

# Slinky Take Two

```
In[1]:= mass = 212.0 / 1000;  
pluslength = 55.4 / 1000;  
density = mass / pluslength;  
gravity = 9.8;  
coils = 84;  
width = pluslength / coils;  
length = pluslength - width;
```

```
In[8]:= mcdata = {0, 2 + 3 / 4, 5 + 1 / 4, 7 + 5 / 8, 10, 12 + 3 / 8, 14 + 5 / 8, 17, 19 + 1 / 8, 21 + 1 / 8,  
23 + 3 / 16, 25 + 1 / 8, 27, 28 + 7 / 8, 30 + 5 / 8, 32 + 1 / 4, 34, 35.5, 36 + 15 / 16, 38 + 3 / 8,  
39 + 3 / 4, 41, 42 + 1 / 8, 43 + 1 / 4, 44 + 3 / 8, 45 + 3 / 8, 46 + 1 / 4, 47 + 1 / 8, 48, 48 + 3 / 4,  
49.5, 50, 50.5, 51, 51 + 3 / 8, 51 + 3 / 4, 52, 52 + 1 / 4, 52 + 3 / 8, 52.5, 52 + 9 / 16, 52 + 5 / 8}
```

```
Out[8]= {0,  $\frac{11}{4}$ ,  $\frac{21}{4}$ ,  $\frac{61}{8}$ , 10,  $\frac{99}{8}$ ,  $\frac{117}{8}$ , 17,  $\frac{153}{8}$ ,  $\frac{169}{8}$ ,  $\frac{371}{16}$ ,  $\frac{201}{8}$ , 27,  $\frac{231}{8}$ ,  
 $\frac{245}{8}$ ,  $\frac{129}{4}$ , 34, 35.5,  $\frac{591}{16}$ ,  $\frac{307}{8}$ ,  $\frac{159}{4}$ , 41,  $\frac{337}{8}$ ,  $\frac{173}{4}$ ,  $\frac{355}{8}$ ,  $\frac{363}{8}$ ,  $\frac{185}{4}$ ,  $\frac{377}{8}$ ,  
48,  $\frac{195}{4}$ , 49.5, 50, 50.5, 51,  $\frac{411}{8}$ ,  $\frac{207}{4}$ , 52,  $\frac{209}{4}$ ,  $\frac{419}{8}$ , 52.5,  $\frac{841}{16}$ ,  $\frac{421}{8}}$ }
```

```
In[9]:= Length[mcdata]
```

```
Out[9]= 42
```

```
In[10]:= mkdata = -2.54 mcdata / 100
```

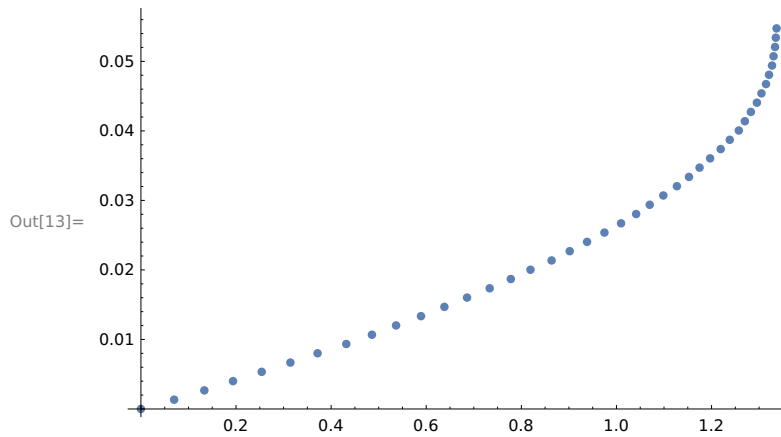
```
Out[10]= {0., -0.06985, -0.13335, -0.193675, -0.254, -0.314325, -0.371475, -0.4318, -0.485775,  
-0.536575, -0.588963, -0.638175, -0.6858, -0.733425, -0.777875, -0.81915, -0.8636,  
-0.9017, -0.938213, -0.974725, -1.00965, -1.0414, -1.06998, -1.09855, -1.12713,  
-1.15253, -1.17475, -1.19698, -1.2192, -1.23825, -1.2573, -1.27, -1.2827, -1.2954,  
-1.30493, -1.31445, -1.3208, -1.32715, -1.33033, -1.3335, -1.33509, -1.33668}
```

```
In[11]:= step[a_, b_, m_, j_] = a + j (b - a) / m;
```

```
In[12]:= plotdata = Table[{-mkdata[[j + 1]], step[0, length, 41, j]}, {j, 0, 41}];
```

