



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



1

Description of the Project[Open](#) [Close](#)[Print](#)**Introduction:**[Open](#) [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 graphical output (maybe with an animation) and a palette.

2

Brainstorming and Theory[Open](#) [Close](#)[Print](#)**Brainstorming**[Open](#) [Close](#)

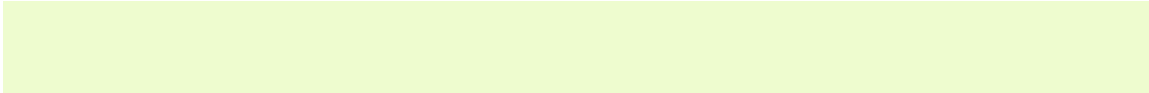
1.

At first, we have to search for useful pictures.
We also need to find out the tangent 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](#) [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 & Close

Print

Section:

Open & Close



The picture we´ve chosen:



Input @ We get the points from the picture

```

data = 80.043299, 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<,
Input @ 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<<

```

```

80.043299, 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@_D = MDNonlinearFit@data,
          a x2 + b x + c, H* model *L
          &x<, &a, b, c<H* parameters *L
          D @@ Chop@#, 10-5D &;

```

Input @ &start, stop< = &Min@#D, Max@#D< &@Firstü Transposeü dataD;

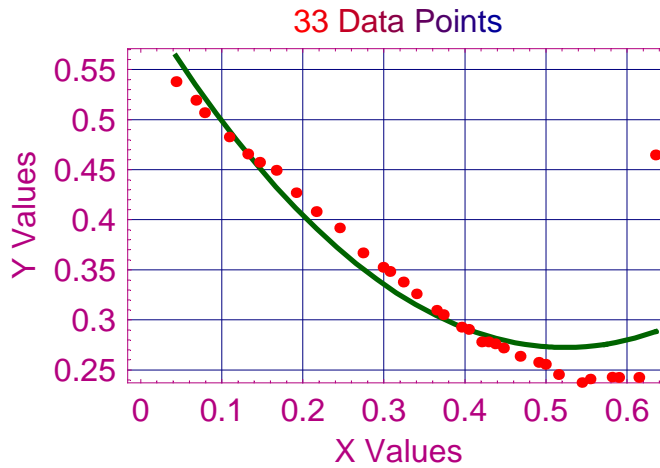
```

MDPlotFitData@data, &fit@D<, &x, start, stop<,
  FrameLabel Ø &"X Values", "Y Values"<,
  Epilog Ø &Red, PointSize@0.02D, Point êü data<,
  PlotStyle Ø &&DarkGreen, Thickness@0.01D<<D

```

, Hy_i - y_i² Sum of Squared Error : 0.0429225

0.618415 - 1.32242 x + 1.2637 x²



Graphics

```

Input @
fit_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 = {{0.641237, 0.195876}, {0.668041, 0.195876},
{0.692784, 0.195876}, {0.715464, 0.189691}, {0.742268, 0.175258},
{0.750515, 0.175258}, {0.764948, 0.173196}, {0.781443, 0.164948},
{0.806186, 0.152577}, {0.816495, 0.148454}, {0.830928, 0.14433}}
    
```

```

{{0.641237, 0.195876}, {0.668041, 0.195876},
{0.692784, 0.195876}, {0.715464, 0.189691}, {0.742268, 0.175258},
{0.750515, 0.175258}, {0.764948, 0.173196}, {0.781443, 0.164948},
{0.806186, 0.152577}, {0.816495, 0.148454}, {0.830928, 0.14433}}
    
```

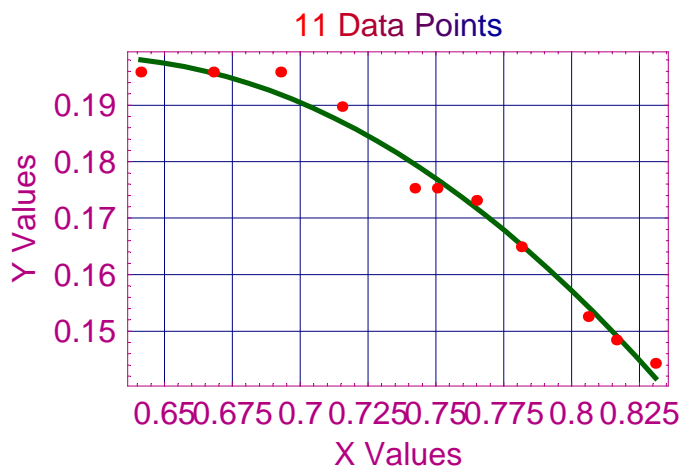
```

fit_D = MDNonlinearFit[data2,
a x^2 + b x + c, {a, b, c}];
Chop[fit_D, 10^-5];
    
```

```

Input @
{start, stop} = {Min[data2[[1]], Max[data2[[1]]], First[Transpose[data2]]];
MDPlotFitData[data2, fit_D, {x, start, stop},
FrameLabel -> {"X Values", "Y Values"},
Epilog -> {Red, PointSize[0.02], Point[data2],
PlotStyle -> {DarkGreen, Thickness[0.01]}},
, {Hy - yi^2 Sum of Squared Error : 0.00006146}
    
```

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



⌋ Graphics ⌋

```
Input @ fit_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<<
```

