]:= Show[correctedplot, curveone,  
  ParametricPlot[{wtest[x, 0.004], x}, {x, 0, length}, PlotStyle -> Purple]]
```



```
In[*]:= Show[correctedplot, curveone,  
  ParametricPlot[{wtest[x, 0.006], x}, {x, 0, length}, PlotStyle -> Purple]]
```



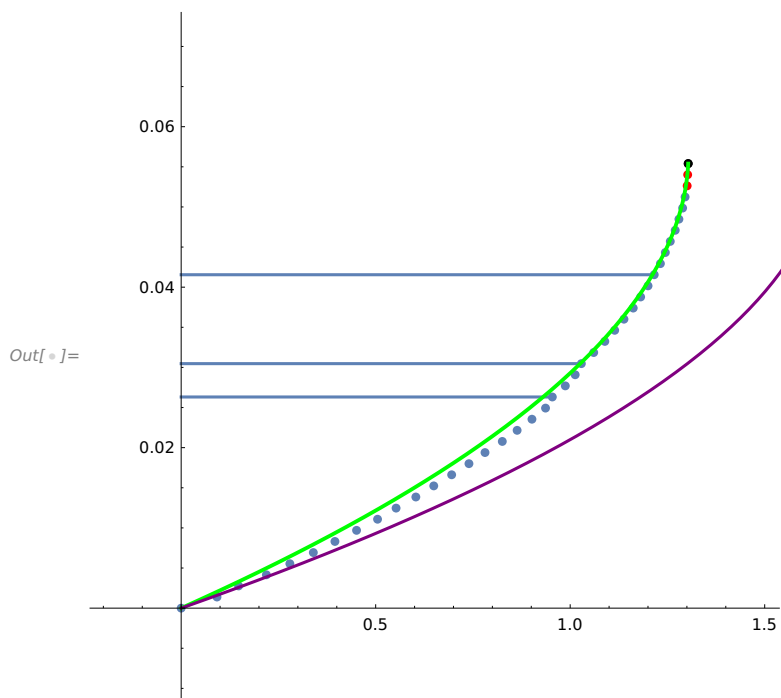
```
In[*]:= Show[correctedplot, curveone,  
  ParametricPlot[{wtest[x, 0.008], x}, {x, 0, length}, PlotStyle -> Purple]]
```




```

In[ ]:= Show[correctedplot, curveone,
  ParametricPlot[{wtest[x, 0.01], x}, {x, 0, length}, PlotStyle -> Purple]]

```



The plots above suggest a variation in the elasticity between elasticityone-0.01 and elasticityone. A crude test of the monotone variation also suggested by these plots may be given as follows:

```

In[ ]:= elasticitytesttwo[x_] = (length - x) (elasticityone - 0.01) / length + x elasticityone / length;

