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.

Measurement of temperature - measurements, displays, mathematical models and prognoses

BHAK Leibnitz, Austria

PC based Math Project for High School Students

1 The project

Open / Close Print

Description of our project:

Open / Close

Every morning when I have breakfast together with my parents I wonder about the same topic: They take their coffee from the coffee machine and though they have no time they have to wait to be able to drink it because of its heat. That made me think about an idea. I could calculate how long it takes the coffee to get to room temperature, so people can drink it. A few days before, our maths teacher Mr. Prof. Narrath mentioned to us a project named 'Comenius' and asked some of us to take part in it. So I, Dietmar Kappel and my partner, Martin Fasching put our hands up and our project started. That was in autumn 2008. From there on we always worked on it. At first we measured the temperature of the coffee every half hour till it reached room temperature, then we put the data into a *MD* Notebook and displayed our measurements with the help of *Mathematica*. Then we wanted to find out if there's a law which describes exactly this cooling down. After finding a law and a cooling constant with the help of Newton's law of temperature we asked ourselves some questions about that topic such as 'what's the temperature after a time of...'. When we had finished we thought about some other drinks which we could measure. In short, we decided also to measure not just heated drinks (coffee and tea) cooling down but we also measured cooled drinks (milk and apple juice) which we took from the fridge. After calculating and displaying cooled drinks, as we did with the heated ones, we wanted to compare the air temperature change within 1 day. We took the data from 'Weather Service Styria (Arge Steiermark)' from two villages, my home town Kitzeck and one other village called Hollenegg which is settled in a totally different region. At first we will show you heated drinks cooling down, then cooled drinks heating up and then we will show you the change of the air temperature.

2 Heated drinks cooling down - coffee:

Open / Close Print

To measure the temperature of coffee and tea we took a cup (125ml) of coffee and tea and measured it with a kitchen machine's temperature gauge (which is shown on a picture) till the temperature reached roomtemperature (21 degrees).

Development of coffee - temperature in the first 3 hours

Open / Close

Here you can see the data written in a table and the data source.

1. The measurepoints shown in a table

Input ▹
```
More… ;
Clear[f, x];     f[x_] := 10 ; (* enter your f *)
start = 0;
stop = 3;
step  = 0.5;

tablevalues = Table[{x, f[x]}, {x, start, stop, step}] // Chop;
MDShowTable[tablevalues,
  {"time in hours", "temperature of coffee in degrees Celsius"}];
```

time in hours	temperature of coffee in degrees Celsius
0	67
0.5	40
1.	32
1.5	27
2.	24
2.5	22
3.	21

2. The data source

Input ▹
```
data1 = {{0, 67}, {0.5, 40},
   {1, 32}, {1.5, 27}, {2, 24}, {2.5, 22}, {3, 21}};
```

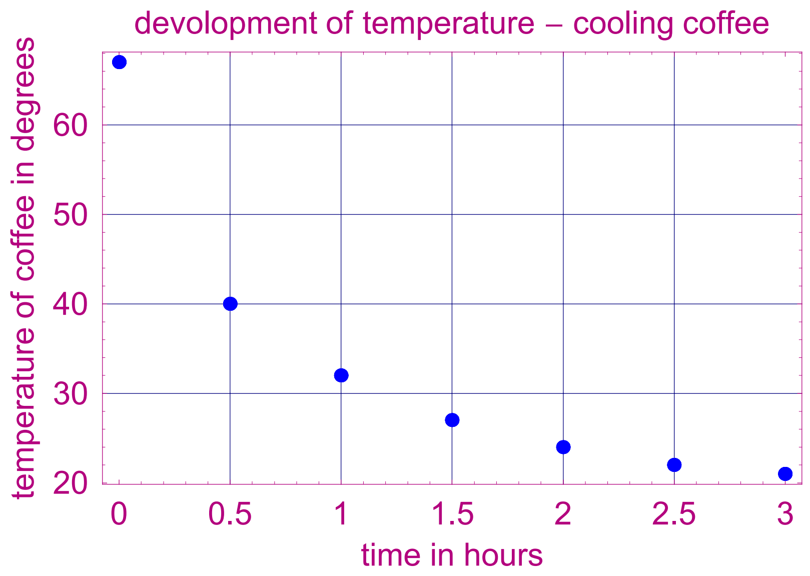
The graphical display

Open / Close

Now we display the data source with the help of *Mathematica* and M@th Desktop

3. The plotting of the measurepoints

```
MDPlotData[data1,
    FrameLabel → {"time in hours",
    "temperature of coffee in degrees"},
  PlotStyle → {Blue, PointSize[.02]},
  PlotLabel -> "devolopment of temperature - cooling coffee"];
```



The attempt to find a mathematical model:

Open / Close

Here you can see the attempt to find a suitable type of function for our topic. We used some function types to find out what's the best one for us and compared it with Newton's law of temperature

Input ▷

```
data1 = {{0, 67}, {0.5, 40},
   {1, 32}, {1.5, 27}, {2, 24}, {2.5, 22}, {3, 21}};
```

4. The linear function

```
(*Fit data*) Clear[x, a, b, c, d, e, f, g, h];
fit[x_] = NonlinearFit[data1,
        a x + b ,       (* model *)
        {x}, {a, b}     (* parameters *)
    ] // Chop[#, 10^-15] &;
```

Input ▷

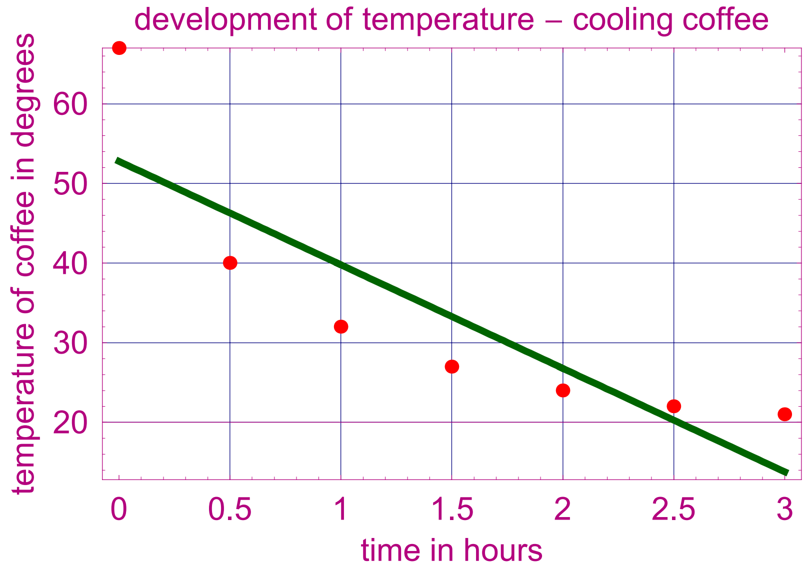
```
{start, stop} = {Min[#], Max[#]} &[First@Transpose@data1];
MDPlotFitData[data1, {fit[x]} , {x, start, stop},
  FrameLabel →
```

```
 {"time in hours", "temperature of coffee in degrees"},
Epilog → {Red, PointSize[0.02], Point /@ data1},
PlotStyle → {{DarkGreen, Thickness[0.01]}},
PlotLabel → "development of temperature - cooling coffee"];
```

Sum of squarred error:

$\sum (y_i - \hat{y}_i)^2$ Sum of Squared Error : 404.429

52.7857 - 13. x



This type of function wouldn't be useful because the temperature becomes negativ after a time of 3 hours and that's not the truth. The sum of sqarred error is a good indicator if the function fits our model. In this case the sum of sqarred error is very high → 404.429. That means that the linear function doesn't fit our mathematical model.

5. Quadratic function

```
(*Fit data*) Clear[x, a, b, c, d, e, f, g, h];
fit[x_] = NonlinearFit[data1,
        a x^2 + b x + c ,    (* model *)
        {x}, {a, b, c }    (* parameters *)
    ] // Chop[#, 10^-15] &;
```

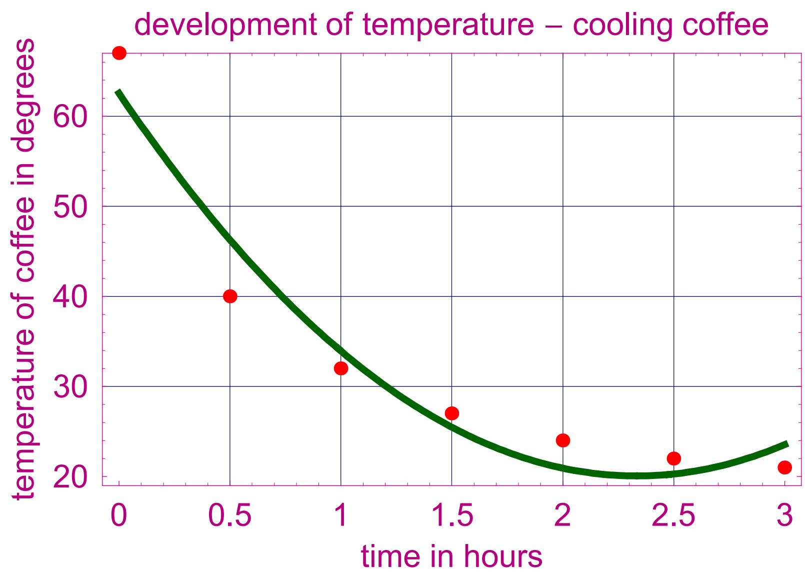
Input ▹

```
{start, stop} = {Min[#], Max[#]} &[First@Transpose@data1];
MDPlotFitData[data1, {fit[x]} , {x, start, stop},
  FrameLabel → {"time in hours",
   "temperature of coffee in degrees"},
```

```
Epilog → {Red, PointSize[0.02], Point /@ data1},
PlotStyle → {{DarkGreen, Thickness[0.01]}},
PlotLabel → "development of temperature - cooling coffee"];
```

$\sum (y_i - \hat{y}_i)^2$ Sum of Squared Error : 84.2381

$62.5476 - 36.4286\,x + 7.80952\,x^2$



The quadratic function does not suit for our model because of its' development after a time of 2^{30} hours . After this time this type of function increases again which would mean a rise of temperature after a certain time. That's why we can not use this type of function. In comparison with the other types of function the quadratic one also has a fairly high sum of sqarred error → 84.2381.

