Education and Culture DG

Lifelong Learning Programme

This project has been funded with support from the European Commission.
This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Describing real life digital pictures with mathematical functions

BHAK Wien 10, Vienna, Austria
Fundacao Escola Profissional de Setubal, Setubal, Portugal
Raelingen videregaaende skole, Raelingen, Norway

PC based Math Project for High School Students

1 Description of the Project

Open ê Close Print

Introduction:

Open ê Close

1. The students from a couple of countries come to Portugal to work with *Mathematica*. The point of it is that we're gonna find the relations between maths and real life. With the support of some basics from the *Mathematica and a M@th Desktop plug-in we're gonna fit the data from pictures and determine the best practice. We're gonna work with this functions, with x-intersection, minima and maxima. Then we transfer everything into real life. The result of it will be a graphicaly output (maybe with an animation) and a palette.*

2 Brainstorming and Theory

Open ê Close Print

Brainstorming

Open ê Close

1. At first, we have to search for useful pictures.
We also need to find out the tagent before we can find the angle.
From the equations we´ll plot the graphs.
Plot-Commands will draw them for us.
Data-Fitting is going to get the data from the picture.

What Mathematics do we need?

Open ê Close

2. We need the algebra to find out the results in the form of numbers instead of fractions. 2nd and 3rd degree curves will help us to see the jumping base at our pc´s. With the help of Data-fitting, we´ll copy the data from a picture to the program. We need the linear functions to find out the best tangent. After all, we need nearly every kind of graphic mathematics to do that :)

3

Developing Models

Open ê Close Print

Section:

Open ê Close

The picture we´ve chosen:

Input @ We get the points from the picture

Input @

```
data = 880.043299, 0.538144<, 80.0680412, 0.519588<, 80.0783505, 0.507216<,
    80.109278, 0.482474<, 80.131959, 0.465979<, 80.146392, 0.457732<,
    80.16701, 0.449485<, 80.191753, 0.426804<, 80.216495, 0.408247<,
    80.245361, 0.391753<, 80.274227, 0.36701<, 80.298969, 0.352577<,
    80.307216, 0.348454<, 80.323711, 0.338144<, 80.340206, 0.325773<,
    80.364948, 0.309278<, 80.373196, 0.305155<, 80.395876, 0.292784<,
    80.404124, 0.290722<, 80.420619, 0.278351<, 80.428866, 0.278351<,
    80.437113, 0.276289<, 80.447423, 0.272165<, 80.468041, 0.263918<,
    80.490722, 0.257732<, 80.498969, 0.25567<, 80.515464, 0.245361<,
    80.54433, 0.237113<, 80.554639, 0.240928<, 80.581443, 0.24268<,
    80.589691, 0.24268<, 80.614433, 0.24268<, 80.635052, 0.46495<<
```

```
880.043299, 0.538144<, 80.0680412, 0.519588<, 80.0783505, 0.507216<,
  80.109278, 0.482474<, 80.131959, 0.465979<, 80.146392, 0.457732<,
  80.16701, 0.449485<, 80.191753, 0.426804<, 80.216495, 0.408247<,
  80.245361, 0.391753<, 80.274227, 0.36701<, 80.298969, 0.352577<,
  80.307216, 0.348454<, 80.323711, 0.338144<, 80.340206, 0.325773<,
  80.364948, 0.309278<, 80.373196, 0.305155<, 80.395876, 0.292784<,
  80.404124, 0.290722<, 80.420619, 0.278351<, 80.428866, 0.278351<,
  80.437113, 0.276289<, 80.447423, 0.272165<, 80.468041, 0.263918<,
  80.490722, 0.257732<, 80.498969, 0.25567<, 80.515464, 0.245361<,
  80.54433, 0.237113<, 80.554639, 0.240928<, 80.581443, 0.24268<,
  80.589691, 0.24268<, 80.614433, 0.24268<, 80.635052, 0.46495<<
```

Input @ Starting the data fitting for the base.

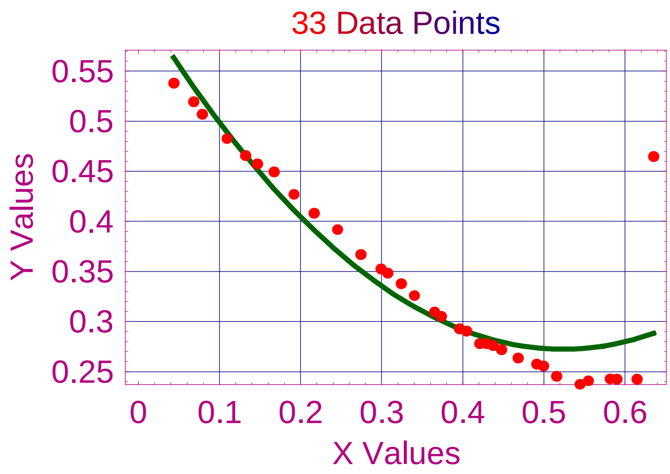
```
fit@x_D = MDNonlinearFit@data,
              a x^2 + b x + c , H* model *L
              8x<, 8a, b, c<H* parameters *L
        D êê Chop@#, 10^-5D &;
```

Input @

```
8start, stop< = 8Min@#D, Max@#D< &@First ü Transpose ü dataD;
MDPlotFitData@data, 8fit@xD< , 8x, start, stop<,
        FrameLabel Ø 8"X Values", "Y Values"<,
  Epilog Ø 8Red, PointSize@0.02D, Point êü data<,
  PlotStyle Ø 88DarkGreen, Thickness@0.01D<<D
```

‚ $(y_i - \hat{y}_i)^2$ Sum of Squared Error : 0.0429225

$0.618415 - 1.32242\, x + 1.2637\, x^2$



Ü Graphics Ü

Input @

```
f@a_D =
  0.582557532793937` - 0.9481602580689489` x + 0.574735780334971` x^2
```

$0.582558 - 0.94816\,x + 0.574736\,x^2$

The flight and landing, same technique as the base.