In[13]:=

ListPlot[plotdata]



Out[13]=

## constant elasticity model

Lagrangian :  $\text{eps} (w'-1)^2/2 - \rho g w$   
 with eps = elasticity assumed constant

Euler-Lagrange ODE:  $\text{eps} w'' = -\rho g$

general solution with  $w'(L) = 1$ :

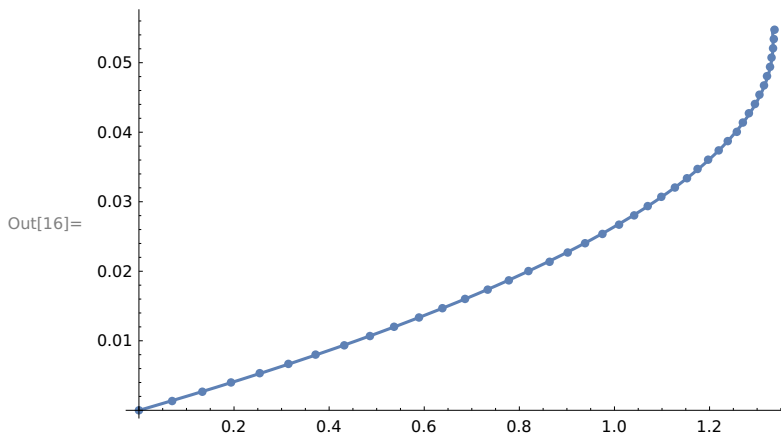
$$w(x) = -\rho g x^2/(2 \text{eps}) + (\rho g L/\text{eps} + 1) x + b.$$

```
In[14]:= fitdata = Table[{step[0, length, 41, j], -mkdata[[j + 1]]}, {j, 0, 41}];
```

```
In[15]:= conone[x_] = Fit[fitdata, {1, x, x^2}, x]
```

```
Out[15]= 0.000689467 + 50.4416 x - 477.238 x^2
```

```
In[16]:= Show[ListPlot[plotdata], ParametricPlot[{conone[x], x}, {x, 0, length}]]
```



This suggests we can fit the data quite well with a quadratic function. The coefficients of this function, however, do not fit the form of our model:

```
In[17]:= elasticconstantquadratic =
  NSolve[-density gravity / (2 findeps) == Coefficient[conone[x], x^2][[1]][[1, 2]]
```

Out[17]= 0.0392905

```
In[18]:= elasticconstantlinear =
  NSolve[density gravity / findeps + 1 == Coefficient[conone[x], x][[1]][[1, 2]]
```

Out[18]= 0.758508

To see this another way, consider choosing the coefficient  $b = 0$  in the model so that  $w(0) = 0$ . Then there is precisely one value of the elasticity constant  $\text{eps}$  for which  $w(\text{length}) = \text{the measured value} = 52 + 5/8 \text{ inches} = 1.337 \text{ m}$ . That is,

