Test Machine Linearizer

P. McGill

v1.1.0

Enter a table of the input and output values of the linearizer.

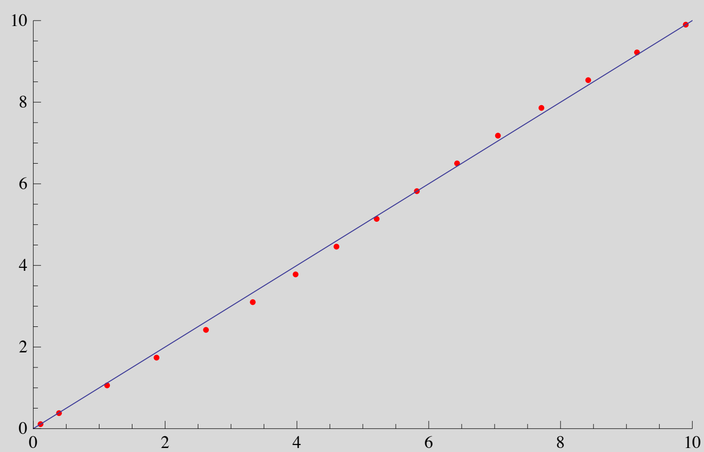
```
data = {{9.90, 9.90}, {9.16, 9.22}, {8.42, 8.54}, {7.71, 7.86}, {7.05, 7.18}, {6.43, 6.50},
    {5.82, 5.82}, {5.21, 5.14}, {4.60, 4.46}, {3.98, 3.78}, {3.33, 3.10},
    {2.62, 2.42}, {1.87, 1.74}, {1.12, 1.06}, {0.39, 0.38}, {0.11, 0.11}};
```

```
data = Reverse[data]
```

```
{{0.11, 0.11}, {0.39, 0.38}, {1.12, 1.06}, {1.87, 1.74}, {2.62, 2.42},
 {3.33, 3.1}, {3.98, 3.78}, {4.6, 4.46}, {5.21, 5.14}, {5.82, 5.82},
 {6.43, 6.5}, {7.05, 7.18}, {7.71, 7.86}, {8.42, 8.54}, {9.16, 9.22}, {9.9, 9.9}}
```

```
dataplot = ListPlot[data, {PlotRange → {{0, 10}, {0, 10}}, PlotStyle → Red}];
```

```
Show[dataplot, Plot[x, {x, 0, 10}]]
```



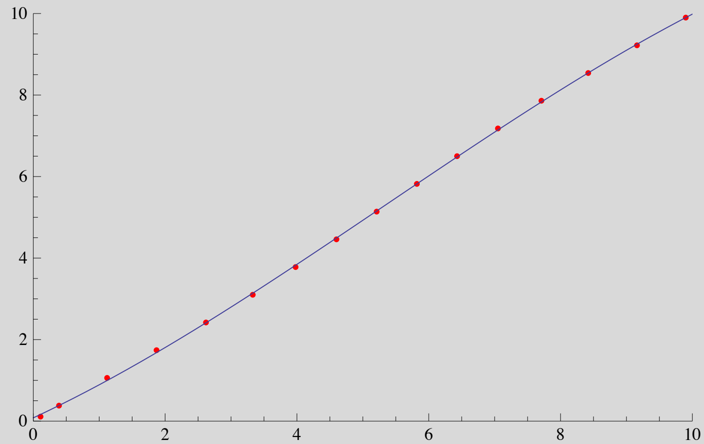
Do a least-squares fit to a cubic.

```
cubiccurvefit = Fit[data, {1, x, x^2, x^3}, x]
```

$0.0805993 + 0.750762\,x + 0.0634223\,x^2 - 0.0039459\,x^3$

```
cubiccurvefitplot = Plot[cubiccurvefit, {x, 0, 10}];
```

```
Show[dataplot, cubiccurvefitplot]
```