Input @

```
data2 = 880.641237, 0.195876<, 80.668041, 0.195876<,
    80.692784, 0.195876<, 80.715464, 0.189691<, 80.742268, 0.175258<,
    80.750515, 0.175258<, 80.764948, 0.173196<, 80.781443, 0.164948<,
    80.806186, 0.152577<, 80.816495, 0.148454<, 80.830928, 0.14433<<
```

```
880.641237, 0.195876<, 80.668041, 0.195876<,
  80.692784, 0.195876<, 80.715464, 0.189691<, 80.742268, 0.175258<,
  80.750515, 0.175258<, 80.764948, 0.173196<, 80.781443, 0.164948<,
  80.806186, 0.152577<, 80.816495, 0.148454<, 80.830928, 0.14433<<
```

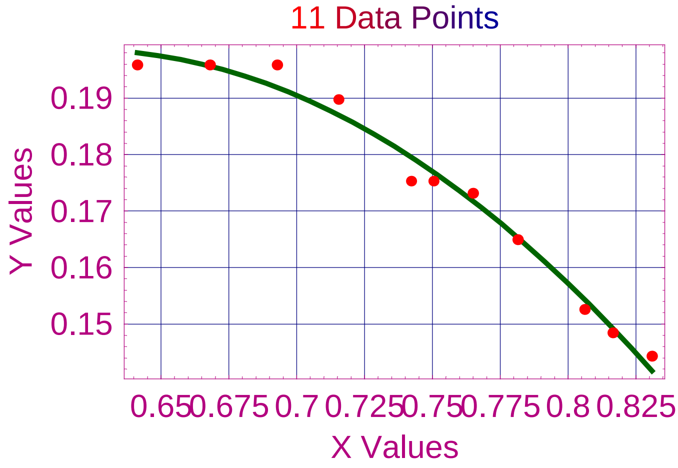
```
fit@x_D = MDNonlinearFit@data2,
                    a x^2 + b x + c , H* model *L
                    8x<, 8a, b, c <H* parameters *L
          D êê Chop@#, 10^-5D &;
```

Input @

```
8start, stop< = 8Min@#D, Max@#D< &@First ü Transpose ü data2D;
MDPlotFitData@data2, 8fit@xD< , 8x, start, stop<,
          FrameLabel Ø 8"X Values", "Y Values"<,
  Epilog Ø 8Red, PointSize@0.02D, Point êü data2<,
  PlotStyle Ø 88DarkGreen, Thickness@0.01D<<D
```

, Hy_i - ŷ_iL^2 Sum of Squared Error : 0.00006146

$-0.295454 + 1.5926\,x - 1.28347\,x^2$



Ü Graphics Ü

Input @

```
f@b_D = -0.29545360071157534` +
    1.5925957083461064` x - 1.2834722698682066` x^2
```

$-0.295454 + 1.5926\,x - 1.28347\,x^2$

Input @

```
data3 = 880.878351, 0.121649<, 80.880412, 0.119588<, 80.896907, 0.115464<,
    80.913402, 0.107216<, 80.934021, 0.107216<, 80.950515, 0.11134<<
```

```
880.878351, 0.121649<, 80.880412, 0.119588<, 80.896907, 0.115464<,
  80.913402, 0.107216<, 80.934021, 0.107216<, 80.950515, 0.11134<<
```

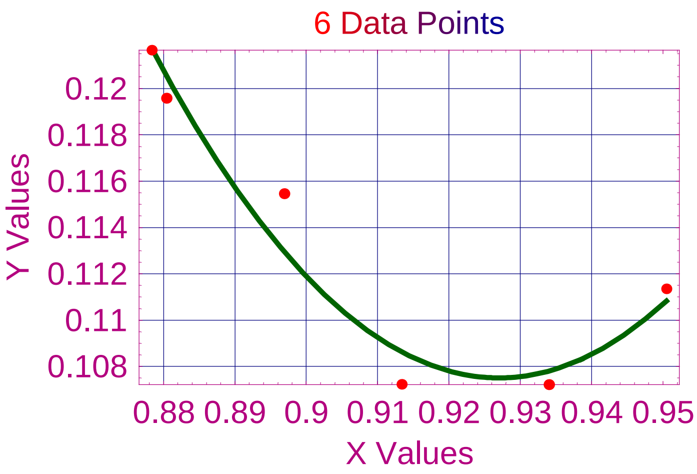
```
fit@x_D = MDNonlinearFit@data3,
                a x^2 + b x + c, H* model *L
                8x<, 8a, b, c<H* parameters *L
          D êê Chop@#, 10^-5D &;
```

Input @

```
8start, stop< = 8Min@#D, Max@#D< &@First ü Transpose ü data3D;
MDPlotFitData@data3, 8fit@xD<, 8x, start, stop<,
        FrameLabel Ø 8"X Values", "Y Values"<,
  Epilog Ø 8Red, PointSize@0.02D, Point êü data3<,
  PlotStyle Ø 88DarkGreen, Thickness@0.01D<<D
```

, $H y_i - \grave{y}_i L^2$ Sum of Squared Error : 9.82812×10^{-6}

$5.26961 - 11.1371\,x + 6.00703\,x^2$



⁃Graphics⁃

Input [

```
f[c_] =
  5.269612998873093` - 11.137133684303336` x + 6.007031603228448` x^2
```

$5.26961 - 11.1371\,x + 6.00703\,x^2$

Transfering the coordinates into real meter values.

Input [

```
UX = 0.043299
EX = 0.635052;
UY = 0.538144;
EY = 0.216495;
m = 0;(* it is the starting value of x in  real life*)
n = 45;(* it is the
    x- value of your maximum in  real life*)
s = 50;(* it is the
    y- value of your maximum in  real life*)
r = 120;(* it is the starting value of y in  real life*)
```

$$\text{dataneu} = \text{Transpose}[\text{data}] * \left\{\frac{(n - m)}{(EX - UX)}, \frac{(s - r)}{(EY - UY)}\right\} + \left\{\frac{(m * EX - n * UX)}{(EX - UX)}, \frac{(r * EY - s * UY)}{(EY - UY)}\right\} \,//\, \text{Transpose}$$

0.043299

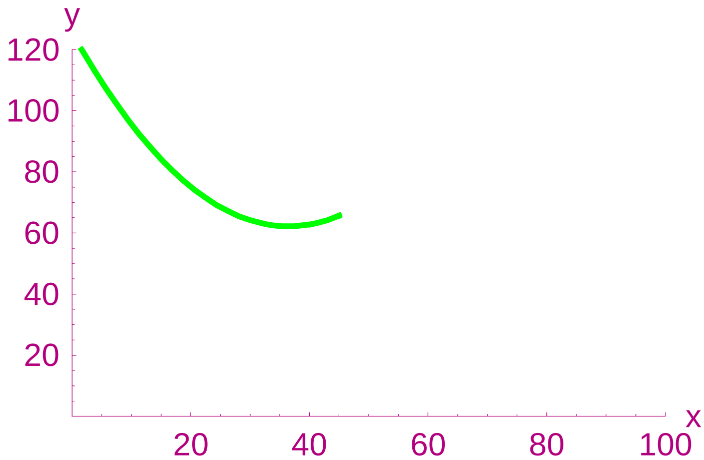
{{4.44089 × 10^{-16}, 120.}, {1.88153, 115.962}, {2.6655, 113.269},
{5.01739, 107.885}, {6.74217, 104.295}, {7.83973, 102.5},
{9.40763, 100.705}, {11.2892, 95.7692}, {13.1707, 91.7307},
{15.3659, 88.1411}, {17.561, 82.7564}, {19.4425, 79.6153},
{20.0696, 78.718}, {21.324, 76.4743}, {22.5784, 73.782},