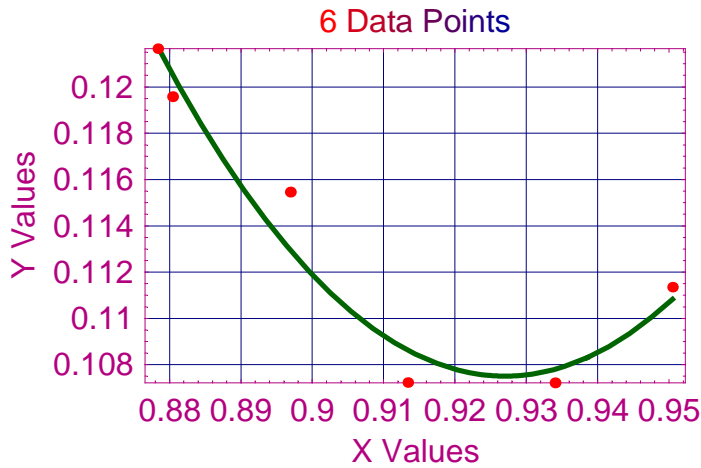
```
880.878351, 0.121649<, 80.880412, 0.119588<, 80.896907, 0.115464<,
80.913402, 0.107216<, 80.934021, 0.107216<, 80.950515, 0.11134<<
```

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

```
Input @ 8start, stop< = 8Min@#D, Max@#D< &@FirstüTransposeü data3D;
MDPlotFitData@data3, 8fit@_D<, 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
```

$$\sum (y_i - \hat{y}_i)^2 \text{ Sum of Squared Error : } 9.82812 \mu 10^{-6}$$

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



⌘ Graphics ⌘

Input @ $f(x) = 5.269612998873093 - 11.137133684303336 x + 6.007031603228448 x^2$
 $5.26961 - 11.1371 x + 6.00703 x^2$

Transferring the coordinates into real meter values.

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

~~dataneu = Transpose@data * 9 $\frac{Hn - ml}{HEX - UXL}$, $\frac{Hs - rl}{HEY - UYL}$ = +~~
~~9 $\frac{Hm * EX - n * UXL}{HEX - UXL}$, $\frac{Hr * EY - s * UYL}{HEY - UYL}$ = ee Transpose~~

0.043299
~~884.44089 μ 10⁻¹⁶, 120.<, 81.88153, 115.962<, 82.6655, 113.269<, 85.01739, 107.885<, 86.74217, 104.295<, 87.83973, 102.5<, 89.40763, 100.705<, 811.2892, 95.7692<, 813.1707, 91.7307<, 815.3659, 88.1411<, 817.561, 82.7564<, 819.4425, 79.6153<, 820.0696, 78.718<, 821.324, 76.4743<, 822.5784, 73.782<~~

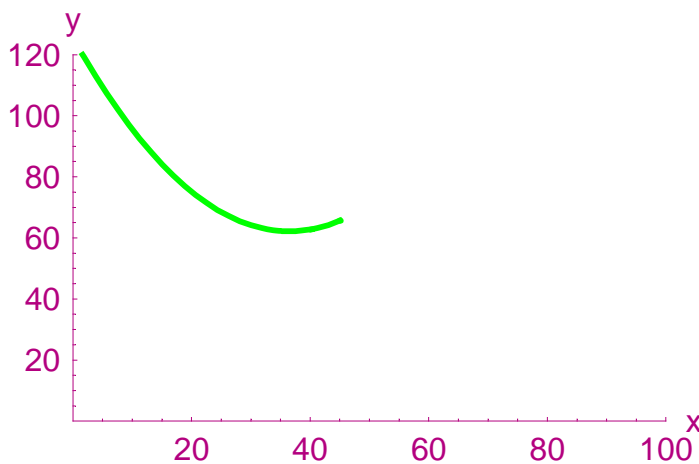
```

24.4599, 70.1922<, 25.0871, 69.2949<, 26.8118, 66.6027<,
27.439, 66.1539<, 28.6934, 63.4616<, 29.3205, 63.4616<,
29.9477, 63.0129<, 30.7317, 62.1154<, 32.2996, 60.3206<,
34.0244, 58.9743<, 34.6515, 58.5256<, 35.9059, 56.2821<,
38.101, 54.4871<, 38.885, 55.3173<, 40.9233, 55.6986<,
41.5505, 55.6986<, 43.432, 55.6986<, 45., 104.071<<
    
```

Input @ Ramp@D = Fit@dataneu, 1, x, x^2<, xD

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

Input @ ramping = Plot@Ramp@D, 1, x, 0, 45<, PlotRange 0 80, 100<, 80, 120<<, PlotStyle 0 88 Green, Thickness@01D, Dashing@0., 0.1<<D



Graphics

```

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 [[m - ml, [hs - rl],
[[EX - UX], [EY - UY]
[[m * EX - n * UX], [r * EY - s * UY]
[[EX - UX], [EY - UY] = ee Transpose
    
```

0.641237

```

830., 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.<<
    
```