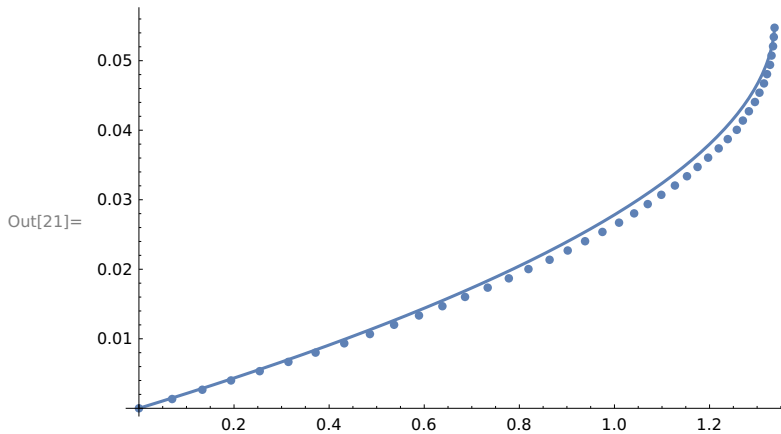
```
In[19]:= elasticconstantends =
  NSolve[density gravity length^2 / (2 findeps) + length == -mkdata[[42]][[1]][[1, 2]]
```

Out[19]= 0.0438302

Notice that this value is between the other two we got from the best quadratic fit.

```
In[20]:= quadraticends[x_] = -density gravity x^2 / (2 elasticconstantends) +
  (density gravity length / elasticconstantends + 1) x;
```

```
In[21]:= Show[ListPlot[plotdata], ParametricPlot[{quadraticcends[x], x}, {x, 0, length}]]
```



And this fit is not so great.

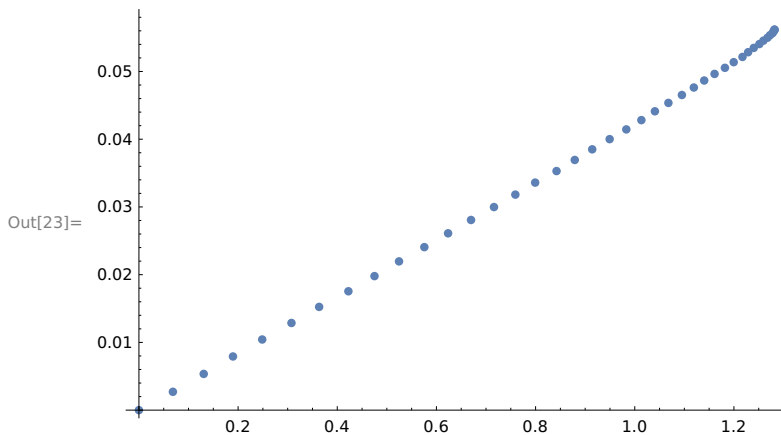
Another possibility is to assume again that we have  $w(0) = 0$ , so that  $b = 0$ , and then write

$$w(x) = (1/\text{eps}) (-\rho g x^2/2 + \rho g \text{length} x) + x.$$

We can then use our data set to consider  $w(x)$  as a linear function of  $(-\rho g x^2/2 + \rho g \text{length} x)$  and solve for the best eps:

```
In[22]:= secondbestfitdata = Table[{-mkdata[[j + 1]] - step[0, length, 41, j], density gravity
  (-step[0, length, 41, j]^2 / 2 + length step[0, length, 41, j])}, {j, 0, 41}];
```

```
In[23]:= ListPlot[secondbestfitdata]
```



```
In[24]:= second[x_] = Fit[secondbestfitdata, {x}, x]
```

Out[24]= 0.0428783 x

```
In[25]:= secondeps = Coefficient[second[x], x]
```

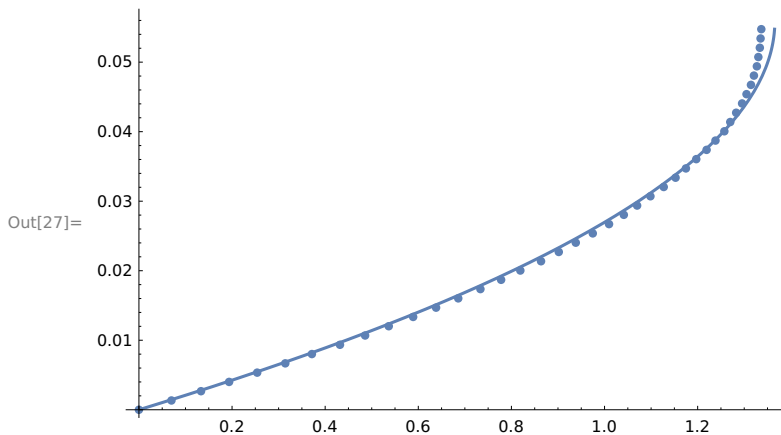
```
Out[25]= 0.0428783
```

```
In[26]:= secondquadratic[x_] =
```

```
  -density gravity x^2 / (2 secondeps) + (density gravity length / secondeps + 1) x;
```

```
In[27]:= Show[ListPlot[plotdata],
```

```
  ParametricPlot[{secondquadratic[x], x}, {x, 0, length}], PlotRange -> All]
```