6. Cubic function

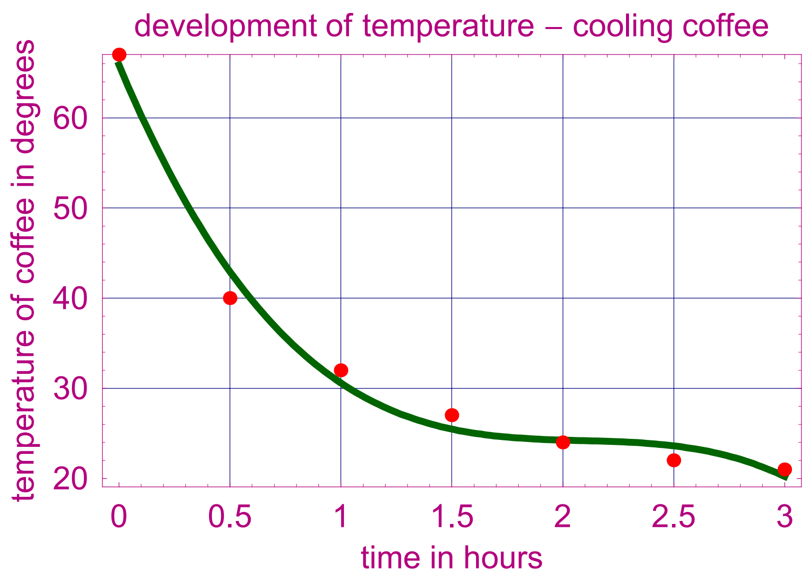
```
(*Fit data*) Clear[x, a, b, c, d, e, f, g, h];
fit[x_] = NonlinearFit[data1,
        a x^3 + b x^2 + c x + d ,    (* model 3 rd order *)
        {x}, {a, b, c, d }           (* parameters *)
    ] // Chop[#, 10^-15] &;
```

Input ▹

```
{start, stop} = {Min[#], Max[#]} &[First@Transpose@data1];
MDPlotFitData[data1, {fit[x]} , {x, start, stop},
  FrameLabel →
   {"time in hours", "temperature of coffee in degrees"},
  Epilog → {Red, PointSize[0.02], Point /@ data1},
  PlotStyle → {{DarkGreen, Thickness[0.01]}},
  PlotLabel → "development of temperature - cooling coffee"];
```

$\sum (y_i - \hat{y}_i)^2$ Sum of Squared Error : 17.5714

$65.881 - 58.6508\,x + 27.8095\,x^2 - 4.44444\,x^3$



The cubic function has already a low sum of sqarred error, but if we look at the development of the curve, we can see, that this type of function would become negativ after a time of 3 hours. That's why we can't use this model. The sum of sqarred error is in comparison with other ones at a low level → 17.5714.

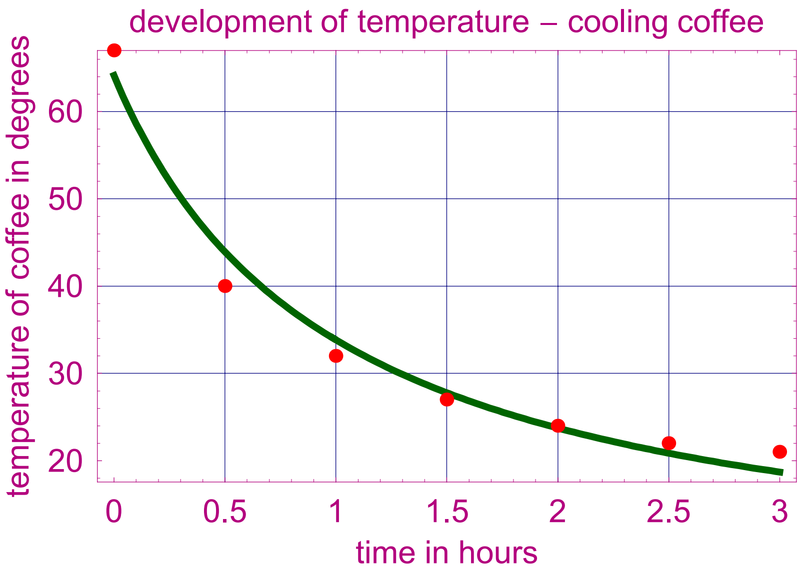
7. Hyperbel

```
fit[x_] = NonlinearFit[data1,
```

```
          a + b/(x + 1),      (* model *)
          {x}, {a, b }    (* parameters *)
     ] // Chop[#, 10^-15] &;

{start, stop} = {Min[#], Max[#]} &[First@Transpose@data1];
MDPlotFitData[data1, {fit[x]} , {x, start, stop},
   FrameLabel → {"time in hours",
     "temperature of coffee in degrees"},
   Epilog → {Red, PointSize[0.02], Point /@ data1},
   PlotStyle → {{DarkGreen, Thickness[0.01]}},
   PlotLabel → "development of temperature - cooling coffee"];
```

$\sum (y_i - \hat{y}_i)^2$ Sum of Squared Error : 34.413

$3.55948 + \frac{60.5649}{1 + x}$



The hyberbel does not fit for our model because of it's high sum of sqarred error → 34.413. Another cause for not using it is the development of the curve. As you can see in the graphic, the curve becomes negative after the third hour, which is not possible.

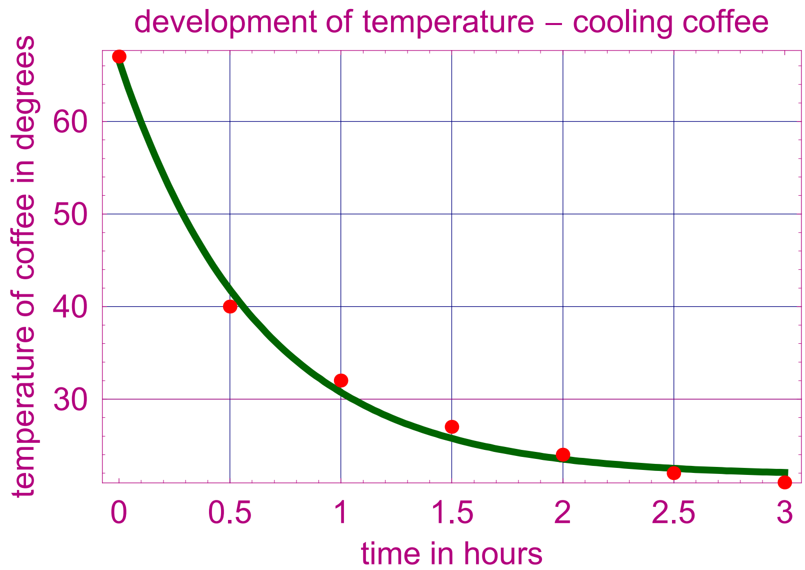
8. Exponential function (the function which Newton used for his law of temperature)

```
(*Fit data*) Clear[x, a, b, c, d, e, f, g, h];
fit[x_] = NonlinearFit[data1,
          a Exp[-b x] + c ,      (* model *)
          {x}, {a, b, c}       (* parameters *)
     ] // Chop[#, 10^-15] &;
```

```
{start, stop} = {Min[#], Max[#]} &[First@Transpose@data1];
MDPlotFitData[data1, {fit[x]} , {x, start, stop},
  FrameLabel →
   {"time in hours", "temperature of coffee in degrees"},
  Epilog → {Red, PointSize[0.02], Point /@ data1},
  PlotStyle → {{DarkGreen, Thickness[0.01]}},
  PlotLabel → "development of temperature - cooling coffee"];
```

$\sum (y_i - \hat{y}_i)^2$ Sum of Squared Error : 8.33648

$21.7089 + 44.8374\, e^{-1.60264\, x}$



The last type of function we used was the exponential one. As you can see above in the graphic, this type is the most suitable one. The sum of sqarred error is very low, which means that this type fits our mathematical model → 8.33648.
Newton found out, that the cooling down has an exponential behavior.

9. Interpretation

The only type of tunction, which is useful for us, is the exponential one. Newton already found out, that the exponential-function describes the decrease and increase of temperature the best way. That's why we use this type of function for our project.