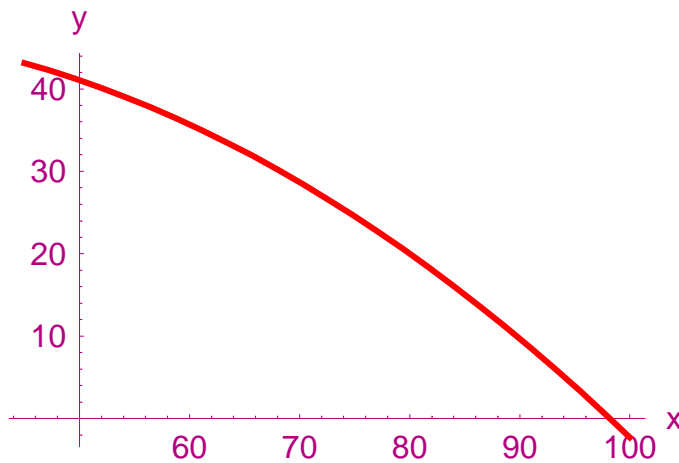
```

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

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

```

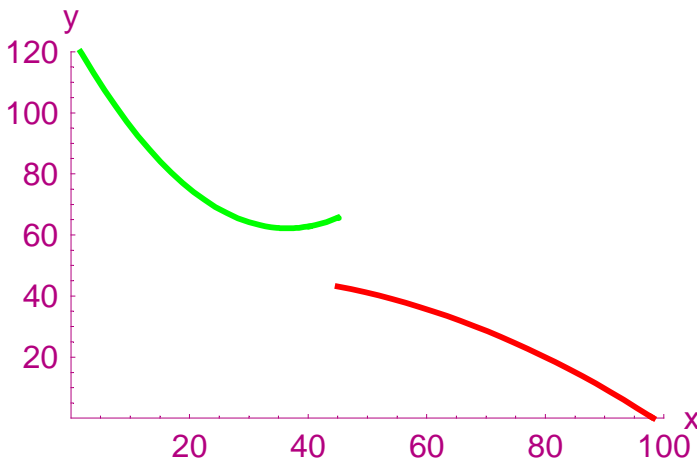
Input @ flightimg = Plot@Flight@xD, 8x, 45, 100<,
PlotStyle @ 88 Red, Thickness@01D, Dashing@0., 0.<<<D
    
```



Graphics

```

Input @ Show@ramping, flightimgD
    
```



Graphics

The jump and flight of the ski jumper.

```

data10 = 844.9729, 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<<

```

```

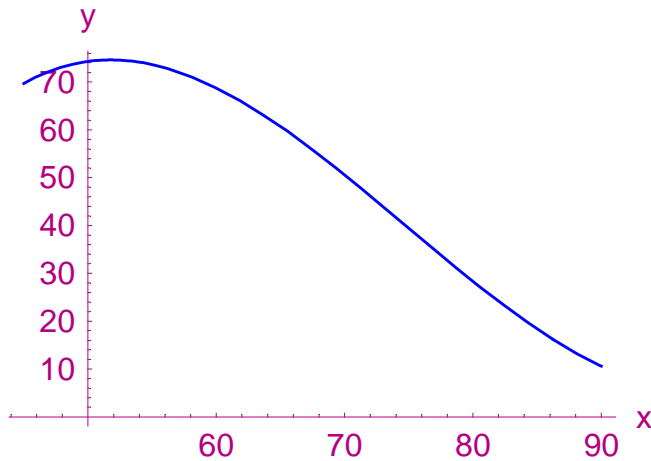
Input @ Jump1@x_D = Fit@data10, 81, x, x^2, x^3<, xD
- 386.714 + 21.6487 x - 0.319906 x^2 + 0.00142684 x^3

```

```

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

```

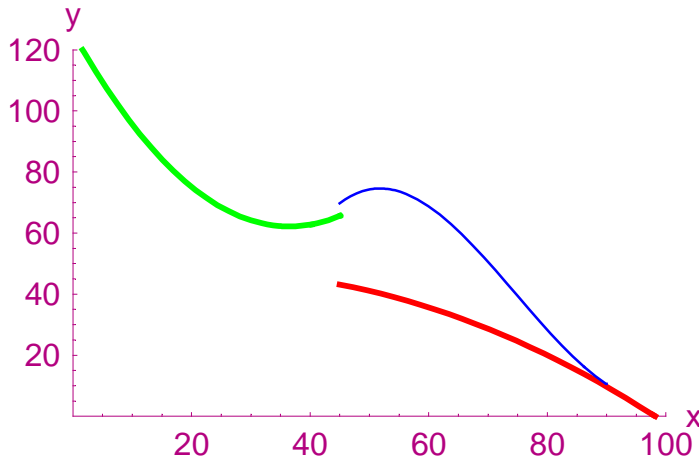


|| Graphics ||