This doesn't fit so great either especially near  $x = \text{length}$  where the extension is small, but the elasticity constant is rather close to the last one; around 0.043.

Finally, we might use  $\text{elasticity} = 0.043$  and attempt to fit a quadratic with nonzero constant term  $b$ :

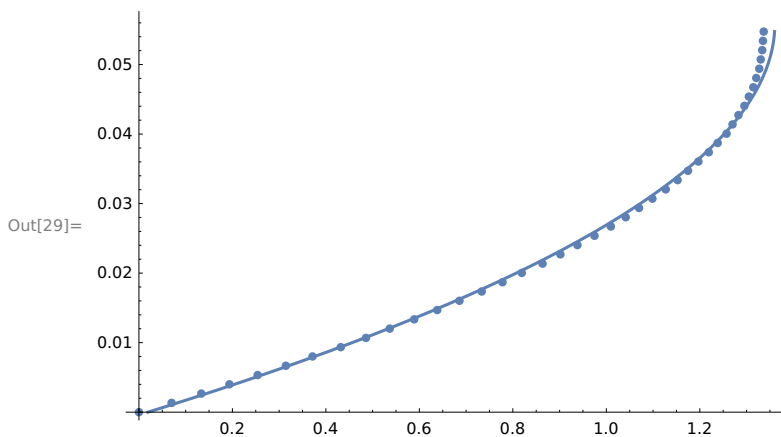
```
In[28]:= finalquadratic[x_] =
```

```
  Fit[fitdata, {-density gravity x^2 / (2 (0.043)) + (density gravity length / 0.043 + 1) x, 1}, x]
```

```
Out[28]= 0.0191804 + 0.984804 (48.7411 x - 436.068 x^2)
```

```
In[29]:= Show[ListPlot[plotdata],
```

```
  ParametricPlot[{finalquadratic[x], x}, {x, 0, length}], PlotRange -> All]
```



That is perhaps about the best we can do with the constant elasticity model. Some modelers have, at

this point, looked at the nice quadratic fit and simply adopted that as a “model” without any really good explanation, except that there is a good fit of the data. We will proceed somewhat differently.

## Preliminary discussion of improvement

The basic assertion is that the elasticity should depend on the extension rate  $w'$ . Here are some calculations intended to give some idea about how to find that dependence:

Given two data values  $(x_j, w_j)$  and  $(x_k, w_k)$  give

$$\begin{aligned} w_j &= -\rho g x_j^2 / (2 \text{eps}) + (\rho g L / \text{eps} + 1) x_j + b \\ w_k &= -\rho g x_k^2 / (2 \text{eps}) + (\rho g L / \text{eps} + 1) x_k + b \end{aligned}$$

and

$$w_k - w_j = -\rho g [x_k^2 - x_j^2] / (2 \text{eps}) + (\rho g L / \text{eps} + 1)(x_k - x_j)$$

or

$$w_k - w_j - (x_k - x_j) = (\rho g / \text{eps}) [ - (x_k^2 - x_j^2) / 2 + L(x_k - x_j) ].$$

which gives

$$\text{eps} = \rho g (x_k - x_j) [ L - (x_k + x_j) / 2 ] / [ w_k - w_j - (x_k - x_j) ]$$

Thus, we define

```
In[30]:= elas[xone_, xtwo_, wone_, wtwo_] =
  density gravity (xtwo - xone) ( length - (xtwo + xone) / 2 ) / (wtwo - wone - (xtwo - xone));
```