824.4599, 70.1922<, 825.0871, 69.2949<, 826.8118, 66.6027<,
827.439, 66.1539<, 828.6934, 63.4616<, 829.3205, 63.4616<,
829.9477, 63.0129<, 830.7317, 62.1154<, 832.2996, 60.3206<,
834.0244, 58.9743<, 834.6515, 58.5256<, 835.9059, 56.2821<,
838.101, 54.4871<, 838.885, 55.3173<, 840.9233, 55.6986<,
841.5505, 55.6986<, 843.432, 55.6986<, 845., 104.071<<

Input @ Ramp@x_D = Fit@dataneu, 81, x, x^2<, xD

$125.524 - 3.47137\,x + 0.0475573\,x^2$

Input @ rampimg = Plot@Ramp@xD, 8x, 0, 45<, PlotRange Ø 880, 100<, 80, 120<<,
PlotStyle Ø 88 Green, Thickness@.01D, Dashing@80., 0.<D< <D



Ü Graphics Ü

UX = 0.641237
EX = 0.830928;
UY = 0.195876;
EY = 0.14433;
m = 30;H* it is the starting value of x in real life*L
n = 100;H* it is the
x- value of your maximum in real life*L
Input @ s = 0;H* it is the
y- value of your maximum in real life*L
r = 45;H* it is the starting value of y in real life*L

dataneu1 = Transpose@data2D * 9 Hn - mL / HEX - UXL , Hs - rL / HEY - UYL = +
9 Hm * EX - n * UXL / HEX - UXL , Hr * EY - s * UYL / HEY - UYL = êê Transpose

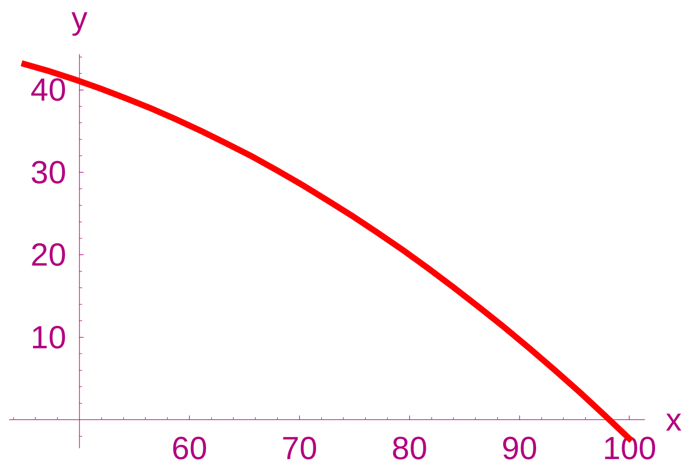
0.641237

8830., 45.<, 839.8912, 45.<, 849.0219, 45.<, 857.3913, 39.6005<,
867.2826, 27.0003<, 870.3259, 27.0003<, 875.652, 25.2002<,
881.739, 17.9997<, 890.8697, 7.19969<, 894.6739, 3.60028<, 8100., 0.<<

Input @ Flight@x_D = Fit@dataneu1, 81, x, x^2<, xD

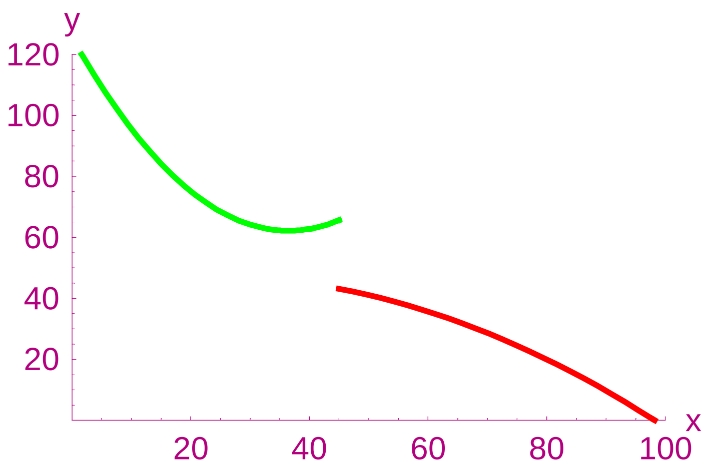
$43.2696 + 0.367301\,x - 0.00822813\,x^2$

Input @ flightimg = Plot@Flight@xD, 8x, 45, 100<,
PlotStyle Ø 88 Red, Thickness@.01D, Dashing@80., 0.<D< <D



Ü Graphics Ü

Input @ Show@rampimg, flightimgD



Ü Graphics Ü

The jump and flight of the ski jumper.

data10 = 8844.9729, 69.4644<,
847.2169, 71.2073<, 849.4609, 73.8215<, 852.1537, 73.8215<,
854.8466, 73.8215<, 857.5394, 72.9501<, 859.3346, 72.0787<,

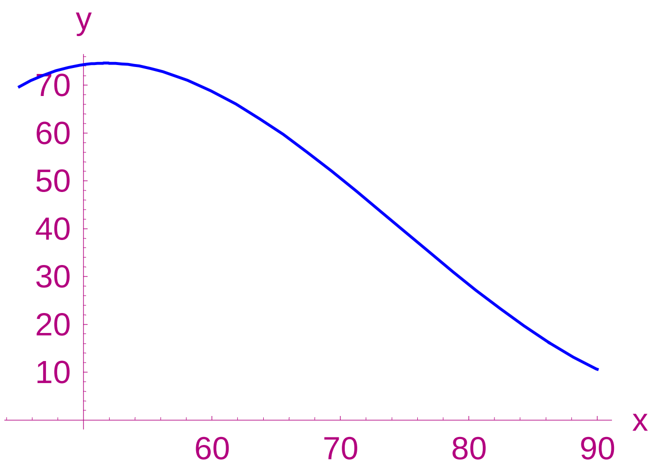
861.1299, 68.593<, 863.8227, 65.9788<, 865.6179, 61.6217<,
867.4131, 57.2646<, 869.2084, 52.036<, 869.2084, 48.5504<,
871.4524, 42.4504<, 875.0428, 36.3505<, 877.2869, 30.2505<,
880.4285, 26.7649<, 882.6725, 23.2792<, 884.0189, 20.6649<,
886.263, 18.9221<, 887.1606, 17.1793<, 889.8534, 12.8222<,
892.0974, 11.0793<, 895.2391, 5.8508<, 897.9319, 1.4937<<

8844.9729, 69.4644<, 847.2169, 71.2073<,
849.4609, 73.8215<, 852.1537, 73.8215<,
854.8466, 73.8215<, 857.5394, 72.9501<, 859.3346, 72.0787<,
861.1299, 68.593<, 863.8227, 65.9788<, 865.6179, 61.6217<,
867.4131, 57.2646<, 869.2084, 52.036<, 869.2084, 48.5504<,
871.4524, 42.4504<, 875.0428, 36.3505<, 877.2869, 30.2505<,
880.4285, 26.7649<, 882.6725, 23.2792<, 884.0189, 20.6649<,
886.263, 18.9221<, 887.1606, 17.1793<, 889.8534, 12.8222<,
892.0974, 11.0793<, 895.2391, 5.8508<, 897.9319, 1.4937<<