```

Input @ Show@rampimg, flightimg, jump1imgD

```



Graphics

```
data11 = {44.9729, 68.593<,
  {46.7681, 72.9501<, 49.4609, 72.9501<, 51.2561, 72.9501<,
  53.949, 72.9501<, 56.6418, 72.9501<, 59.3346, 72.9501<,
  62.0275, 72.0787<, 65.6179, 67.7216<, 68.3107, 64.2359<,
  68.3107, 64.2359<, 71.0036, 59.0074<, 73.6964, 52.036<,
  76.3892, 45.0647<, 79.0821, 41.579<, 79.9797, 38.0933<,
  80.4285, 36.3505<, 82.6725, 28.5077<, 84.9165, 21.5364<,
  86.7118, 17.1793<, 90.3022, 13.6936<, 93.4439, 10.2079<,
  95.6879, 6.72222<, 97.0343, 4.97938<, 98.8295, 1.4937<<
```

Input @

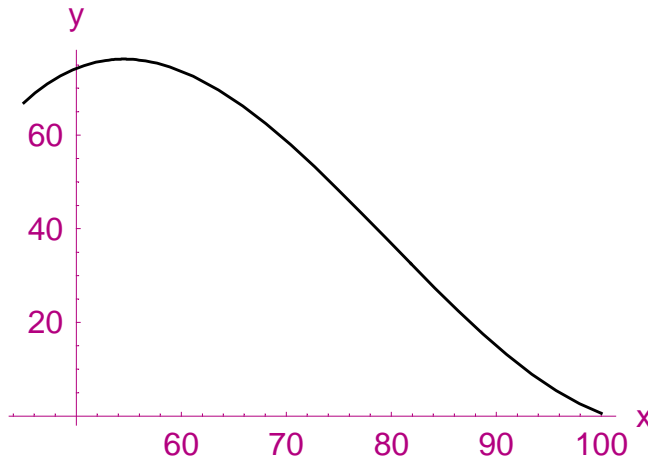
```
{44.9729, 68.593<, 46.7681, 72.9501<,
  49.4609, 72.9501<, 51.2561, 72.9501<,
  53.949, 72.9501<, 56.6418, 72.9501<, 59.3346, 72.9501<,
  62.0275, 72.0787<, 65.6179, 67.7216<, 68.3107, 64.2359<,
  68.3107, 64.2359<, 71.0036, 59.0074<, 73.6964, 52.036<,
  76.3892, 45.0647<, 79.0821, 41.579<, 79.9797, 38.0933<,
  80.4285, 36.3505<, 82.6725, 28.5077<, 84.9165, 21.5364<,
  86.7118, 17.1793<, 90.3022, 13.6936<, 93.4439, 10.2079<,
  95.6879, 6.72222<, 97.0343, 4.97938<, 98.8295, 1.4937<<
```

Input @

```
Jump2@_D = Fit@data11, {1, x, x^2, x^3<, xD
-390.36 + 20.6993 x - 0.288389 x^2 + 0.00120493 x^3
```

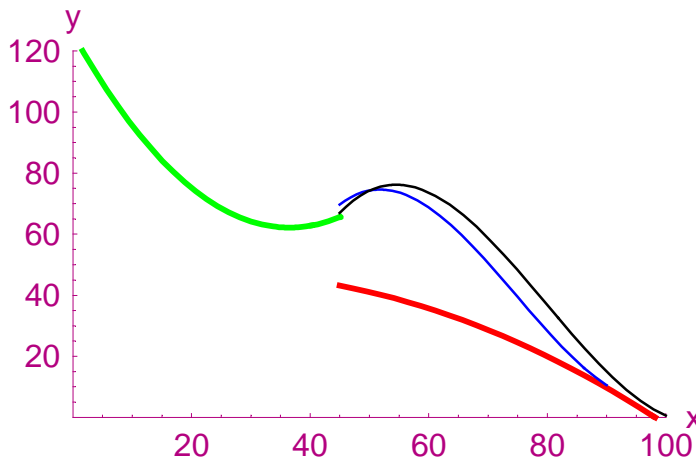
Input @

```
jump2img = Plot@Jump2@_D, {x, 45, 100<,
PlotStyle {Black, Thickness@0.05D, Dashing@{0., 0.}<<
```



⌋ Graphics ⌋

Input @ Show@rampimg, flightimg, jump1img, jump2img



⌋ Graphics ⌋

data12 = 8844.9729, 67.7216<, 846.3193, 71.2073<, 849.9097, 71.2073<,
 852.6025, 71.2073<, 854.8466, 71.2073<, 857.9882, 65.9788<,
 859.7834, 62.4931<, 861.5787, 58.136<, 862.9251, 57.2646<,
 863.8227, 54.6503<, 865.6179, 49.4218<, 868.3107, 42.4504<,
 870.5548, 32.8648<, 871.9012, 30.2505<, 879.5309, 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<<

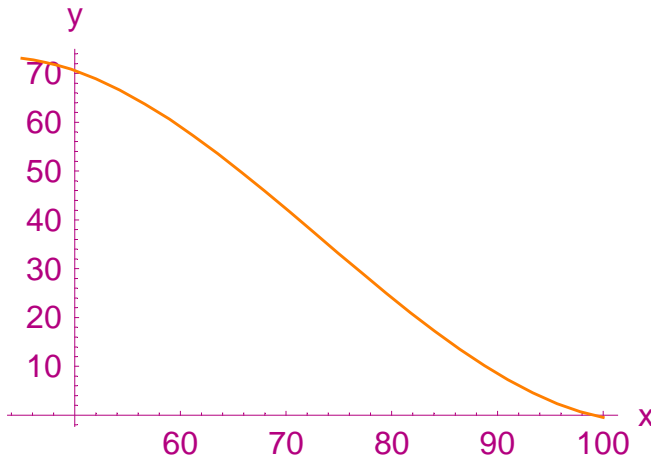
Input @

8844.9729, 67.7216<, 846.3193, 71.2073<, 849.9097, 71.2073<,
 852.6025, 71.2073<, 854.8466, 71.2073<, 857.9882, 65.9788<,
 859.7834, 62.4931<, 861.5787, 58.136<, 862.9251, 57.2646<,
 863.8227, 54.6503<, 865.6179, 49.4218<, 868.3107, 42.4504<,
 870.5548, 32.8648<, 871.9012, 30.2505<, 879.5309, 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<<
```