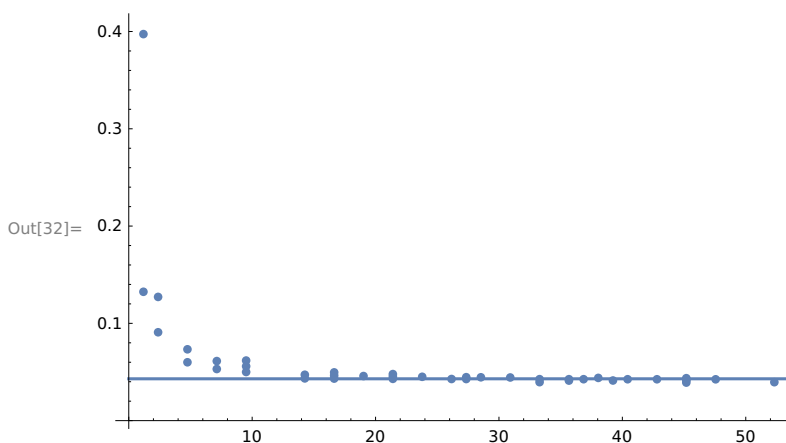
```
In[31]:= elasticitydata =
  Table[{{(-mkdata[[j + 2]] + mkdata[[j + 1]]) / (step[0, length, 41, j + 1] - step[0, length, 41, j]),
    elas[step[0, length, 41, j], step[0, length, 41, j + 1],
    -mkdata[[j + 1]], -mkdata[[j + 2]]}}, {j, 0, 40}]
```

```
Out[31]:= {{52.3169, 0.0395159}, {47.5608, 0.042477}, {45.1827, 0.04363}, {45.1827, 0.0424967},
  {45.1827, 0.0413635}, {42.8047, 0.0425187}, {45.1827, 0.039097}, {40.4267, 0.0425433},
  {38.0486, 0.0439226}, {39.2376, 0.0412474}, {36.8596, 0.0425864},
  {35.6706, 0.0426028}, {35.6706, 0.0411586}, {33.2925, 0.042639}, {30.9145, 0.0443548},
  {33.2925, 0.039538}, {28.5365, 0.0445487}, {27.3475, 0.0446587}, {27.3475, 0.0427583},
  {26.1584, 0.042789}, {23.7804, 0.0450578}, {21.4024, 0.0478554}, {21.4024, 0.0454013},
  {21.4024, 0.0429472}, {19.0243, 0.0458355}, {16.6463, 0.0496018},
  {16.6463, 0.0464017}, {16.6463, 0.0432016}, {14.2682, 0.0471709},
  {14.2682, 0.0433972}, {9.51216, 0.0617627}, {9.51216, 0.0558806}, {9.51216, 0.0499984},
  {7.13412, 0.061219}, {7.13412, 0.0530564}, {4.75608, 0.073317}, {4.75608, 0.0599867},
  {2.37804, 0.12717}, {2.37804, 0.0908354}, {1.18902, 0.397339}, {1.18902, 0.132446}}
```

The first coordinate here is approximating the extension rate  $w'$  and the second is using the two points to determine a value of the elasticity constant. The problem with using only two points (at a time) is that there is some instability in the resulting data, especially for small extensions when  $w'$  is close to 1 at the end. This is somewhat expected because, first of all, the coils are all bunched up there and it is difficult to get accurate height measurements. In fact, it is probably the case that the accuracy of the measurement of the height of any given coil is significantly outside the width of the equilibrium coils, so one can expect this data to be rather bad (from point to point for the last few points). In addition, the coil tends to “bounce” a bit, so reading the height is difficult for that reason too.

At any rate, this is what you see in the data. For relatively large extension, the elasticity is around our best constant value 0.043. It gets fuzzy for small extension, i.e.,  $w'$  close to 1, but is uniformly higher.