Input @ Jump1@x_D = Fit@data10, 81, x, x^2, x^3<, xD

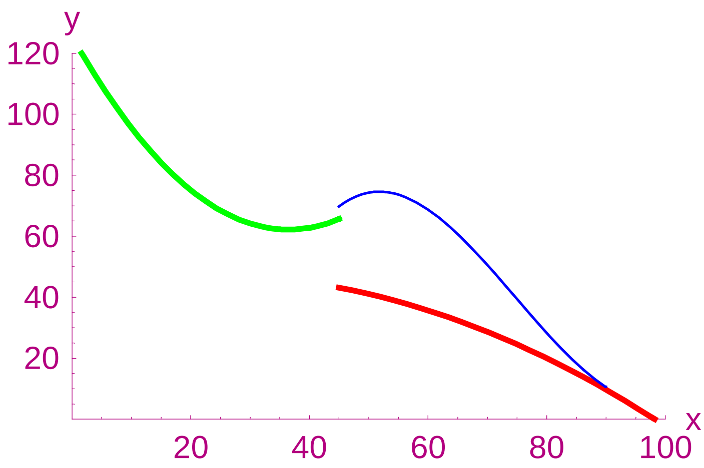
$$-386.714 + 21.6487\,x - 0.319906\,x^2 + 0.00142684\,x^3$$

Input @ jump1img = Plot@Jump1@xD, 8x, 45, 90<,
PlotStyle Ø 88Blue, Thickness@.005D, Dashing@80., 0.<D<<D



Ü Graphics Ü

Input @ Show@rampimg, flightimg, jump1imgD



Ü Graphics Ü

Input @

```
data11 = 8844.9729, 68.593<,
    846.7681, 72.9501<, 849.4609, 72.9501<, 851.2561, 72.9501<,
    853.949, 72.9501<, 856.6418, 72.9501<, 859.3346, 72.9501<,
    862.0275, 72.0787<, 865.6179, 67.7216<, 868.3107, 64.2359<,
    868.3107, 64.2359<, 871.0036, 59.0074<, 873.6964, 52.036<,
    876.3892, 45.0647<, 879.0821, 41.579<, 879.9797, 38.0933<,
    880.4285, 36.3505<, 882.6725, 28.5077<, 884.9165, 21.5364<,
    886.7118, 17.1793<, 890.3022, 13.6936<, 893.4439, 10.2079<,
    895.6879, 6.72222<, 897.0343, 4.97938<, 898.8295, 1.4937<<
```

```
8844.9729, 68.593<, 846.7681, 72.9501<,
  849.4609, 72.9501<, 851.2561, 72.9501<,
  853.949, 72.9501<, 856.6418, 72.9501<, 859.3346, 72.9501<,
  862.0275, 72.0787<, 865.6179, 67.7216<, 868.3107, 64.2359<,
  868.3107, 64.2359<, 871.0036, 59.0074<, 873.6964, 52.036<,
  876.3892, 45.0647<, 879.0821, 41.579<, 879.9797, 38.0933<,
  880.4285, 36.3505<, 882.6725, 28.5077<, 884.9165, 21.5364<,
  886.7118, 17.1793<, 890.3022, 13.6936<, 893.4439, 10.2079<,
  895.6879, 6.72222<, 897.0343, 4.97938<, 898.8295, 1.4937<<
```

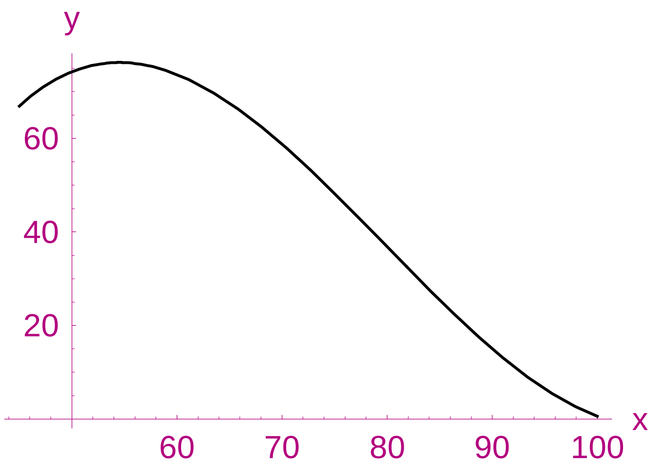
Input @

```
Jump2@x_D = Fit@data11, 81, x, x^2, x^3<, xD
```

$$-390.36 + 20.6993\, x - 0.288389\, x^2 + 0.00120493\, x^3$$

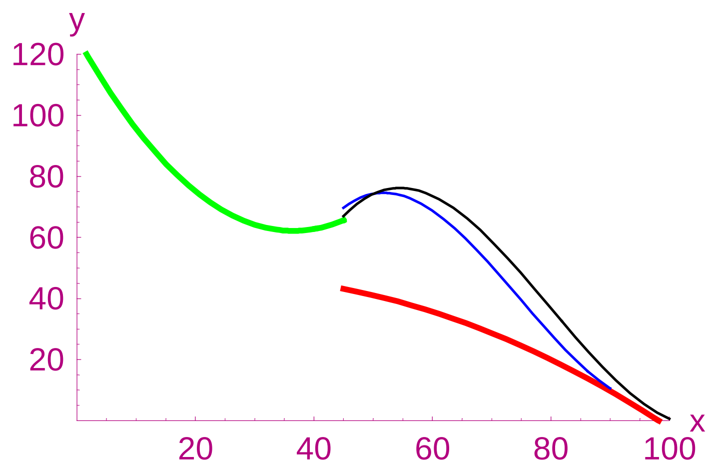
Input @

```
jump2img = Plot@Jump2@xD, 8x, 45, 100<,
PlotStyle Ø 88Black, Thickness@.005D, Dashing@80., 0.<D<<D
```



Ü Graphics Ü

Input @ Show@rampimg, flightimg, jump1img, jump2imgD



Ü Graphics Ü

Input @ data12 = 88 44.9729, 67.7216<, 8 46.3193, 71.2073<, 8 49.9097, 71.2073<,
8 52.6025, 71.2073<, 8 54.8466, 71.2073<, 8 57.9882, 65.9788<,
8 59.7834, 62.4931<, 8 61.5787, 58.136<, 8 62.9251, 57.2646<,
8 63.8227, 54.6503<, 8 65.6179, 49.4218<, 8 68.3107, 42.4504<,
8 70.5548, 32.8648<, 8 71.9012, 30.2505<, 8 79.5309, 22.4078<,
8 81.3261, 20.6649<, 8 83.1213, 17.1793<, 8 86.263, 14.565<,
8 89.8534, 12.8222<, 8 92.0974, 8.46506<, 8 93.4439, 10.2079<,
8 94.7903, 6.72222<, 8 96.1367, 4.10796<, 8 100, 0<, 8 102, - 10<<