10. The mathematical model which we use

Input ▷
```
tempcoffee[x_] =
  21.70892326666338` + 44.837395610164016` e^(-1.6026434444559938` x);
```

Prognoses, questions, answers:

Open / Close

Here you can find some questions which we answer and some prognoses which we put up

11. What`s the temperature at the time $t = \frac{3}{4}$ hours respectively at the time $t = 2^{48}$ hours?

Input ▷
```
tempcoffee[0.75]
```

```
35.1869
```

At the time of t = 0.75 there is a temperature of 35.18 degrees

Input ▷
```
tempcoffee[2.8]
```

```
22.2133
```

At the time of t = 2,8 there is a temperature of 22,21 degrees

12. At which time there is a temperature of 30 degrees respectively 50 degrees?

Input ▷
```
NSolve[tempcoffee[x] == 30, x]
```

```
{{x → 1.05317}}
```

Input ▷
```
1.0531741644755177` * 60
```

```
63.1904
```

After the time of 1.05 hours - this means 63.19 minutes - there is a temperature of 30 degrees

Input ▷
```
NSolve[tempcoffee[x] == 50, x]
```

```
{{x → 0.287335}}
```

Input ▷
```
0.28733532159926356` * 60
```

```
17.2401
```

After the time of 0.28 hours - this means 17.24 minutes - there is a temperature of 50 degrees.

3 Heated drinks cooling down - tea

Open / Close Print

Development of tea - temperature in the first 3 hours: :

Open / Close

Here you can see the development of the temperature of tea cooling down in the first 3 hours. We did the same as we did with coffee.

1. The measurepoints shown in a table

Input ▹
```
More… ;
Clear[f, x];     f[x_] := 10 ; (* enter your f *)
start = 0;
stop = 3;
step  = 0.5;

tablevalues = Table[{x, f[x]}, {x, start, stop, step}] // Chop;
MDShowTable[tablevalues,
  {"time in hours", "temperature of tea in degrees"}];
```

time in hours	temperature of tea in degrees Celsius
0	75
0.5	45
1.	34
1.5	28
2.	25
2.5	23
3.	21

2. The data source

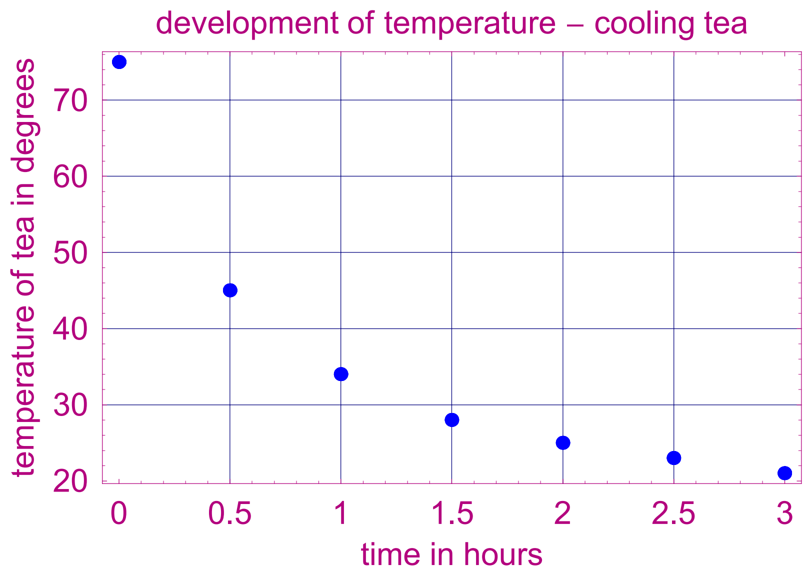
Input ▹
```
data2 = {{0, 75}, {0.5, 45},
   {1, 34}, {1.5, 28}, {2, 25}, {2.5, 23}, {3, 21}};
```

The graphical display:
Open / Close

Now we displayed the data source with the help of *Mathematica and M@th Desktop.*

3. The plotting of the measurepoints

Input ▹
```
MDPlotData[data2,
    FrameLabel →
   {"time in hours", "temperature of tea in degrees"},
  PlotStyle → {Blue, PointSize[.02]},
  PlotLabel -> "development of temperature - cooling tea"];
```



The mathematical model:
Open / Close

As we already know the exponential function is the best one for our experiments.

Input ▹
```
data2 = {{0, 75}, {0.5, 45},
   {1, 34}, {1.5, 28}, {2, 25}, {2.5, 23}, {3, 21}};
```

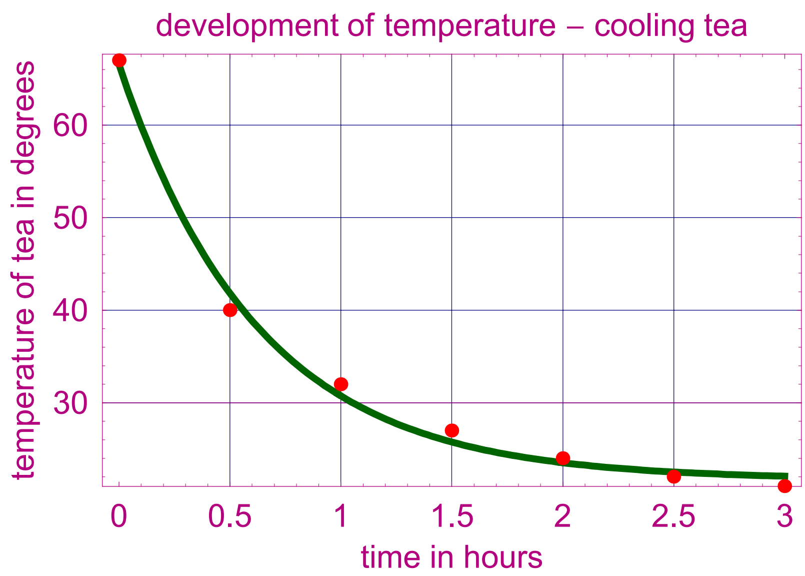
4. The exponential function

```
(*Fit data*) Clear[x, a, b, c, d, e, f, g, h];
fit[x_] = NonlinearFit[data1,
        a Exp[-b x] + c ,     (* model *)
```

```
        {x}, {a, b, c}      (* parameters *)
    ] // Chop[#, 10^-15] &;

{start, stop} = {Min[#], Max[#]} &[First@Transpose@data1];
MDPlotFitData[data1, {fit[x]} , {x, start, stop},
  FrameLabel →
   {"time in hours", "temperature of tea in degrees"},
  Epilog → {Red, PointSize[0.02], Point /@ data1},
  PlotStyle → {{DarkGreen, Thickness[0.01]}},
  PlotLabel -> "development of temperature - cooling tea"];
```

$\sum (y_i - \hat{y}_i)^2$ Sum of Squared Error : 8.33648

$21.7089 + 44.8374\, e^{-1.60264\, x}$



As seen above at coffee, the model we use for tea temperature cooling down is the exponential function. In this case the sum of sqarred error is also quite low so that we can use it for our calculations.

5. The mathematical model

Input ▹
```
temptea[x_] =
  21.818952370400307` + 52.809261000666815` e^(-1.530889442781991` x);
```

Prognoses, questions, answers:
Open / Close

Now we use the expo-function for solving some problems and answering some questions.

6. What`s the temperature at the time t = 27' respectively at the time t =2^{12} hours?

Input ▷
```
temptea[0.45]
```

48.336

At the time of t = 0.45 there is a temperature of 48.336 degrees

Input ▷
```
tempcoffee[2.20]
```

23.0284

At the time of t = 2,20 there is a temperature of 23.02 degrees

7. At which time there is a temperature of 32 degrees respectively 55 degrees?

Input ▷
```
NSolve[temptea[x] == 32, x]
```

{{x → 1.0753}}

Input ▷
```
1.075295583515886` * 60
```

64.5177

After the time of 1.07 hours - this means 64.52 minutes - there is a temperature of 30 degrees

Input ▷
```
NSolve[tempcoffee[x] == 55, x]
```

{{x → 0.185789}}

Input ▷
```
0.18578874832890208` * 60
```

11.1473

After the time of 0.18 hours - this means 11.14 minutes - there is a temperature of 55 degrees.

4 The theoretical model - the cooling constant (based on Newton's law of temperature):

Open / Close Print

Coffee:

Open / Close

We took Newton's law of temperature and applied it on our examples.

1. Cooling constant

time in hours	temperature of coffee in degrees Celsius
0	67
0.5	40
1.	32
1.5	27
2.	24
2.5	22
3.	21

Tc[t] = T1 + (T2 - T1) * e^{k*t}

k ... cooling constant
T1... Room temperature
T2... Temperature of the beverage at the time t = 0
TCconstant.... Temperature of coffee - cooling constant
TCaverage.... Temperature of coffee - average rate of change
TCmoment.... Temperature of coffee - instantanous rate of change

$21.7089 + 44.8374\, e^{k*x}$

the calculation of the cooling constant with the help of one measurepoint and the Newton's law of temperature

Input ▹ data1

```
{{0, 67}, {0.5, 40}, {1, 32}, {1.5, 27}, {2, 24}, {2.5, 22}, {3, 21}}
```

2. The calculation of the cooling constant

```
Input ▹ Clear[T, t, k, glg1]

Input ▹ TCconstant[t_] = 21 + (67 - 21) ⅇ^(k*t)

        21 + 46 ⅇ^(k t)

Input ▹ glg1 = TCconstant[0.5] == 40

        21 + 46 ⅇ^(0.5 k) == 40

Input ▹ NSolve[glg1, k]

        {{k → -1.7684}}

Input ▹ k1 = -1.7684048346453092`

        -1.7684

Input ▹ glg2 = TCconstant[1] == 32

        21 + 46 ⅇ^k == 32

Input ▹ NSolve[glg2, k]

        {{k → -1.43075}}

Input ▹ k2 = -1.4307461236907244`

        -1.43075

Input ▹ glg3 = TCconstant[1.5] == 27

        21 + 46 ⅇ^(1.5 k) == 27

Input ▹ NSolve[glg3, k]

        {{k → -1.35792}}

Input ▹ k3 = -1.3579212848406934`

        -1.35792

Input ▹ glg4 = TCconstant[2] == 24

        21 + 46 ⅇ^(2 k) == 24

Input ▹ NSolve[glg4, k]

        {{k → -1.36501}}

Input ▹ k4 = -1.3650145539104928`

        -1.36501

Input ▹ glg5 = TCconstant[2.5] == 22
```

$21 + 46\, e^{2.5\, k} == 22$

Input ▹ `NSolve[glg5, k]`

`{{k → -1.53146}}`

Input ▹ ``k5 = -1.531456558595638` ``

`-1.53146`

Input ▹ `glg6 = TCconstant[3] == 21`

$21 + 46\, e^{3\, k} == 21$

Input ▹ `k6 = NSolve[glg6, k]`

`{{k → -∞}}`

Input ▹ `k = (k1 + k2 + k3 + k4 + k4) / 5`

`-1.45742`

Input ▹ ``k = -1.4574202701995425` ``

`-1.45742`

Input ▹ `TCconstant[t]`

$21 + 46\, e^{-1.45742\, t}$

3. The Comparison of the cooling constants

We calculated the cooling constant k with the help of our measurepoints and compared it with the cooling konstant k which the computer calculated.