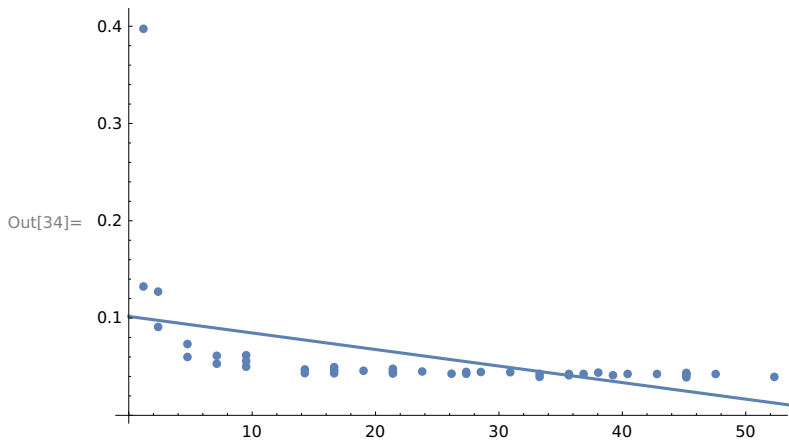
```
In[32]:= Show[ListPlot[elasticitydata, AxesOrigin -> {0, 0}, PlotRange -> All], Plot[0.043, {x, 0, 55}]]
```



```
In[33]:= elaone[x_] = Fit[elasticitydata, {1, x}, x]
```

```
Out[33]= 0.101686 - 0.00170085 x
```

```
In[34]:= Show[ListPlot[elasticitydata, AxesOrigin -> {0, 0}, PlotRange -> All],
  Plot[elaone[x], {x, 0, 55}]]
```



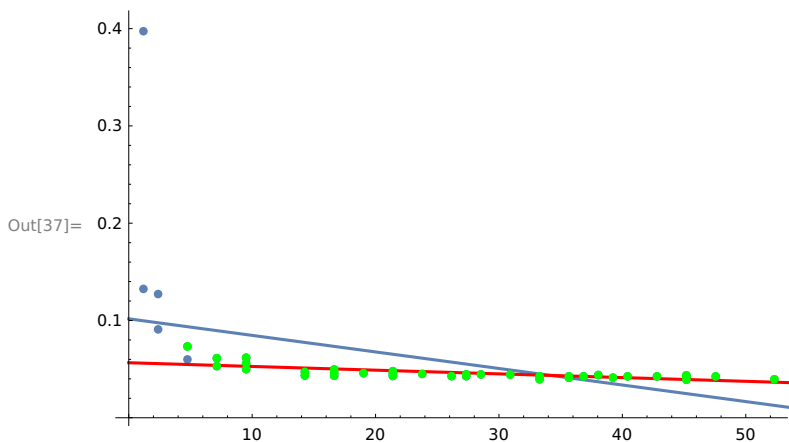
```
In[35]:= eladata =
```

```
Table[{-mkdata[[j + 2]] + mkdata[[j + 1]] / (step[0, length, 41, j + 1] - step[0, length, 41, j]),
  elas[step[0, length, 41, j], step[0, length, 41, j + 1],
  -mkdata[[j + 1]], -mkdata[[j + 2]]], {j, 0, 35}];
```

```
In[36]:= elatwo[x_] = Fit[eladata, {1, x - 1}, x]
```

```
Out[36]= 0.0561842 - 0.000381929 (-1 + x)
```

```
In[37]:= Show[ListPlot[elasticitydata, AxesOrigin -> {0, 0}, PlotRange -> All],
  Plot[elaone[x], {x, 0, 55}], Plot[elatwo[x], {x, 0, 55}, PlotStyle -> Red],
  ListPlot[eladata, PlotStyle -> Green]]
```



Let us remember the coefficients of  $\text{elatwo}[w'] = \alpha(w'-1) + \beta$



```
In[41]:= alphas = -0.00038192900950803854` ;
betas = 0.05618417202447363` ;
```

## Variable elasticity model

As suggested above, we will consider a variable elasticity of the form  $\epsilon = a(w'-1) + b$  where presumably the coefficient  $a$  is small and negative and the coefficient  $b$  is somewhere around 0.043.