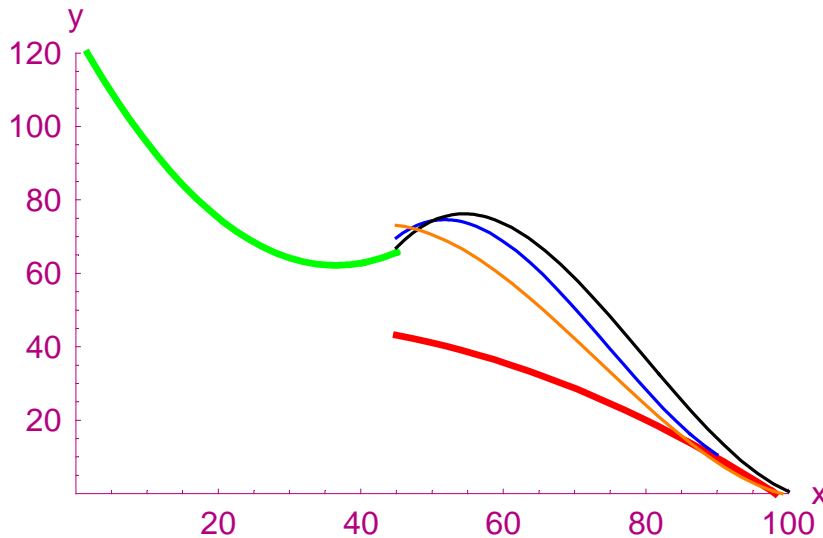
```
Input @ Jump3@x_D = Fit@data12, #1, x, x^2, x^3<, xD
-91.264 + 8.87985 x - 0.146004 x^2 + 0.000662823 x^3
```

```
Input @ jump3img = Plot@Jump3@x, #x, 45, 100<,
PlotStyle {Orange}, Thickness{0.005}, Dashing{0., 0.1}<<
```



Graphics

```
Input @ Show@ramping, flightimg, jump1img, jump2img, jump3imgD
```



Graphics

```
data100 = 843.3789, 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

```

```

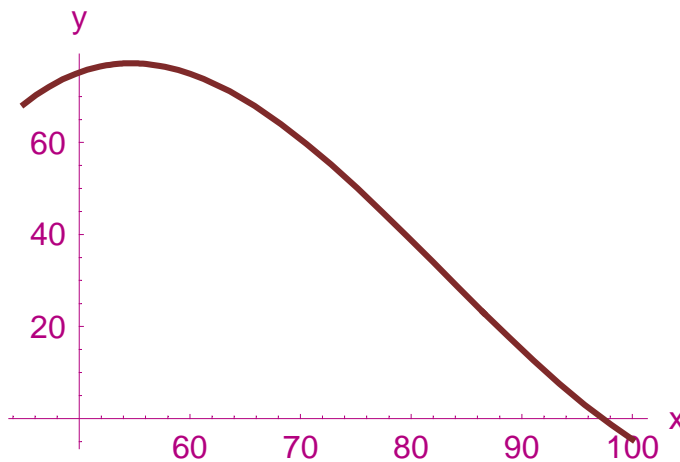
Input @ FinalJump1@x_D = Fit@data100, 81, x, x^2<, xD
9.9422 + 2.74407 x - 0.0292397 x^2

```

```

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

```

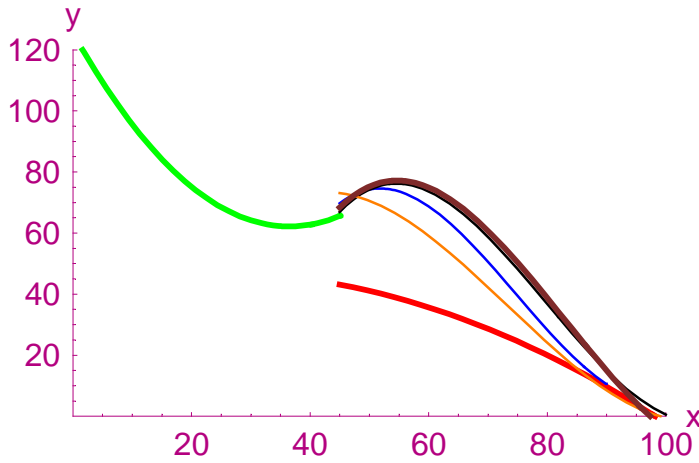


⌋ Graphics ⌋

```

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

```



Graphics

Finding the reflection point.