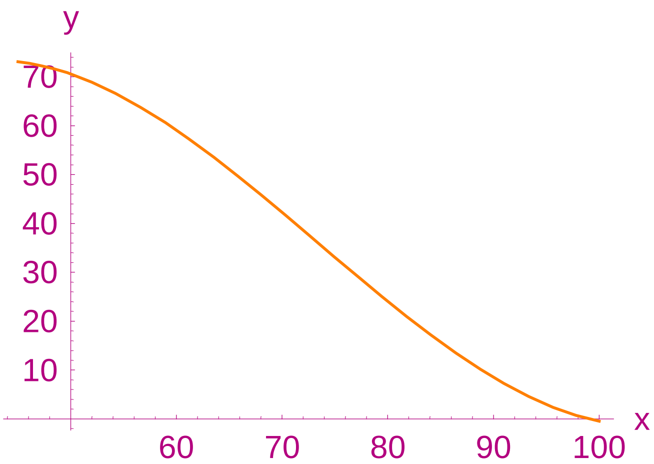
88 44.9729, 67.7216<, 8 46.3193, 71.2073<, 8 49.9097, 71.2073<,
8 52.6025, 71.2073<, 8 54.8466, 71.2073<, 8 57.9882, 65.9788<,
8 59.7834, 62.4931<, 8 61.5787, 58.136<, 8 62.9251, 57.2646<,
8 63.8227, 54.6503<, 8 65.6179, 49.4218<, 8 68.3107, 42.4504<,
8 70.5548, 32.8648<, 8 71.9012, 30.2505<, 8 79.5309, 22.4078<,

881.3261, 20.6649<, 883.1213, 17.1793<, 886.263, 14.565<,
889.8534, 12.8222<, 892.0974, 8.46506<, 893.4439, 10.2079<,
894.7903, 6.72222<, 896.1367, 4.10796<, 8100, 0<, 8102, - 10<<

Input @ Jump3@x_D = Fit@data12, 81, x, x^2, x^3<, xD

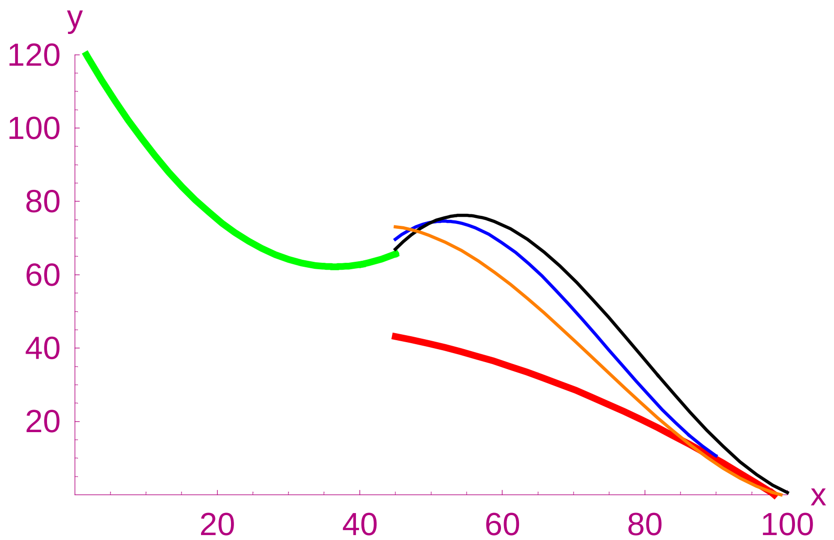
$-91.264 + 8.87985\,x - 0.146004\,x^2 + 0.000662823\,x^3$

Input @ jump3img = Plot@Jump3@xD, 8x, 45, 100<,
PlotStyle Ø 88Orange, Thickness@.005D, Dashing@80., 0.<D<<D



Ü Graphics Ü

Input @ Show@rampimg, flightimg, jump1img, jump2img, jump3imgD



Ü Graphics Ü

data100 = 8843.3789, 67.1717<, 845.2458, 70.7967<, 848.9797, 74.4217<,
850.4733, 74.4217<, 851.9669, 74.4217<, 853.4604, 74.4217<,
856.821, 73.6967<, 860.5549, 72.2467<, 863.1686, 70.7967<,
867.276, 67.1717<, 869.8897, 62.8218<, 871.3833, 60.6468<,

871.7567, 60.6468<, 874.3704, 55.5718<, 877.3575, 49.0469<,
879.2245, 43.2469<, 881.8383, 33.097<, 883.3318, 25.1221<,
886.6923, 17.1471<, 889.3061, 13.5222<, 893.04, 6.99723<,
894.5336, 5.54724<, 896.4005, 3.37226<, 897.8941, 0.472286<,
899.0143, - 2.42769<, 8100.881, - 5.32766<, 8102.375, - 7.50265<<

8843.3789, 67.1717<, 845.2458, 70.7967<, 848.9797, 74.4217<,
850.4733, 74.4217<, 851.9669, 74.4217<, 853.4604, 74.4217<,
856.821, 73.6967<, 860.5549, 72.2467<, 863.1686, 70.7967<,
867.276, 67.1717<, 869.8897, 62.8218<, 871.3833, 60.6468<,
871.7567, 60.6468<, 874.3704, 55.5718<, 877.3575, 49.0469<,
879.2245, 43.2469<, 881.8383, 33.097<, 883.3318, 25.1221<,
886.6923, 17.1471<, 889.3061, 13.5222<, 893.04, 6.99723<,
894.5336, 5.54724<, 896.4005, 3.37226<, 897.8941, 0.472286<,
899.0143, - 2.42769<, 8100.881, - 5.32766<, 8102.375, - 7.50265<<

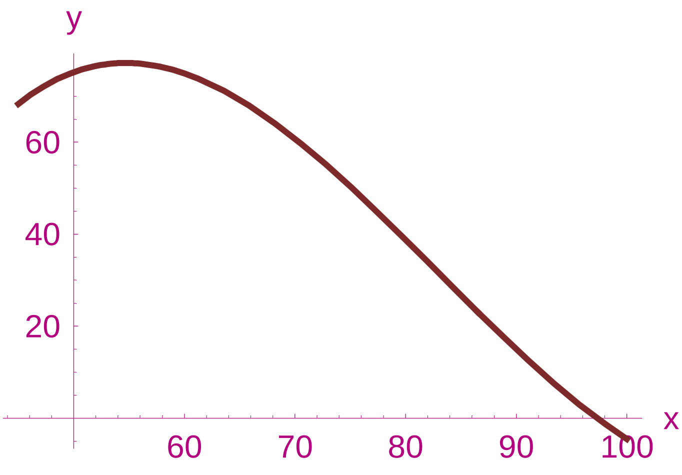
Input @ FinalJump@x_D = Fit@data100, 81, x, x^2, x^3<, xD