Input ▹

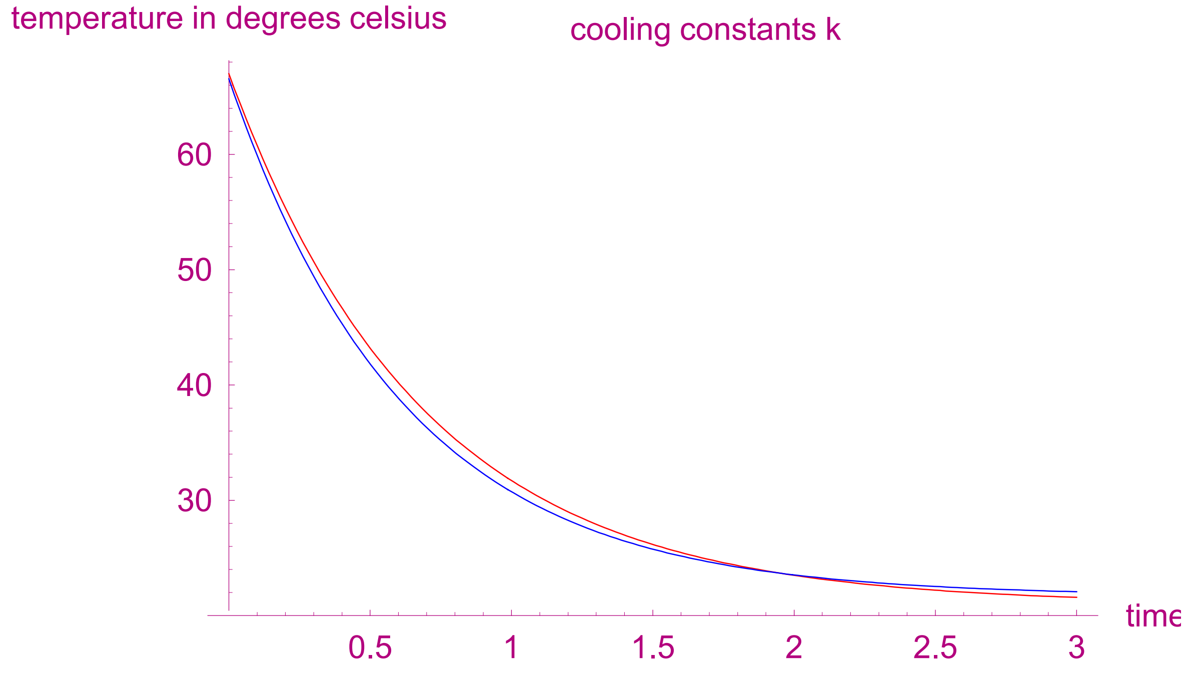
```
TCconstant[t] = 21 + 46 ⅇ^(-1.4574202701995425` t);
```

Input ▹

```
tempcoffee[t_] =
  21.70892326666338` + 44.837395610164016` ⅇ^(-1.6026434444559938` t);
```

Input ▹

```
MDPlot[{TCconstant[t], tempcoffee[t]}, {t, 0, 3}, AxesLabel →
   {"time in hours", "temperature in degrees celsius"},
  PlotLabel -> "              cooling constants k",
  PlotStyle → {Red, Blue}];
```



As you can see in the graphic the cooling constant we calculated compared to the cooling constant we got from the computer is very similar.

4. What's the average change of temperature? - The average rate of change

We took Newton's law of temperature and applied it on our example to calculate the average rate of change

Input ▷ `Clear[T, t, k]`

Input ▷ `k = -1.4574202701995425``

`-1.45742`

Input ▷ `TCaverage[t_] = 21 + (67 - 21) * ⅇ^(-1.4574202701995425`*t)`

$21 + 46\, e^{-1.45742\, t}$

Input ▷ `a = {0.5, TCaverage[0.5]}`

`{0.5, 43.1964}`

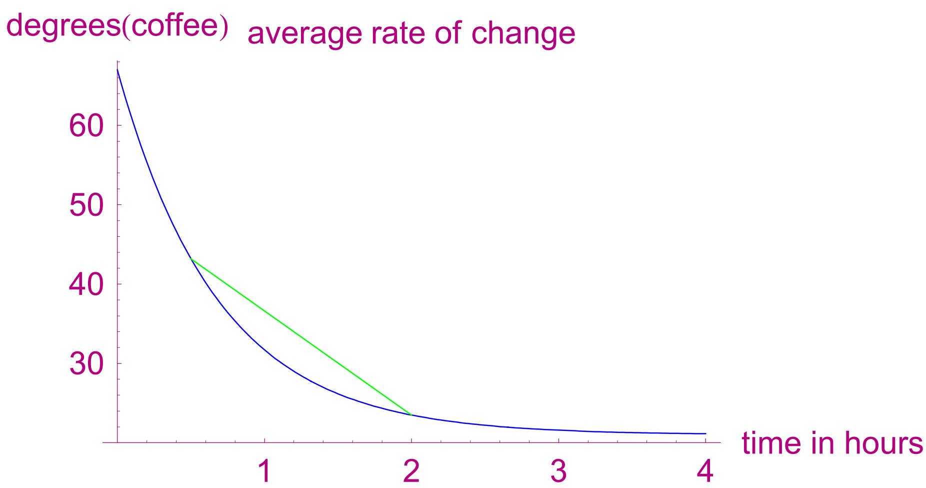
Input ▷ `b = {2, TCaverage[2]}`

`{2, 23.4938}`

Input ▹
```
secant = {Green, Line[{a, b}]}
```

```
{RGBColor[0., 1., 0.], Line[{{0.5, 43.1964}, {2, 23.4938}}]}
```

Input ▹
```
MDPlot[{ TCaverage[t]}, {t , 0 , 4},
 PlotStyle → {Blue , Dashing[{.03}]}, Epilog → secant,
 PlotLabel → " average rate of change ",
 AxesLabel → {"time in hours", "degrees(coffee)"}];
```



Input ▹ $m = \frac{TCaverage[3] - TCaverage[1]}{3 - 1}$

```
-5.06491
```

that means that the temperature of coffee decreases by 5.06 °Celcius per hour between time = 3 hours and time = 1 hour

5. What's the change of temperature exactly at the time of t = 18 min.?

Input ▹
```
Clear[TCmoment, tg, d, t, T, k]
```

Input ▹
```
k = -1.4574202701995425`;
TCmoment[t_] = 21 + (67 - 21) ⅇ^(k*t)
```

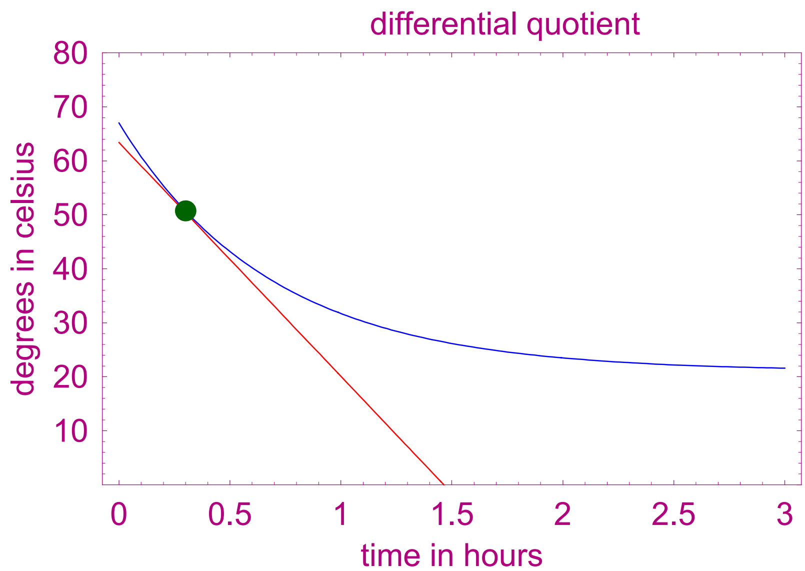
$21 + 46\, e^{-1.45742\, t}$

Input ▹
```
TCmoment'[t]
```

$-67.0413\, e^{-1.45742\, t}$

Input ▹
```
a = TCmoment'[0.3]
```

```
-43.297

Input ▹ tg[t_] = a * t + d

d - 43.297 t

Input ▹ glg1 = tg[0.3] == TCmoment[0.3]

-12.9891 + d == 50.708

Input ▹ NSolve[glg1, d]

{{d → 63.6971}}

Input ▹ d = 63.37784494712284`

63.3778

Input ▹ tg[t]

63.3778 - 43.297 t

        MDPlot[{ TCmoment[t], tg[t]}, {t , 0 , 3}, PlotStyle → {Blue, Red},
         PlotLabel → "                differential quotient ",
         PlotRange → {0, 80},
Input ▹  AxesLabel → {"time in hours", "temperature in degrees"},
         Frame → True,
         FrameLabel → {"time in hours", "degrees in celsius"}, Epilog →
          {DarkGreen, PointSize[0.03], Point[{0.3, TCmoment[0.3]}]}];
```



That means the change of temperature the differntial quotient at the time of 1 is 21.0191 degrees

Tea:

Open / Close

5 cooled drinks heating up - milk:

Open / Close Print

To measure the temperature of milk and applejuice we put the drink into the fridge and after it reached the temperature of 7 (milk) respectively 8 (applejuice) degrees. Then we took it out and measured every half hour till the temperature reaches room temperature.