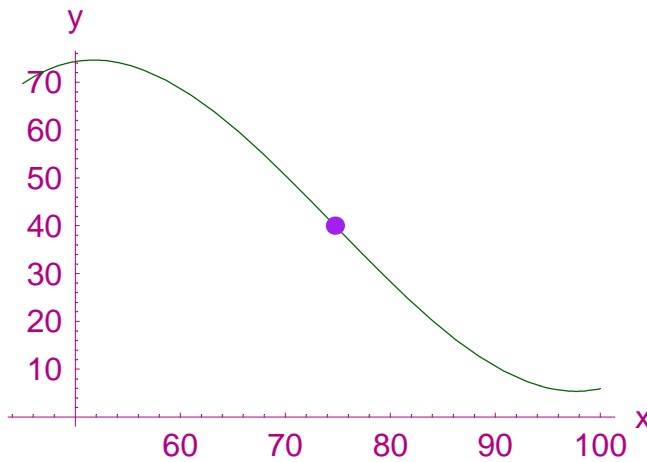
Input @ NSolve@Jump1["x] == 0, x]
 x 74.7352337789981<<

Input @ Jump1@74.73523377899810427409413`15.4028]
 40.0108

Input @

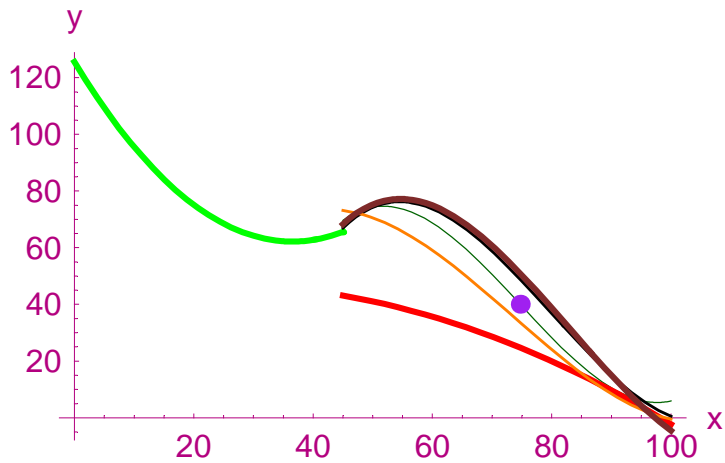
ref1img =

Input @ MDPlot@Jump1["x], {x, 45, 100}, Epilog @ {Purple, PointSize@0.03},
 Point@74.73523377899810427409413`15.4028,
 40.010834295666655`<<<



Graphics

Input @ Show@ref1img, rampimg, flightimg, jump2img, jump3img, finaljumpimg]



¶ Graphics ¶

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

Input @ NSolve@Jump2"@x] ã 0, x]

⚡x Ø 79.7804814508412<<

Input @ Jump2@79.780481450841179185330926`15.3824]

37.3212

Input @ NSolve@Jump3"@x] ã 0, x]

⚡x Ø 73.4253595794036<<

Input @ Jump3@73.42535957940362514317364`15.2202]

35.9764

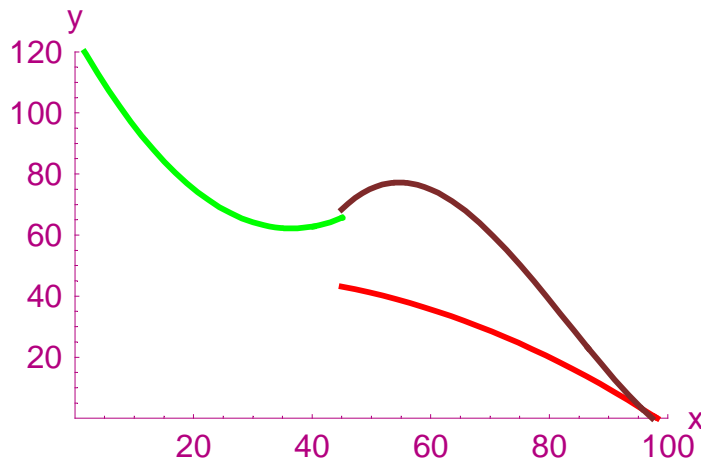
Input @ NSolve@FinalJump"@x] ã 0, x]

⚡x Ø 82.9482956920179<<

Input @ FinalJump@82.948295692017941769561944`15.3528]

31.6816

Input @ Show@ramping, flighting, finaljumping]



Graphics

Getting the angle of the jump.

```
Input @ BestJump[x_] = -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
```

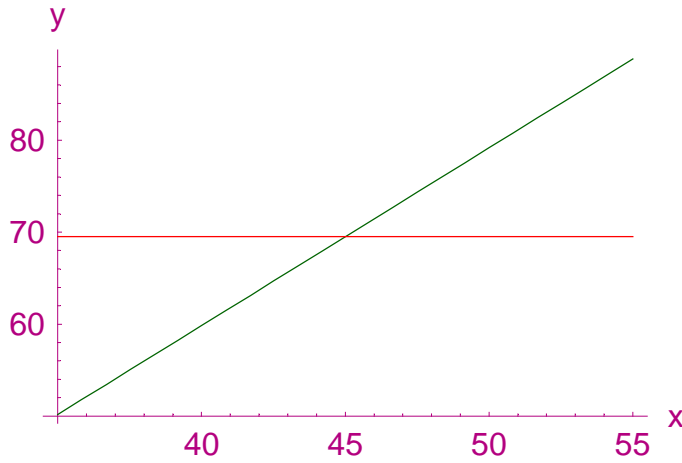
```
Input @ BestJump'[45]
      1.93182
```

```
Input @ NSolve[69.5 == 1.9318169265204892` * 45 + d, d]
      {{d} -> -17.43176169342202<<
```

```
Input @ Clear[Tangent]
```