Here is a derivation:

Lagrangian :  $\epsilon (w'-1)^2/2 - \rho g w$

Euler-Lagrange ODE:  $-(d/dx)[ \epsilon' (w'-1)^2/2 + \epsilon (w'-1) ] = \rho g$

Note that here  $\epsilon' = (d\epsilon / dp) = a$  is constant where  $p$  denotes the  $p = w'$  dependence.

Thus, we get

$$[ a(w'-1) + a (w'-1) + \epsilon ] w'' = -\rho g.$$

That is,

$$[ 3 a (w'-1) + b ] w'' = -\rho g.$$

Integrating once from  $x$  to  $L$  (and remembering the condition  $w'(L) = 1$ ) we get

$$-[ 3 a w'^2/2 + (b - 3a) w' ] + b-3a/2 = -\rho g (L-x).$$

This can be written as

$$3 a w'^2 + 2(b-3a) w' - 2 \rho g(L-x) + 3a-2b = 0.$$

Solving the quadratic,

$$\begin{aligned} w' &= (1/(6a)) [ 2(3a-b) + 2 \sqrt{ (b-3a)^2 - 3a(-2 \rho g(L-x) + 3a-2b) } ] \\ &= (1/(3a)) [ 3a-b + \sqrt{ b^2 + 6a \rho g(L-x) } ] \end{aligned}$$

We take the “+“ sign in the quadratic formula because  $3a-b$  is assumed to be negative,  $b$  being rather

larger than a and positive, and we need an opposite sign to get the quadratic model (constant elasticity) in the limit as a tends to 0, i.e.,  $\epsilon = a w' + b$  tends to b.

Integrating this time from 0 to x and assuming  $w(0) = 0$  we get

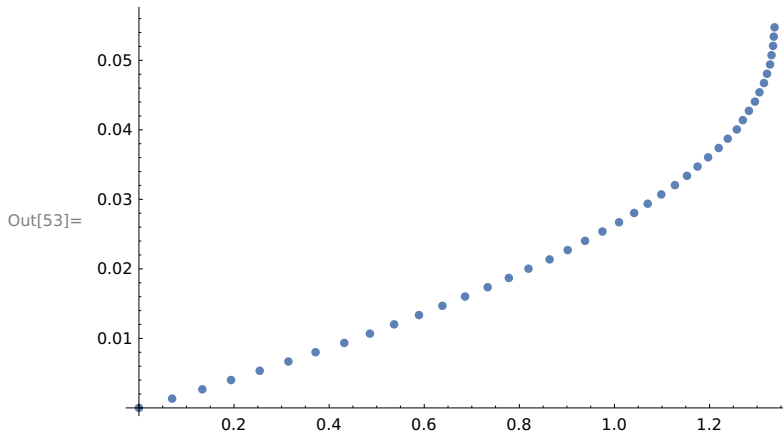
$$w(x) = \frac{1-b/(3a)}{3a} x - \frac{1/(3a)}{3a} \left( \frac{2}{3} [b^2 + 6a \rho g(L-x)]^{3/2} \frac{1}{(6a \rho g)} \right) + \frac{1/(3a)}{3a} \left( \frac{2}{3} [b^2 + 6a \rho g L]^{3/2} \frac{1}{(6a \rho g)} \right).$$

Substituting our choice  $\epsilon(w')$  for  $\epsilon$  here we get

$$wtwo[x_] = (1 - betatwo / (3 alphas)) x - (1 / (27 alphas^2 density gravity)) ((betatwo^2 + 6 alphas density gravity (length - x))^{3/2} - (betatwo^2 + 6 alphas density gravity length)^{3/2})$$

Out[52]= 50.0354 x - 6770.47 ((0. + 0.0000608839 i) + Abs[0.00315666 - 0.0859382 (0.0547405 - x)]<sup>3/2</sup>)

In[53]:= Show[ListPlot[plotdata], ParametricPlot[{wtwo[x], x}, {x, 0.02, length}], PlotRange -> All]



In[49]:= alphas

Out[49]= -0.000381929

In[48]:= Plot[betatwo^2 + 6 alphas density gravity (length - x), {x, 0, length}]