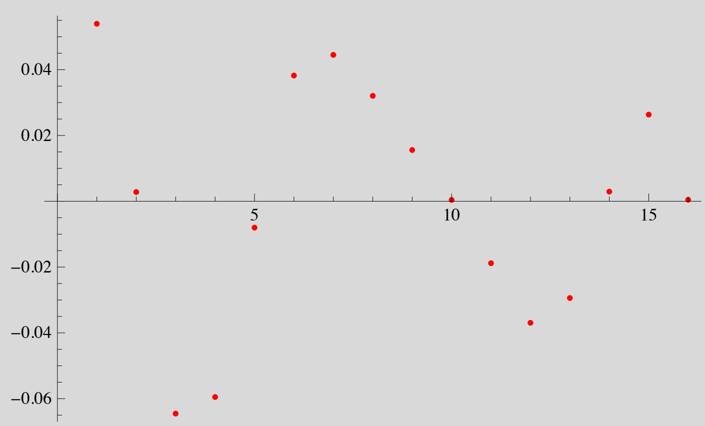
Find the error as the curve fit output minus the ideal output (i.e. actual minus desired). Note that this is a table of errors ranging from the first data point to the sixteenth data point.

```
cubicerror[i_] := cubiccurvefit - data[[i, 2]] /. x → data[[i, 1]]
```

```
cubicerrors = Table[cubicerror[i], {i, 16}]
```

```
{0.0539453, 0.00280896, -0.064534, -0.0594974, -0.00801432,
 0.0382135, 0.0444976, 0.0320414, 0.0155771, 0.00041362, -0.0188231,
 -0.0369383, -0.029409, 0.00292759, 0.0263383, 0.000452748}
```

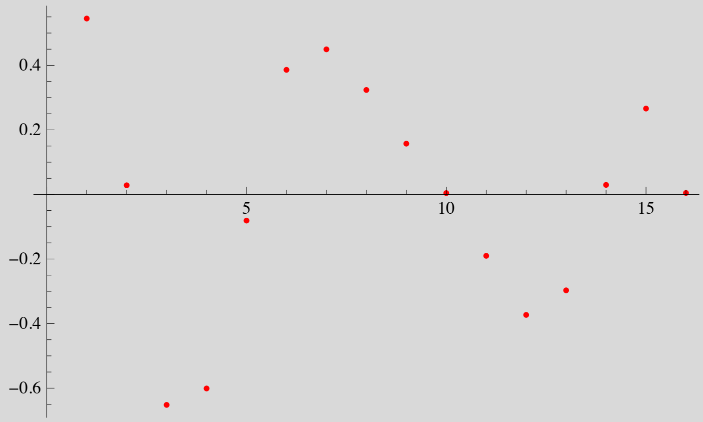
```
ListPlot[cubicerrors, PlotStyle → Red]
```



Let’s look at the error as a percentage of full scale:

```
cubicpercenterrors = Table[100 cubicerror[i] / 9.90, {i, 16}];
```

```
ListPlot[cubicpercenterrors, {PlotStyle → Red, PlotRange → All}]
```



What position error does this correspond to?

```
Max[Abs[cubicpercenterrors]] 72 inches / 100
```

```
0.469338 inches
```

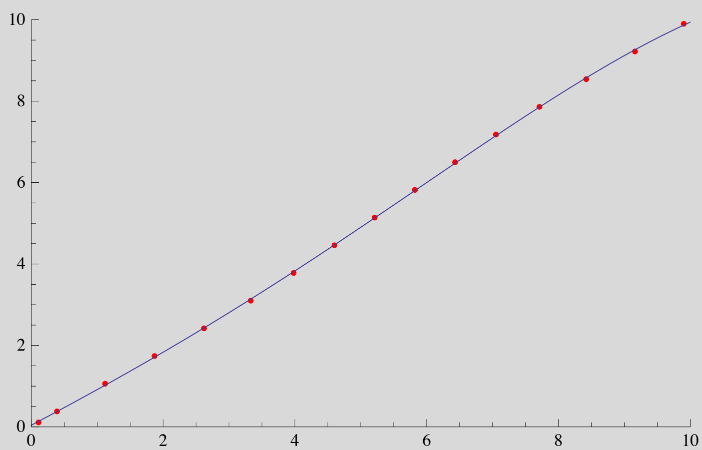
Not so great. Can we reduce the error with a quartic fit?

```
quarticcurvefit = Fit[data, {1, x, x^2, x^3, x^4}, x]
```