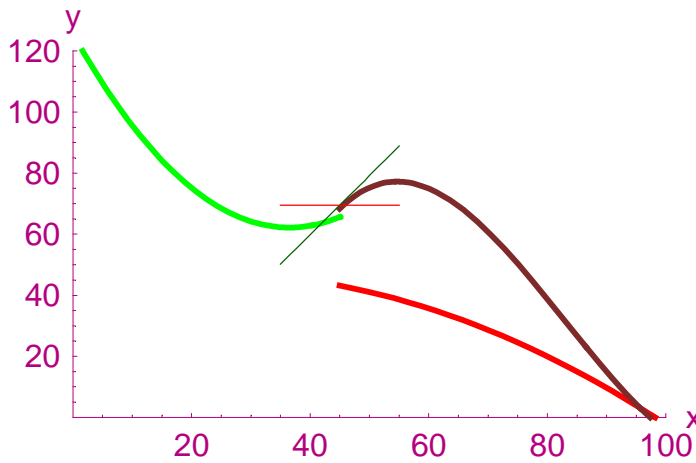
```
Input @ Tangent[x_] =
      1.9318169265204892` x + -17.4317616934220183111392543651
      -17.4317616934220183111392543651 + 1.93182 x
```

```
Input @ Tangentimg = MDPlot[Tangent[x], 69.5, 8x, 35, 55<D
```



Graphics

Input @ Show ramping, flighting, finaljumping, Tangenting



Graphics

m1 = 0;

m2 = 1.9;

Input @ $j = \text{ArcTan} \left(\frac{m2 - m1}{1 + m2 * m1} \right) \text{ in } \text{Degrees}$

62.2415

Input @ Bestjump1(x) = -386.71390307804813` + 21.648669817120318` x - 0.3199063335947178` x² + 0.0014268429914450743` x³

-386.714 + 21.6487 x - 0.319906 x² + 0.00142684 x³

Input @ Bestjump1'(45)

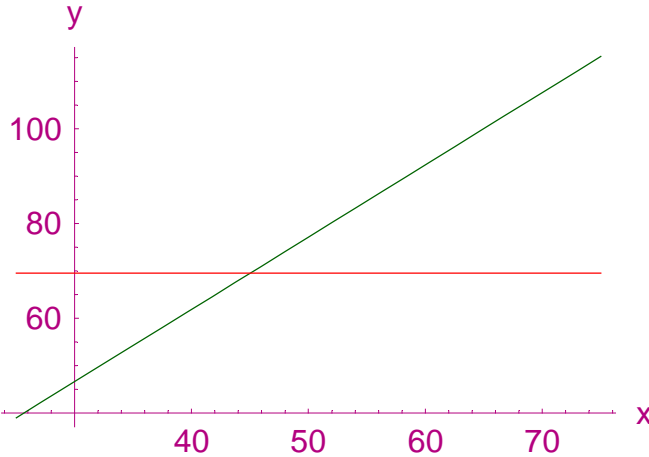
1.52517

Input @

Input @ NSolve[69.5 == 1.5251709666245397` * 45 + d, d]
 {{d -> 0.867306501895712}}

Input @ Tangent2[x_?D] =
 1.5251709666245397` x + 0.867306501895711789984488859772682
 0.867306501895711789984488859772682 + 1.52517 x

Input @ MDPlot[Tangent2[x], {x, 25, 75}]



Graphics

m1 = 0;
 Input @ m2 = 1.5251709666245397`;
 j = ArcTan[Abs[m2 - m1] / (1 + m2 * m1)] * 180 / Pi
 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[45]
 -0.23386

data9 = {{44.0752, 68.593},
 {46.7681, 69.4644}, {49.9097, 69.4644}, {53.0514, 69.4644},
 {57.9882, 65.1073}, {60.681, 63.3645}, {62.9251, 59.0074},
 Input @ {65.1691, 55.5217}, {68.3107, 51.1646}, {70.5548, 47.6789},
 {75.4916, 38.9647}, {79.5309, 31.122}, {82.6725, 25.8935},
 {86.7118, 19.7935}, {88.507, 15.4364}, {90.3022, 9.33648},
 {88.507, 9.33648}, {94.7903, 5.8508}, {97.9319, 3.23654}}

```

844.0752, 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, #1, 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 == 0.644233647878738` * 45 + d, dD
{{d -> 40.50948584545679<<
    
```

```

Input @ perfectangent@x_D =
0.644233647878738` x + 40.5094858454567869898710341658` 15.6482
40.50948584545679 + 0.644234 x
    
```

```

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



Graphics