$-342.212 + 18.3519\, x - 0.250354\, x^2 + 0.00100607\, x^3$

Input @ FinalJump1@x_D = Fit@data100, 81, x, x^2<, xD

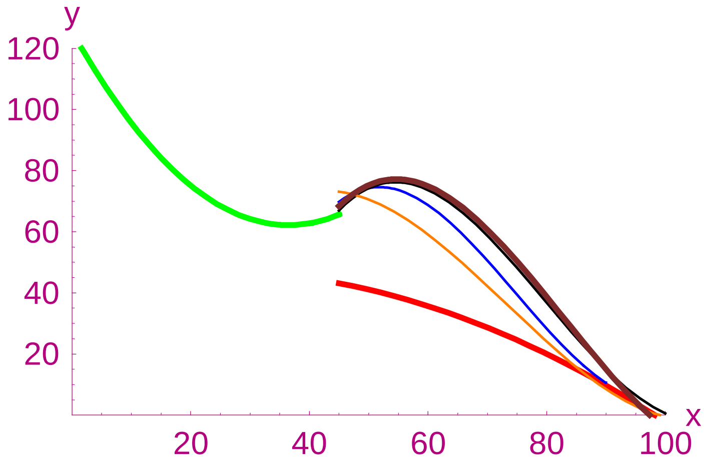
$9.9422 + 2.74407\, x - 0.0292397\, x^2$

Input @ finaljumpimg = Plot@FinalJump@xD, 8x, 45, 100<,
PlotStyle Ø 88 Brown, Thickness@.01D, Dashing@80., 0.<D<<D



ÜGraphicsÜ

Input @ Show@rampimg, flightimg, jump1img, jump2img, jump3img, finaljumpimgD



Ü Graphics Ü

Finding the reflection point.

Input @ NSolve@Jump1''@xD ã 0, xD

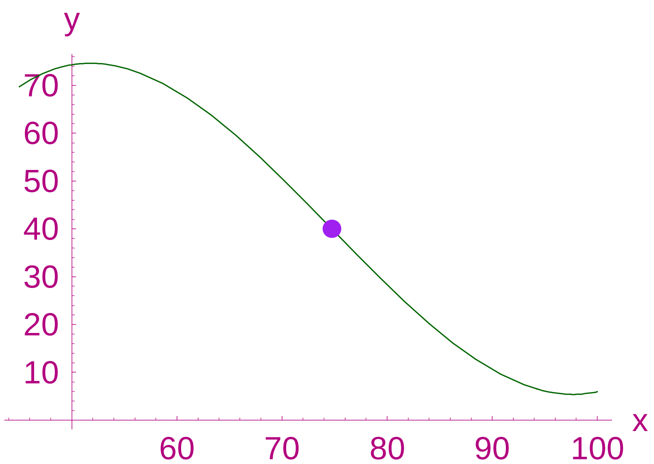
88x Ø 74.7352337789981<<

Input @ Jump1@74.73523377899810427409413`15.4028D

40.0108

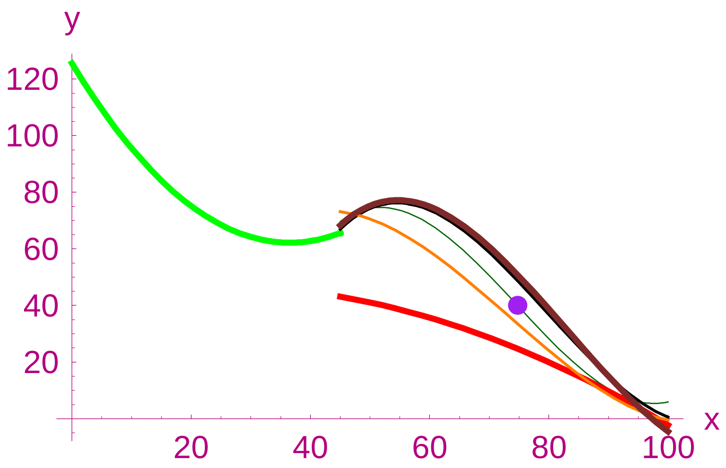
Input @

Input @ ref1img =
MDPlot@8Jump1@xD<, 8x , 45 , 100 <, Epilog Ø 8 Purple, PointSize@0.03D ,
Point@8 74.73523377899810427409413`15.4028,
40.010834295666655` <D < D



Ü Graphics Ü

Input @ Show@ref1img, rampimg, flightimg, jump2img, jump3img, finaljumpimgD



Ü Graphics Ü

Finding the reflection point for the other jumps (without the graphs).

Input @ NSolve@Jump2''@xD ã 0, xD

88x Ø 79.7804814508412<<

Input @ Jump2@79.78048145084117918533092 6`15.3824D

37.3212

Input @ NSolve@Jump3''@xD ã 0, xD

88x Ø 73.4253595794036<<

Input @ Jump3@73.42535957940362514317364`15.2202D

35.9764

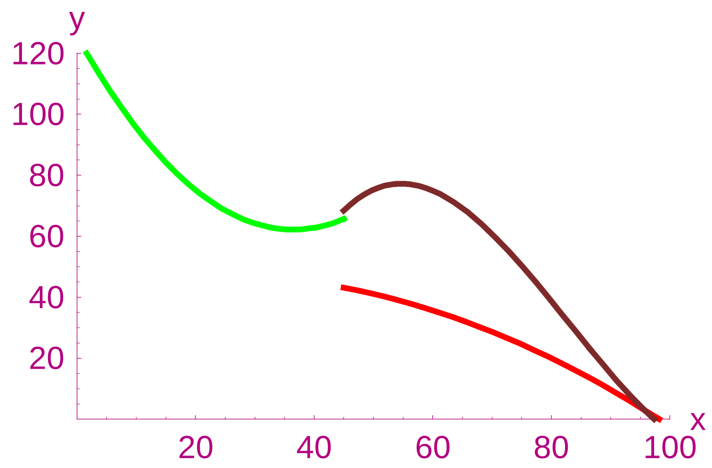
Input @ NSolve@FinalJump''@xD ã 0, xD

88x Ø 82.9482956920179<<

Input @ FinalJump@82.94829569201794176956194 4`15.3528D

31.6816

Input @ Show@rampimg, flightimg, finaljumpimgD



Ü Graphics Ü

Getting the angle of the jump.