$0.0405363 + 0.859428\, x + 0.0116751\, x^2 + 0.00423185\, x^3 - 0.0004094\, x^4$

```
quarticcurvefitplot = Plot[quarticcurvefit, {x, 0, 10}];
```

```
Show[dataplot, quarticcurvefitplot]
```

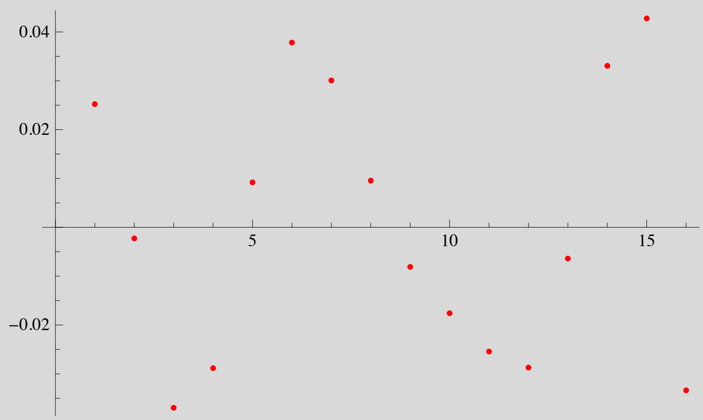


Find the error as the curve fit output minus the ideal output:

```
quarticerror[i_] := quarticcurvefit - data[[i, 2]] /. x → data[[i, 1]]
```

```
quarticerrors = Table[quarticerror[i], {i, 16}];
```

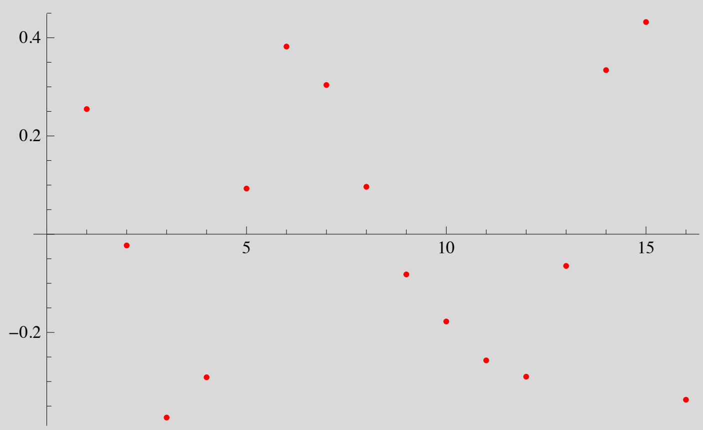
```
ListPlot[quarticerrors, PlotStyle → Red]
```



Let’s look at the error as a percentage of full scale:

```
quarticpercenterrors = Table[100 quarticerror[i] / 9.90, {i, 16}];
```

```
ListPlot[quarticpercenterrors, PlotStyle → Red]
```



Looks better, but how good is it?

```
Max[Abs[quarticpercenterrors]] 72 inches / 100
```

```
0.310989 inches
```