The development of milk - temperature of milk in the first 3.5 hours

Open / Close

Here you can see the measurepoints shown in a table and the data source

1. Measurepoints shown in a table

Input ▹
```
More... ;
Clear[f, x];     f[x_] := 10 ; (* enter your f *)
start = 0;
stop = 3.5;
step  = 0.5;

tablevalues = Table[{x, f[x]}, {x, start, stop, step}] // Chop;
MDShowTable[tablevalues,
  {"time in hours", "temperature of milk in degrees"}];
```

time in hours	temperature of milk in degrees
0	7
0.5	11
1.	13
1.5	15
2.	17
2.5	18
3.	19
3.5	20

2. The data source

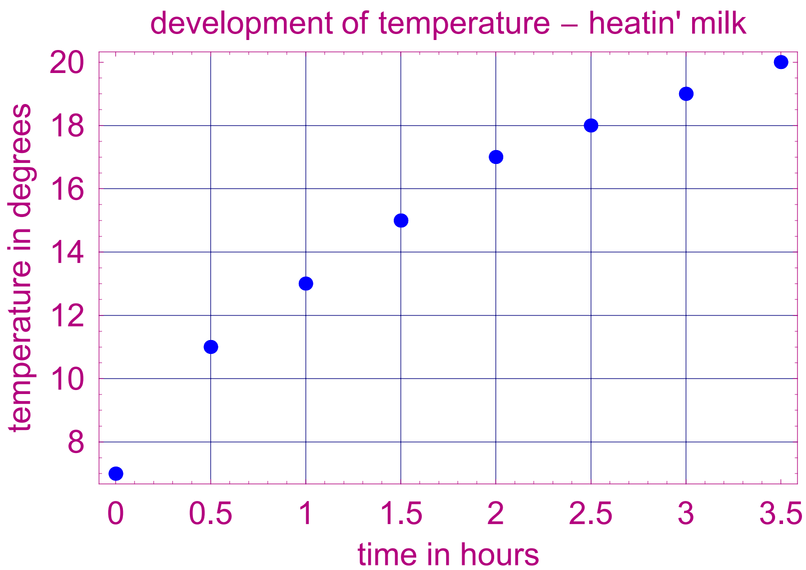
Input ▹
```
data3 = {{0, 7}, {0.5, 11}, {1, 13},
   {1.5, 15}, {2, 17}, {2.5, 18}, {3, 19}, {3.5, 20}};
```

The graphical display:
Open / Close

Now we use *Mathematica* to plot the data source

3. The measurepoints plotted

Input ▹
```
MDPlotData[data3,
    FrameLabel →
   {"time in hours", "temperature in degrees"},
  PlotStyle → {Blue, PointSize[.02]},
  PlotLabel -> "development of temperature - heatin' milk"];
```



The mathematical model
Open / Close

As we already know: the change of temperature is described with the expo function, also for increases of temperature

Input ▹
```
data3 = {{0, 7}, {0.5, 11}, {1, 13},
   {1.5, 15}, {2, 17}, {2.5, 18}, {3, 19}, {3.5, 20}};
```

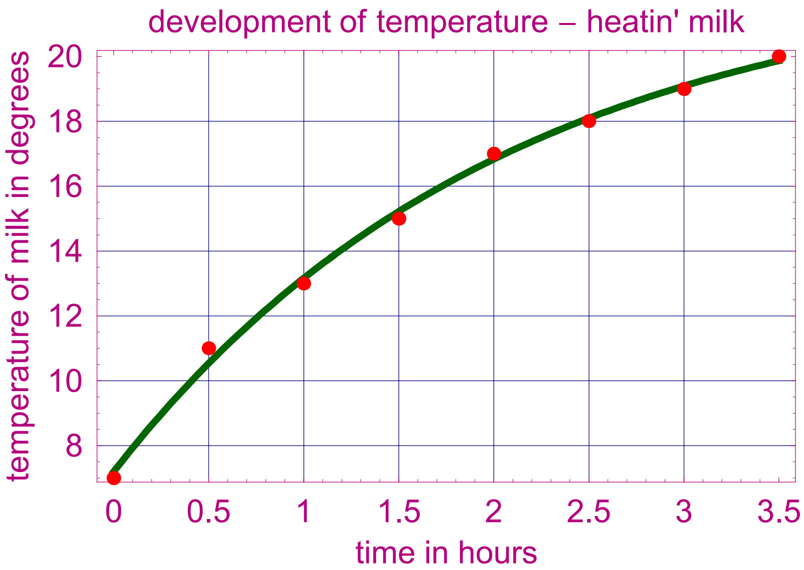
4. Exponential function

```
(*Fit data*) Clear[x, a, b, c, d, e, f, g, h];
fit[x_] = NonlinearFit[data3,
        a Exp[-b x] + c ,      (* model *)
        {x}, {a, b, c}       (* parameters *)
    ] // Chop[#, 10^-15] &;
```

Input ▷
```
{start, stop} = {Min[#], Max[#]} &[First@Transpose@data3];
MDPlotFitData[data3, {fit[x]} , {x, start, stop},
  FrameLabel →
   {"time in hours", "temperature of milk in degrees"},
  Epilog → {Red, PointSize[0.02], Point /@ data3},
  PlotStyle → {{DarkGreen, Thickness[0.01]}},
  PlotLabel -> "development of temperature - heatin' milk"];
```

$\sum (y_i - \hat{y}_i)^2$ Sum of Squared Error : 0.387674

$22.6923 - 15.4983\, e^{-0.486291\, x}$



5. The mathematical model

Input ▷
```
Clear[tempmilk]
```

Input ▷
```
tempmilk[x_] =
  22.69230434184433` - 15.49832075516016` e^(-0.486291471608051` x);
```

Prognoses, questions, answers:

Open / Close

6. What's the temperature at a time of t = 0.25 respectively at the time t = 2.75?

Input ▷
```
tempmilk[0.25]
```
```
8.96813
```

After a time of t = 0.25 there is a temperature of 8.96 degrees.

Input ▷
```
tempmilk[2.75]
```
```
18.6232
```

After a time of t = 2.75 there is a temperature of 18.62 degrees.

7. At which time there is a temperature of 8.5 respectively 18.5 degrees?

Input ▷
```
NSolve[tempmilk[x] == 8.5, x]
```
```
{{x → 0.181027}}
```

Input ▷
```
0.1810268426984801` * 60
```
```
10.8616
```

That means that the temperature beeing searched (8.5 degrees) is reached at a time of 10.86 minutes

Input ▷
```
NSolve[tempmilk[x] == 18.5, x]
```
```
{{x → 2.68868}}
```

Input ▷
```
2.6886779041006683` * 60
```
```
161.321
```

That means that the temperature beeing searched (8.5 degrees) is reached at a time of 161.32 minutes.

6 cooled drinks heating up - applejuice:

Open / Close Print

Development of applejuice - temperature in the first 3.5 hours:

Open / Close

Here you can see the measurepoints shown in a table and the data source

1. The measurement shown in a table

Input ▹

```
More… ;
Clear[f, x];     f[x_] := 10 ; (* enter your f *)
start = 0;
stop = 3;
step  = 0.5;

tablevalues = Table[{x, f[x]}, {x, start, stop, step}] // Chop;
MDShowTable[tablevalues, {"time in hours",
   "temperature of applejuice in degrees"}, Frame → False];
```

time in hours	temperature of applejuice in degrees
0	8
0.5	12
1.	14
1.5	16
2.	18
2.5	19
3.	20

2. The data source

Input ▹

```
data4 = {{0, 8}, {0.5, 12},
   {1, 14}, {1.5, 16}, {2, 18}, {2.5, 19}, {3, 20}};
```