```

Input @ m1 = 0;
m2 = 0.644233647878738`;
j = ArcTan[Abs[Abs[m2 - m1] / (1 + m2 * m1)]] * 180 / Pi
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
```

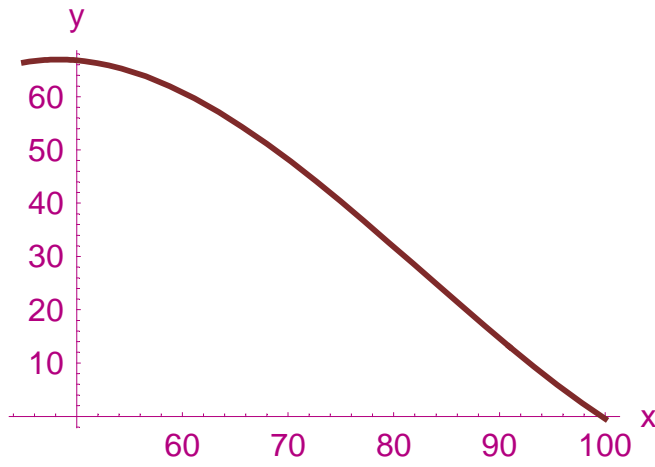
```
Input @ NSolve[69.5 == 0.35200056063936236` * 45 + d, dD
88d 53.6599747712287<<
```

```
Input @ Finalperfecttangent[x_D =
0.35200056063936236` x + 53.6599747712286898604361340404` 15.6495
53.6599747712287 + 0.352001 x
```

Finding the angle.

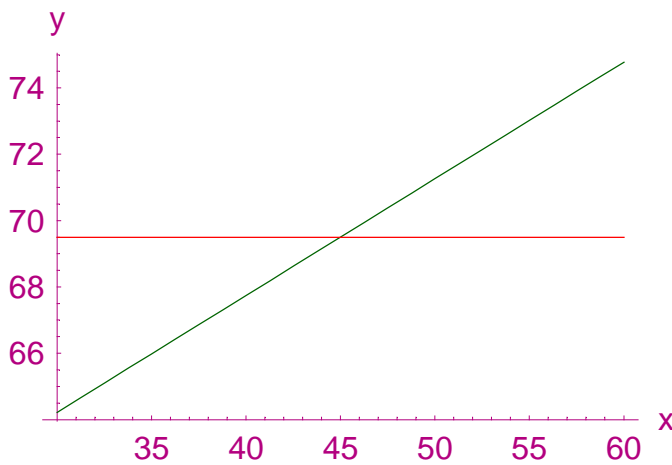
```
m1 = 0;
Input @ m2 = 0.35200056063936236`;
j = ArcTan[Abs[ArcTan[m2, m1] - ArcTan[m1, m2]] / (1 + m2 * m1)] * 180 / Pi
19.3921
```

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



Graphics

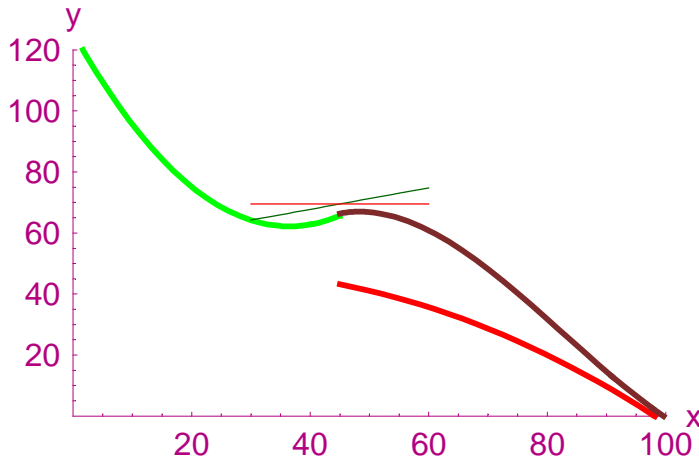
Input @ great = MDPlot@Finalperfecttangent@x, 69.5<, 30, 60<



Graphics

And the result is:

Input @ Show@ramping, flighting, finally, great



⌘ Graphics ⌘

Input @ DONE!

4

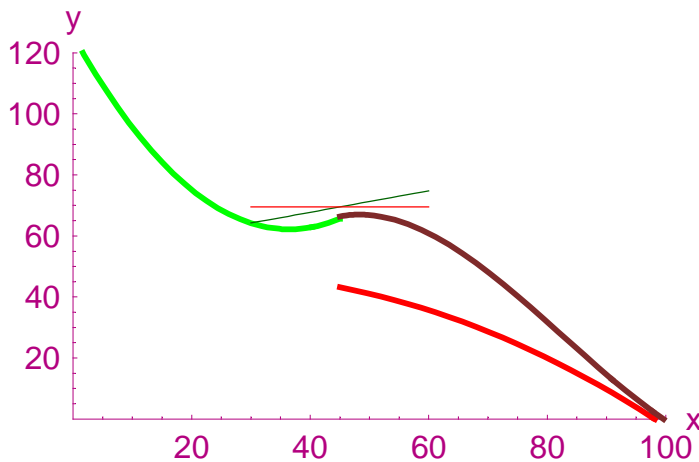
Result and Summary

Open & Close

Print

Section:

Open & 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 careful 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](#) [Close](#)

[Print](#)

Participants

[Open](#) [Close](#)



The members of our group:

- Adrian Trzeciak
- Fabio Juritsch
- Nikolas Pock
- João Joaquim
- Diogo Justino
- Joana Silva

Our Experience with the Project

[Open](#) [Close](#)

1.

The project was a great idea. Most of us are going to use this program for the rest of our lives. 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 portugual students at school and the differences between the countries.

[New Chapter](#)

[Cut Last Chapter](#)

INTERNET 

Open & Close

Print