Input @ BestJump@x_D = - 342.2116849935606` + 18.351855004875652` x - 0.2503543010886187` $x^2$ + 0.001006065682242061` x^3

- 342.212 + 18.3519 x - 0.250354 x^2 + 0.00100607 x^3

Input @ BestJump'@45D

1.93182

Input @ NSolve@69.5 ã 1.9318169265204892` * 45 + d, dD

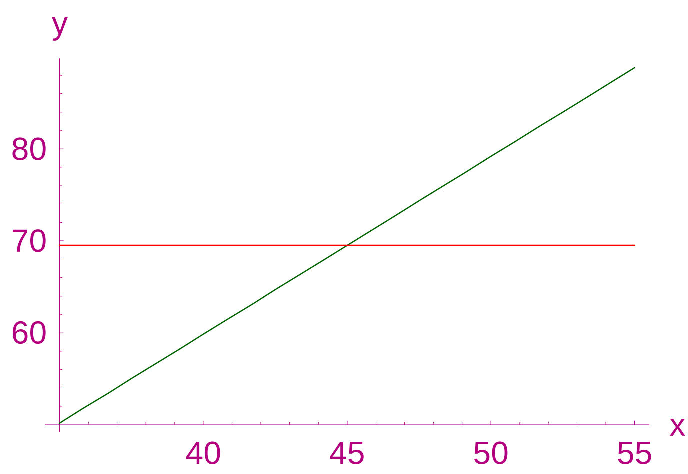
88d Ø - 17.43176169342202<<

Input @ Clear@TangentD

Input @ Tangent@x_D = 1.9318169265204892` x + - 17.4317616934220183111392543651

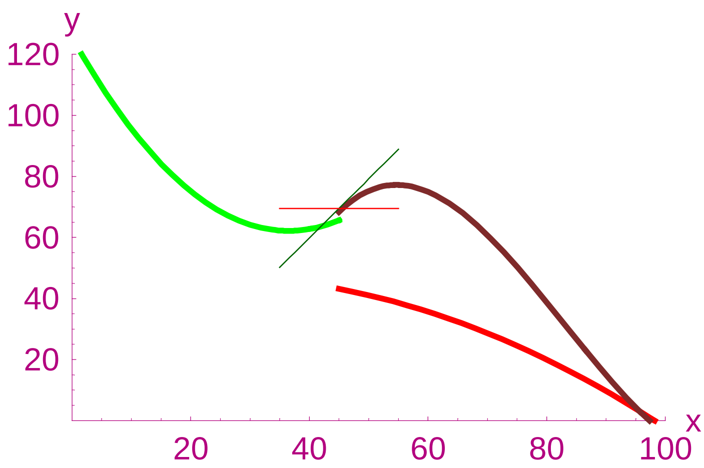
- 17.4317616934220183111392543651 + 1.93182 x

Input @ Tangentimg = MDPlot@8Tangent@xD, 69.5<, 8x , 35 , 55 <D



ÜGraphicsÜ

Input @ Show@rampimg, flightimg, finaljumpimg, TangentimgD



ÜGraphicsÜ

Input @

m1 = **0**;

m2 = **1.9**;

j = ArcTanA AbsA $\frac{m2\ -\ m1}{1 + m2 * m1}$ EE í ° êê N

62.2415

Input @ Bestjump1@x_D = - 386.71390307804813` + 21.648669817120318` x - 0.3199063335947178` $x^2$ + 0.0014268429914450743` x^3

- 386.714 + 21.6487 x - 0.319906 x^2 + 0.00142684 x^3

Input @ Bestjump1'@45D

1.52517

Input @

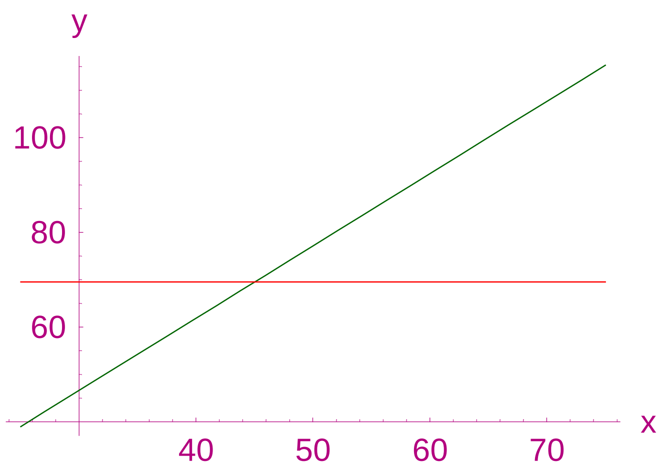
Input @ NSolve@69.5 ã 1.5251709666245397`* 45 + d, dD

88d Ø 0.867306501895712<<

Input @ Tangent2@x_D =
1.5251709666245397` x + 0.867306501895711789984488859772682

0.867306501895711789984488859772682 + 1.52517 x

Input @ MDPlot@8Tangent2@xD, 69.5<, 8x , 25 , 75<D



Ü Graphics Ü

Input @
m1 = 0;
m2 = 1.5251709666245397`;
j = ArcTanA AbsA (m2 - m1)/(1 + m2 * m1) EE í ° êê N

56.7486

Input @ Bestjump3@x_D = - 91.26396897041042` + 8.879851878006809` x - 0.14600400501852454` $x^2$ + 0.0006628227897230927` x^3

- 91.264 + 8.87985 x - 0.146004 x^2 + 0.000662823 x^3

Input @ Bestjump3'@45D

- 0.23386

Input @ data9 = 8844.0752, 68.593<,
846.7681, 69.4644<, 849.9097, 69.4644<, 853.0514, 69.4644<,
857.9882, 65.1073<, 860.681, 63.3645<, 862.9251, 59.0074<,
865.1691, 55.5217<, 868.3107, 51.1646<, 870.5548, 47.6789<,
875.4916, 38.9647<, 879.5309, 31.122<, 882.6725, 25.8935<,
886.7118, 19.7935<, 888.507, 15.4364<, 890.3022, 9.33648<,
888.507, 9.33648<, 894.7903, 5.8508<, 897.9319, 3.23654<<

8844.0752, 68.593<, 846.7681, 69.4644<,
849.9097, 69.4644<, 853.0514, 69.4644<,
857.9882, 65.1073<, 860.681, 63.3645<, 862.9251, 59.0074<,
865.1691, 55.5217<, 868.3107, 51.1646<, 870.5548, 47.6789<,
875.4916, 38.9647<, 879.5309, 31.122<, 882.6725, 25.8935<,
886.7118, 19.7935<, 888.507, 15.4364<, 890.3022, 9.33648<,
888.507, 9.33648<, 894.7903, 5.8508<, 897.9319, 3.23654<<

Input @ PerfectJump@x_D = Fit@data9, 81, x, x^2, x^3<, xD

$-192.579 + 12.5969\,x - 0.187097\,x^2 + 0.000804285\,x^3$

Input @ PerfectJump'@45D

0.644234

Input @ NSolve@69.5 ã 0.644233647878738` * 45 + d, dD

88d Ø 40.50948584545679<<

Input @ perfectangent@x_D =
0.644233647878738` x + 40.509485845456786989871034165 8`15.6482

40.50948584545679 + 0.644234 x

Input @ perfection = MDPlot@8perfectangent@xD, 69.5<, 8x , 30 , 60 <D

Ü Graphics Ü

Input @
m1 = 0;
m2 = 0.644233647878738`;
j = ArcTan A Abs A $\frac{m2 - m1}{1 + m2 * m1}$ EE í ° êê N