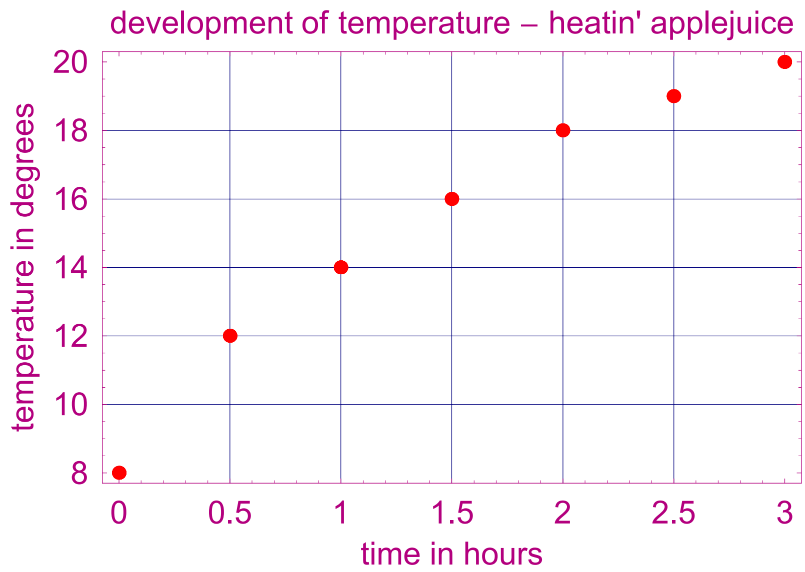
The graphical display:

Open / Close

Now we use the 'fit data - function' to plot our points

3. The measurepoints plotted

Input ▹
```
MDPlotData[data4,
    FrameLabel →
   {"time in hours", "temperature in degrees"},
  PlotStyle → {Red, PointSize[.02]}, PlotLabel ->
   "development of temperature - heatin' applejuice"];
```



The mathematical model:

Open / Close

Input ▹
```
data4
```

```
{{0, 8}, {0.5, 12}, {1, 14}, {1.5, 16}, {2, 18}, {2.5, 19}, {3, 20}}
```

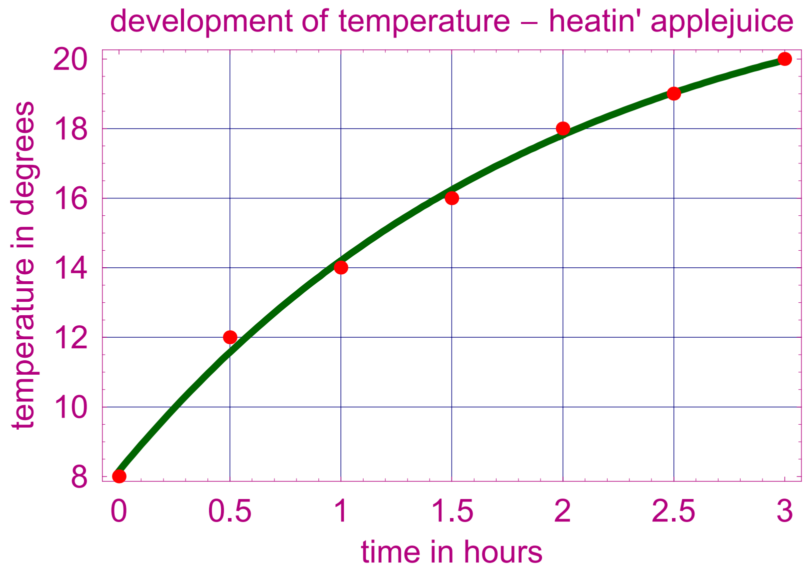
4. Exponential function

```
(*Fit data*) Clear[x, a, b, c, d, e, f, g, h];
fit[x_] = NonlinearFit[data4,
        a Exp[-b x] + c ,      (* model *)
        {x}, {a, b, c}       (* parameters *)
```

```
      ] // Chop[#, 10^-15] &;

{start, stop} = {Min[#], Max[#]} &[First@Transpose@data4];
MDPlotFitData[data4, {fit[x]} , {x, start, stop},
   FrameLabel → {"time in hours", "temperature in degrees"},
   Epilog → {Red, PointSize[0.02], Point /@ data4},
   PlotStyle → {{DarkGreen, Thickness[0.01]}}, PlotLabel ->
    "development of temperature - heatin' applejuice"];
```

$\sum (y_i - \hat{y}_i)^2$ Sum of Squared Error : 0.346876

$23.1555 - 14.998\, e^{-0.516492\, x}$



5. The mathematical model

Input ▹

```
tempapple[x_] =
  23.155529634834107` - 14.997978301780428` e^(-0.5164916767667128` x);
```

Prognoses, questions, answers:

Open / Close

6 the theoretical model - the heat co-efficient (based on Newton's law of temperature):

Open / Close Print

Milk:

Open / Close

We took the law of temperature from Newton and applied it on our examples.

1. Heat co-efficient

time in hours	temperature of milk in degrees
0	7
0.5	11
1.	13
1.5	15
2.	17
2.5	18
3.	19
3.5	20

$T[t] = T1 + (T2 - T1) * e^{k*t}$

k ... heat co-efficient
T1... Room temperature
T2... Temperature of the beverage at the time t = 0 TMco...
Temperature of milk - heat co-efficient TMaverage..
Temperature of milk - average rate of change TMmoment....
Temperature of milk - differential quotient

the calculation of the heat co-efficient with the help of two measurepoints and the Newton's law of temperature

Input ▹
```
data3 = {{0, 7}, {0.5, 11}, {1, 13},
   {1.5, 15}, {2, 17}, {2.5, 18}, {3, 19}, {3.5, 20}};
```

Input ▹
```
Clear[T, TMco, t, k, glg1]
```

Input ▹
```
TMco[t_] = 20 + (7 - 20) * e^(k*t)
```

$20 - 13\, e^{k\, t}$

Input ▷ `glg13 = TMco[0.5] == 11`

$20 - 13\, e^{0.5\, k} == 11$

Input ▷ `NSolve[glg13, k]`

`{{k → -0.73545}}`

Input ▷ `k13 = -0.7354495602506347``

`-0.73545`

Input ▷ `glg14 = TMco[1] == 13`

$20 - 13\, e^{k} == 13$

Input ▷ `NSolve[glg14, k]`

`{{k → -0.619039}}`

Input ▷ `k14 = -0.6190392084062234``

`-0.619039`

Input ▷ `glg15 = TMco[1.5] == 15`

$20 - 13\, e^{1.5\, k} == 15$

Input ▷ `NSolve[glg15, k]`

`{{k → -0.637008}}`

Input ▷ `k15 = -0.6370076300182909``

`-0.637008`

Input ▷ `glg16 = TMco[2] == 17`

$20 - 13\, e^{2\, k} == 17$

Input ▷ `NSolve[glg16, k]`

`{{k → -0.733169}}`

Input ▷ `k16 = -0.7331685343967135``

`-0.733169`

Input ▷ `glg17 = TMco[2.5] == 18`

$20 - 13\, e^{2.5\, k} == 18$

Input ▷ `NSolve[glg17, k]`

`{{k → -0.748721}}`

Input ▹ `k17 = -0.7487208707606365``

-0.748721

Input ▹ `glg18 = TMco[3] == 19`

$20 - 13\, e^{3k} == 19$

Input ▹ `NSolve[glg18, k]`

{{k → -0.854983}}

Input ▹ `k18 = -0.8549831191538456``

-0.854983

Input ▹ `glg19 = TMco[3.5] == 20`

$20 - 13\, e^{3.5k} == 20$

Input ▹ `NSolve[glg19, k]`

{{k → -∞}}

Input ▹ `k = (k13 + k14 + k15 + k16 + k17 + k18) / 6`

-0.721395

Input ▹ `TMco[t]`

$20 - 13\, e^{-0.721395\, t}$

The heat co-efficient for cooled milk heating up is 0.7214

2. Comparison of k

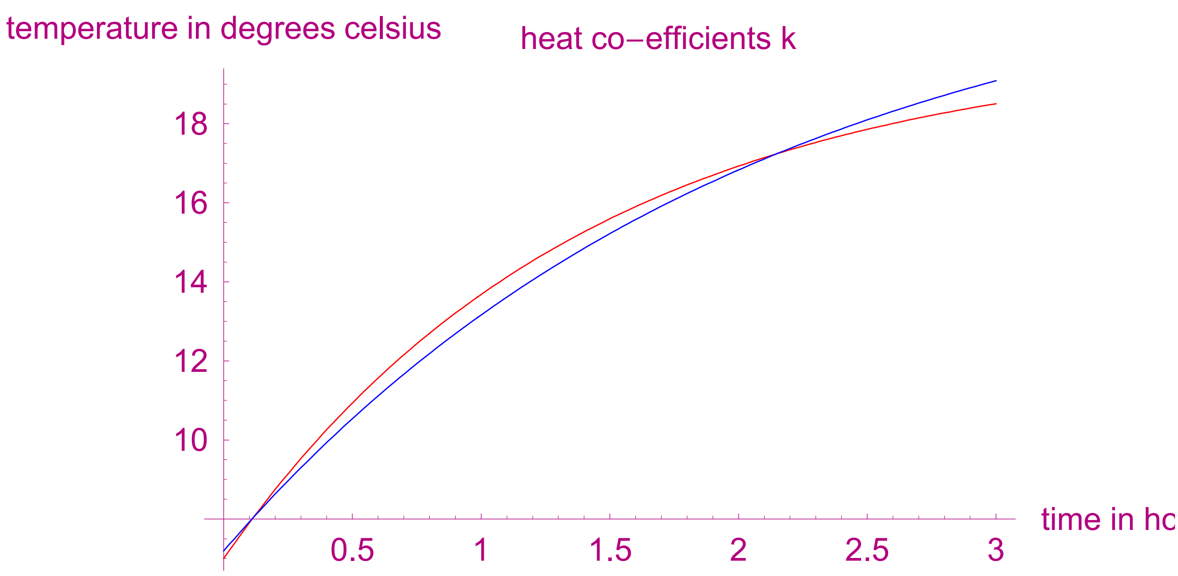
Input ▹
```
tempmilk[t_] =
 22.69230434184433` - 15.49832075516016` ⅇ^(-0.486291471608051` t)
```