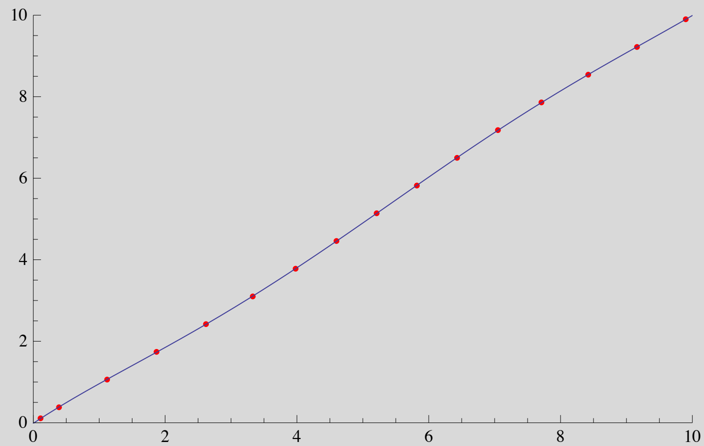
Lets push it one more order.

```
quinticcurvefit = Fit[data, {1, x, x^2, x^3, x^4, x^5}, x]
```

$-0.0137674 + 1.08168\,x - 0.153537\,x^2 + 0.0490865\,x^3 - 0.00547533\,x^4 + 0.000202089\,x^5$

```
quinticcurvefitplot = Plot[quinticcurvefit, {x, 0, 10}];
```

```
Show[dataplot, quinticcurvefitplot]
```

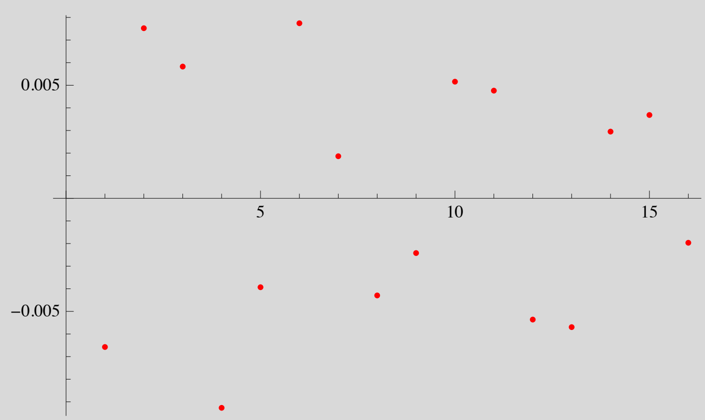


Find the error as the curve fit output minus the ideal output:

```
quinticerror[i_] := quinticcurvefit - data[[i, 2]] /. x → data[[i, 1]]
```

```
quinticerrors = Table[quinticerror[i], {i, 16}];
```

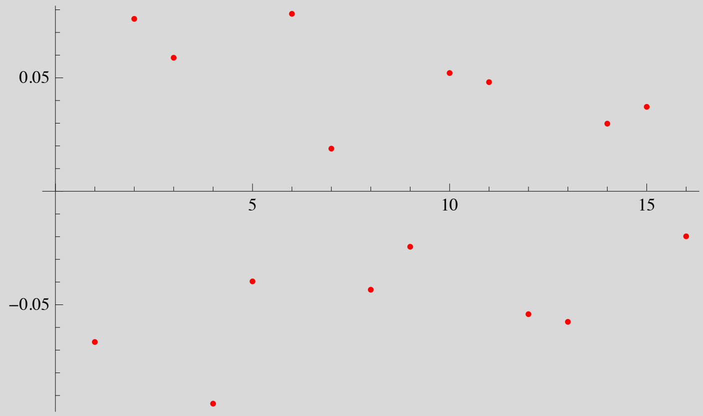
```
ListPlot[quinticerrors, PlotStyle → Red]
```



Let's look at the error as a percentage of full scale:

```
quinticpercenterrors = Table[100 quinticerror[i] / 9.90, {i, 16}];
```

```
ListPlot[quinticpercenterrors, PlotStyle → Red]
```




```
Max[Abs[quinticpercenterrors]] 72 inches / 100
```

```
0.0673854 inches
```

Wow, this is a big improvement over the quartic fit, and probably smaller than other errors in the system.