32.791

```
data8 = 8842.28, 65.9788<, 850.3585, 65.9788<,
    852.1537, 65.1073<, 854.3978, 64.2359<,
    857.0906, 62.4931<, 859.3346, 61.6217<, 861.1299, 60.7502<,
    863.3739, 59.0074<, 866.5155, 55.5217<, 867.4131, 52.036<,
    869.2084, 48.5504<, 871.0036, 46.8075<, 874.1452, 43.3218<,
    874.594, 39.8362<, 877.2869, 35.4791<, 879.0821, 31.122<,
    881.3261, 29.3791<, 883.5701, 25.8935<, 884.9165, 22.4078<,
    887.6094, 19.7935<, 888.9558, 16.3078<, 890.751, 14.565<,
    892.0974, 11.0793<, 892.995, 10.2079<, 894.3415, 6.72222<<
```

```
8842.28, 65.9788<, 850.3585, 65.9788<, 852.1537, 65.1073<, 854.3978, 64.2359<,
  857.0906, 62.4931<, 859.3346, 61.6217<, 861.1299, 60.7502<,
  863.3739, 59.0074<, 866.5155, 55.5217<, 867.4131, 52.036<,
  869.2084, 48.5504<, 871.0036, 46.8075<, 874.1452, 43.3218<,
  874.594, 39.8362<, 877.2869, 35.4791<, 879.0821, 31.122<,
  881.3261, 29.3791<, 883.5701, 25.8935<, 884.9165, 22.4078<,
  887.6094, 19.7935<, 888.9558, 16.3078<, 890.751, 14.565<,
  892.0974, 11.0793<, 892.995, 10.2079<, 894.3415, 6.72222<<
```

Input @ `FinalPerfectJump@x_D = Fit@data8, 81, x, x^2, x^3<, xD`

$-106.138 + 8.30577\,x - 0.121364\,x^2 + 0.000488723\,x^3$

Input @ `FinalPerfectJump'@45D`

0.352001

Input @ `NSolve@69.5 ã 0.35200056063936236`* 45 + d, dD`

88d Ø 53.6599747712287<<

Input @
```
Finalperfecttangent@x_D =
  0.35200056063936236` x + 53.659974771228689860436134040 4`15.6495
```

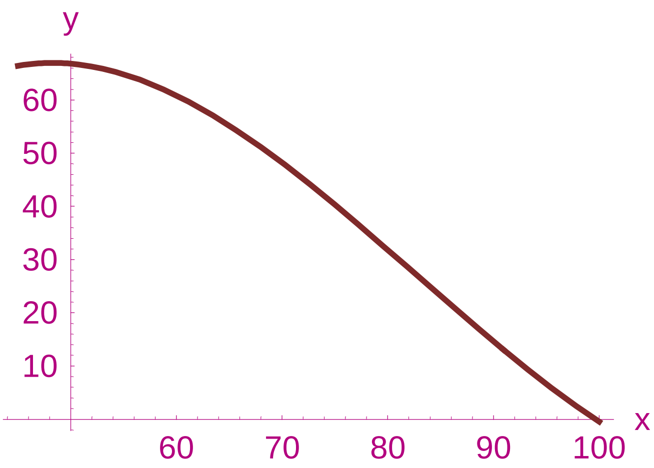
53.6599747712287 + 0.352001 x

Finding the angle.

Input @
```
m1 = 0;
m2 = 0.35200056063936236`;
j = ArcTanA AbsA (m2 - m1)/(1 + m2 * m1) EE í ° êê N
```

19.3921

Input @ finally = Plot@FinalPerfectJump@xD, 8x, 45, 100<,
PlotStyle Ø 88Brown, Thickness@.01D, Dashing@80., 0.<D<<D



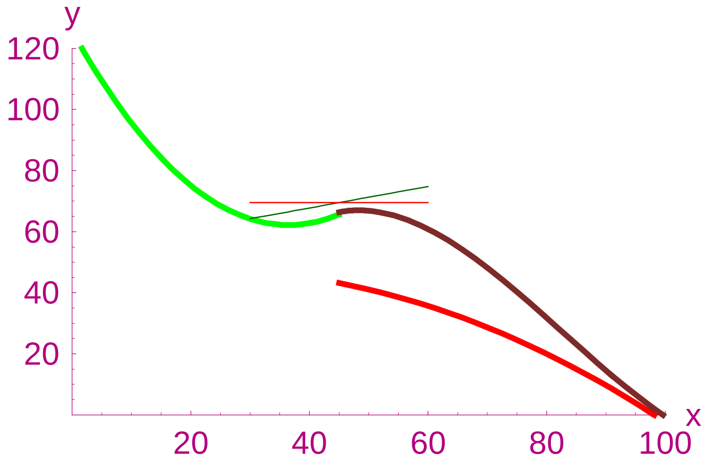
Ü Graphics Ü

Input @ great = MDPlot@8Finalperfecttangent@xD, 69.5<, 8x , 30 , 60<D

Ü Graphics Ü

And the result is:

Input @ Show@rampimg, flightimg, finally, greatD



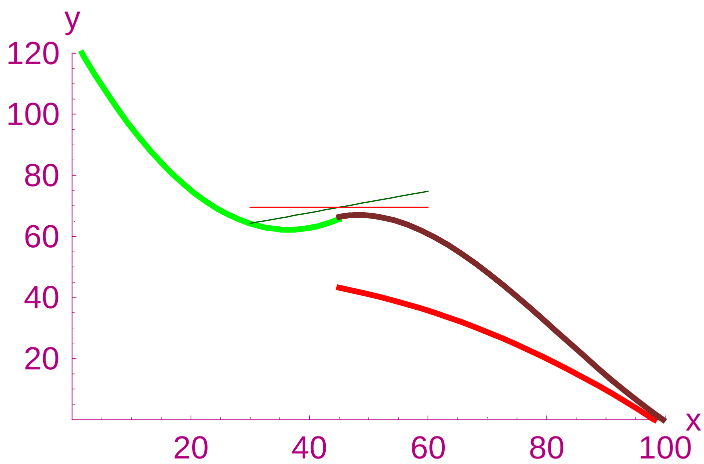
Ü Graphics Ü

Input @ DONE!

4 Result and Summary

Open ê Close Print

Section:
Open ê Close



1. *Mathematica* and M@th Desktop allows us to transfer the data from pictures to the PC. We have just found the ski jumping base equation or even the ski jumper´s one when he´s jumping. It wasn´t easy though, we had to be really carefull when it comes to the angle of the jump. After a couple of tries, 60s or 40s we finally managed to get a 19 degrees angle, and it worked just like a charm. We´ve all learned many new functions of the program, and the project was really interesting for us :)

5

Our Team

Open ê Close Print

Participants

Open ê Close

The members of our group:
- Adrian Trzeciak
- Fabio Juritsch
- Nikolas Pock
- João Joaquim
- Diogo Justino
- Joana Silva

Our Experience with the Project

Open ê Close

1. The project was a great idea. Most of us are going to use this program for the rest of our lifes. We meet new, interesting or boring people, mostly interesting. The project was completely great, the days when we were at school were amazing. It was kind of interesting to see the portugal students at school and the differences between the countries.

New Chapter Cut Last Chapter

INTERNET W3 Open Close Print