$22.6923 - 15.4983\, e^{-0.486291\, t}$

Input ▹ `TMco[t_] = 20 - 13 ⅇ^(-0.7213948204977241` t)`

$20 - 13\, e^{-0.721395\, t}$

Input ▹
```
MDPlot[{TMco[t], tempmilk[t]}, {t, 0, 3}, AxesLabel →
   {"time in hours", "temperature in degrees celsius"},
  PlotLabel -> "            heat co-efficients k",
  PlotStyle → {Red, Blue}];
```



3. Average rate of change

Input ▷
```
Clear[T, t, k]
```

Input ▷
```
TMaverage[t_] = 20 + (7 - 20) * ⅇ^(-0.7213948204977241`*t)
```

$20 - 13\, e^{-0.721395\, t}$

Input ▷
```
a = {0.5, TMaverage[0.5]}
```

{0.5, 10.9365}

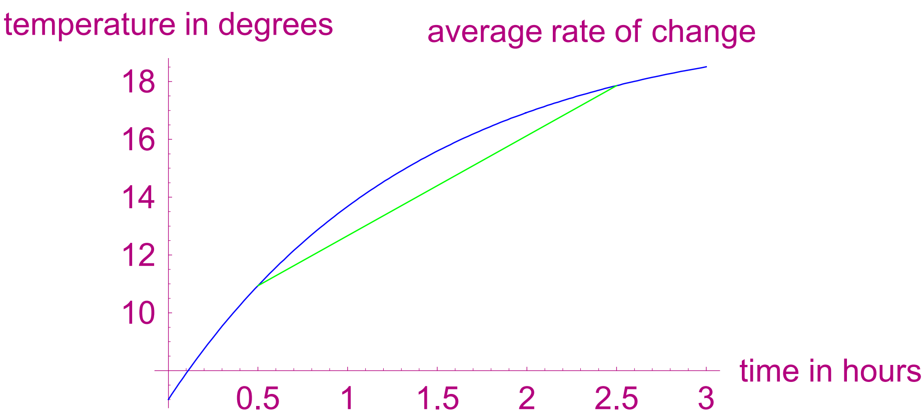
Input ▷
```
b = {2.5, TMaverage[2.5]}
```

{2.5, 17.8586}

Input ▷
```
sekante = {Green, Line[{a, b}]}
```

{RGBColor[0., 1., 0.], Line[{{0.5, 10.9365}, {2.5, 17.8586}}]}

Input ▷
```
MDPlot[{ TMaverage[t]}, {t , 0 , 3},
 PlotStyle → {Blue , Dashing[{.03}]}, Epilog → sekante,
 PlotLabel →
  "                                          average rate of change ",
 AxesLabel → {"time in hours", "temperature in degrees"}];
```



Input ▷ $m = \frac{\text{TMaverage[3] - TMaverage[1]}}{3-1}$

```
2.413
```

that means, that the average decrease of milk-temperature between t=1 and t=3 is 2.39977 degrees.

4. Differential quotient

Input ▷
```
Clear[tg, d, t, T, k, glg1]
```

Input ▷
```
k = -0.7213948204977241`;
TMmoment[t_] = 21 + (67 - 21) * e^(k*t)
```

$21 + 46\, e^{-0.721395\, t}$

Input ▷
```
a = TMmoment'[1]
```

```
-16.13
```

Input ▷
```
tg[t_] = a * t + d
```

```
d - 16.13 t
```

Input ▷
```
glg1 = tg[1] == TMmoment[1]
```

```
-16.13 + d == 43.3594
```

Input ▷
```
NSolve[glg1, d]
```

```
{{d → 59.4893}}
```

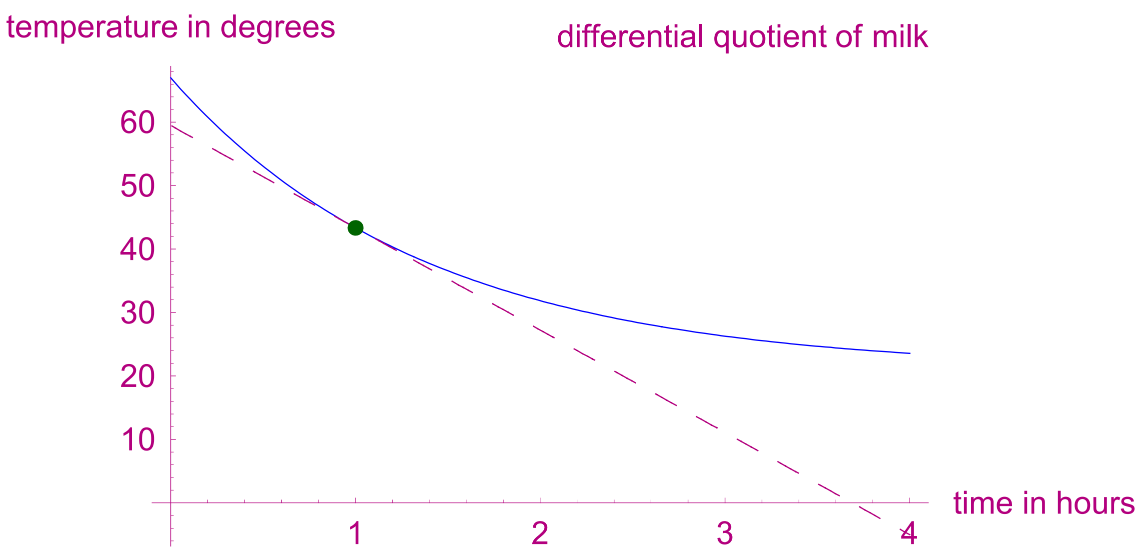
Input ▷
```
d = 59.48934617727001`
```

59.4893

Input ▹
```
tg[t]
```

59.4893 - 16.13 t

Input ▹
```
MDPlot[{ TMmoment[t], tg[t]},
 {t , 0 , 4}, PlotStyle → {Blue , Dashing[{.03}]},
 PlotLabel →
  "                                                     differential
    quotient of milk",
 AxesLabel → {"time in hours", "temperature in degrees"},
 Epilog → {DarkGreen, PointSize[0.02], Point[{1, TMmoment[1]}]}];
```



Applejuice:
Open / Close

5. Heat co-efficient

time in hours	temperature of applejuice in degrees
0	8
0.5	12
1.	14
1.5	16
2.	18
2.5	19
3.	20

T[t] = T1 + (T2 - T1) * e^{k*t}

k ... heat co-efficient
T1... Room temperature
T2... Temperature of the beverage at the time t = 0
Temperature of applejuice - the heat co-efficient TAco...
Temperature of applejuice - average rate of change TAaverage...

the calculation of the cooling constant with the help of two measurepoints and the Newton's law of temperature

```
Input ▷ Clear[T, t, k, glg1]

Input ▷ data4 = {{0, 8}, {0.5, 12},
           {1, 14}, {1.5, 16}, {2, 18}, {2.5, 19}, {3, 20}};

Input ▷ TAco[t_] = 20 + (8 - 20) * e^(k*t)

        20 - 12 e^(k t)

Input ▷ glg20 = TAco[0.5] == 12

        20 - 12 e^(0.5 k) == 12

Input ▷ NSolve[glg20, k]

        {{k → -0.81093}}

Input ▷ k20 = -0.8109302162163288`

        -0.81093

Input ▷ glg21 = TAco[1] == 14

        20 - 12 e^k == 14

Input ▷ NSolve[glg21, k]

        {{k → -0.693147}}

Input ▷ k21 = -0.6931471805599453`
```

```
-0.693147
```