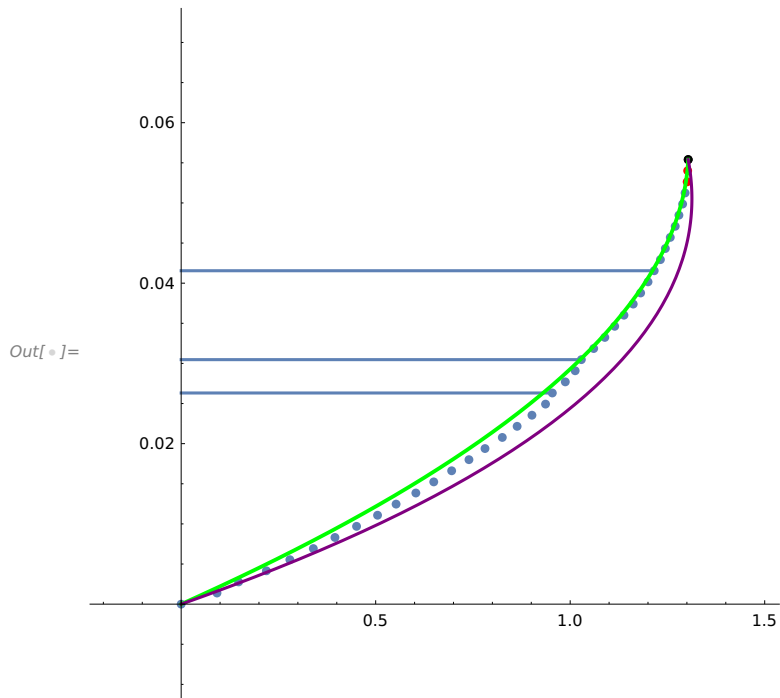
```

```

In[ ]:= wtesttwo[x_] = -density gravity x^2 / (2 elasticitytest[x]) +
  (density gravity length / elasticitytest[x] + 1) x;

```

```
In[ ]:= Show[correctedplot, curveone,
  ParametricPlot[{wtesttwo[x], x}, {x, 0, length}, PlotStyle -> Purple]]
```

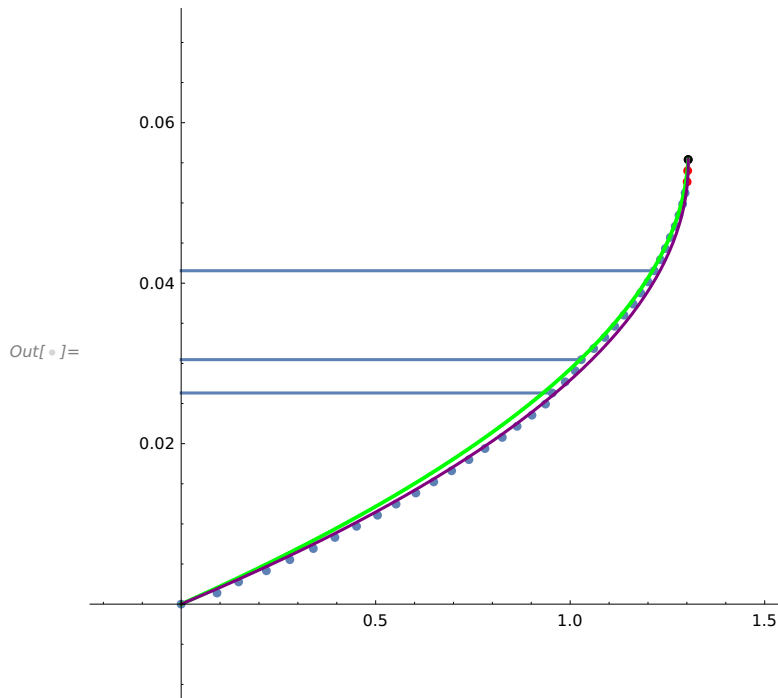


This is not a very good fit, but consider what happens if we are a little less ambitious with our “correction” as follows:

```
In[ ]:= elasticitytestthree[x_] =
  (length - x) (elasticityone - 0.003) / length + x (elasticityone) / length;
```

```
In[ ]:= wtestthree[x_] = -density gravity x^2 / (2 elasticitytestthree[x]) +
  (density gravity length / elasticitytestthree[x] + 1) x;
```

```
In[ ]:= Show[correctedplot, curveone,
  ParametricPlot[{wtestthree[x], x}, {x, 0, length}, PlotStyle -> Purple]]
```



This also does not constitute an improved model, and clearly the fit is not ideal, though it may be an improvement over the original model fit---it is clearly so in the region of large extension. It does at some level confirm the suggestion that a nontrivial monotone decreasing dependence of the elasticity on the magnitude extension should give a better model.

Improved Model

What can you do? Hint(s): Use the calculus of variations approach minimizing the potential energy. Assume the elasticity has a simple (affine) dependence on w' . Derive a (totally) different (and better) ODE. Solve that ODE numerically.