Input ▷ `glg22 = TAco[1.5] == 16`

$20 - 12\, e^{1.5\, k} == 16$

Input ▷ `NSolve[glg22, k]`

```
{{k → -0.732408}}
```

Input ▷ `k22 = -0.7324081924454064``

```
-0.732408
```

Input ▷ `glg23 = TAco[2] == 18`

$20 - 12\, e^{2\, k} == 18$

Input ▷ `NSolve[glg23, k]`

```
{{k → -0.89588}}
```

Input ▷ `k23 = -0.8958797346140275``

```
-0.89588
```

Input ▷ `glg24 = TAco[2.5] == 19`

$20 - 12\, e^{2.5\, k} == 19$

Input ▷ `NSolve[glg24, k]`

```
{{k → -0.993963}}
```

Input ▷ `k24 = -0.9939626599152002``

```
-0.993963
```

Input ▷ `k = (k20 + k21 + k22 + k23 + k24) / 5`

```
-0.825266
```

Input ▷ `k = -0.8252655967501816``

```
-0.825266
```

Input ▷ `TAco[t]`

$20 - 12\, e^{-0.825266\, t}$

The heat co-efficient of cooled applejuice heating up is 0.8253

6. Average rate of change

Input ▹ `Clear[T, t, k]`

Input ▹ `TAaverage[t_] = 20 + (8 - 20) * ⅇ^(-0.8109`*t)`

$20 - 12\, e^{-0.8109\, t}$

Input ▹ `a = {0.1, TAaverage[0.1]}`

`{0.1, 8.93467}`

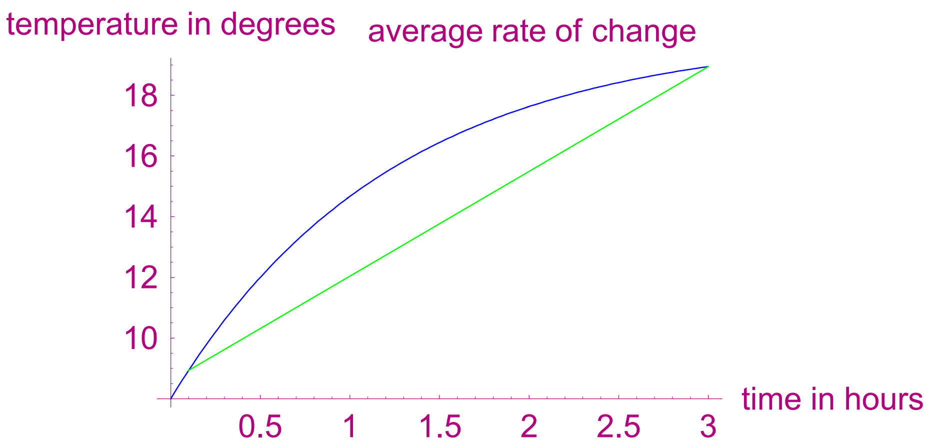
Input ▹ `b = {3, TAaverage[3]}`

`{3, 18.9464}`

Input ▹ `sekante = {Green, Line[{a, b}]}`

`{RGBColor[0., 1., 0.], Line[{{0.1, 8.93467}, {3, 18.9464}}]}`

Input ▹
```
MDPlot[{ TAaverage[t]}, {t , 0 , 3},
 PlotStyle → {Blue , Dashing[{.03}]}, Epilog → sekante,
 AxesLabel → {"time in hours", "temperature in degrees"},
 PlotLabel → "                    average rate of change "];
```



Input ▹ $m = \frac{\text{TAaverage[3] - TAaverage[1]}}{3 - 1}$

`2.13995`

that means, that the increase of the temperature of applejuice between t=1 and t=3 is 2.13995 degrees per day

7

Temperature of Kitzeck on the 13th June:

Open / Close Print

Development of the air temperature in Kitzeck within 1 day:

Open / Close

1. The measurepoints shown in a table.

```
More… ;
Clear[f, x];     f[x_] := 10 ; (* enter your f *)
start = 0.10;
stop = 23.10;
step  = 1;

tablevalues = Table[{x, f[x]}, {x, start, stop, step}] // Chop;
MDShowTable[tablevalues,
  {"time in hours", "temperature in degrees"}];
```

Input ▹

time in hours	temperature in degrees
0.1	15.3
1.1	14.5
2.1	12.9
3.1	12.9
4.1	12.9
5.1	12.7
6.1	12.5
7.1	12.5
8.1	13.7
9.1	14.5
10.1	17.6
11.1	22.7
12.1	22.4
13.1	26.7
14.1	24.7
15.1	23.4
16.1	21.6
17.1	20.8
18.1	20.0

19.1	18.4
20.1	17.3
21.1	16.9
22.1	16.7
23.1	16.5

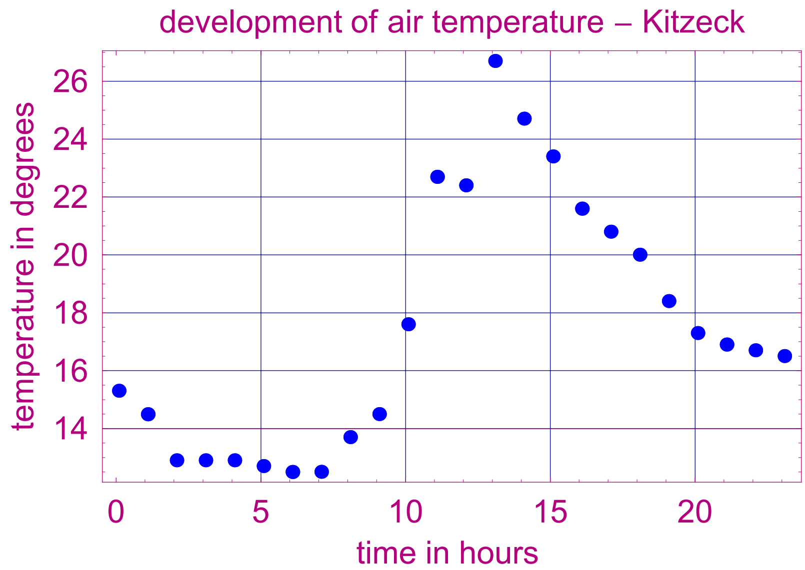
2. The data source

Input ▹
```
data5 = {{0.10, 15.3}, {1.10, 14.5}, {2.10, 12.9}, {3.10, 12.9},
   {4.10, 12.9}, {5.10, 12.7}, {6.10, 12.5}, {7.10, 12.5},
   {8.10, 13.7}, {9.10, 14.5}, {10.10, 17.6}, {11.10, 22.7},
   {12.10, 22.4}, {13.10, 26.7}, {14.10, 24.7}, {15.10, 23.4},
   {16.10, 21.6}, {17.10, 20.8}, {18.10, 20.0}, {19.10, 18.4},
   {20.10, 17.3}, {21.10, 16.9}, {22.10, 16.7}, {23.10, 16.5}};
```

The graphical display:
Open / Close

3. The measurepoints plotted

Input ▹
```
MDPlotData[data5,
    FrameLabel →
   {"time in hours ", "temperature in degrees"},
  PlotStyle → {Blue, PointSize[.02]},
  PlotLabel -> "development of air temperature - Kitzeck"];
```



4. Data fitted

```
data5 = {{0.10, 15.3}, {1.10, 14.5}, {2.10, 12.9}, {3.10, 12.9},
   {4.10, 12.9}, {5.10, 12.7}, {6.10, 12.5}, {7.10, 12.5},
   {8.10, 13.7}, {9.10, 14.5}, {10.10, 17.6}, {11.10, 22.7},
   {12.10, 22.4}, {13.10, 26.7}, {14.10, 24.7}, {15.10, 23.4},
   {16.10, 21.6}, {17.10, 20.8}, {18.10, 20.0}, {19.10, 18.4},
   {20.10, 17.3}, {21.10, 16.9}, {22.10, 16.7}, {23.10, 16.5}};
```

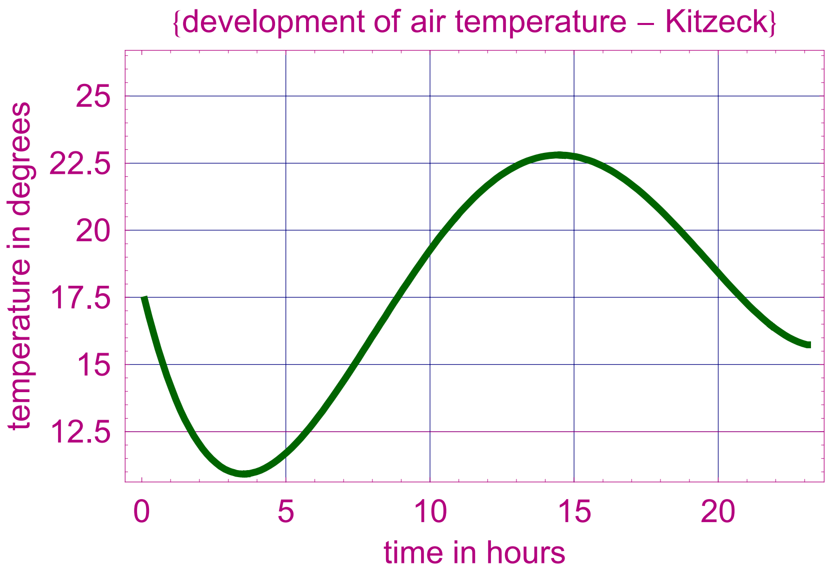
```
(* Fit data with model a0 + a1 x + a2 x^2+ … +an x^n *)
Clear[n, x];

n = 4; (* ≥ 1, degree of the polynomial fit *)
fit[x_] = PolynomialFit[data5, n][x];
```

Input ▹

```
{start, stop} = {Min[#], Max[#]} &[First@Transpose@data5];
MDPlotFitData[data5,
  {Evaluate@Expand@{fit[x]}} , {x, start, stop},
 PlotStyle → {{DarkGreen, Thickness[0.01]}},
  FrameLabel → {"time in hours", "temperature in degrees"},
  PlotLabel → {"development of air temperature - Kitzeck"}];
```

$\sum (y_i - \hat{y}_i)^2$ Sum of Squared Error : 70.226

$17.8556 - 4.4926\,x + 0.887913\,x^2 - 0.0518235\,x^3 + 0.000936476\,x^4$



5. Extrema und Wendepunkte

Input ▹

```
Clear[data5]
```

```
data5 = {{0.10, 15.3}, {1.10, 14.5}, {2.10, 12.9}, {3.10, 12.9},
   {4.10, 12.9}, {5.10, 12.7}, {6.10, 12.5}, {7.10, 12.5},
   {8.10, 13.7}, {9.10, 14.5}, {10.10, 17.6}, {11.10, 22.7},
```

Input ▹

```
{12.10, 22.4}, {13.10, 26.7}, {14.10, 24.7}, {15.10, 23.4},
{16.10, 21.6}, {17.10, 20.8}, {18.10, 20.0}, {19.10, 18.4},
{20.10, 17.3}, {21.10, 16.9}, {22.10, 16.7}, {23.10, 16.5}};
```

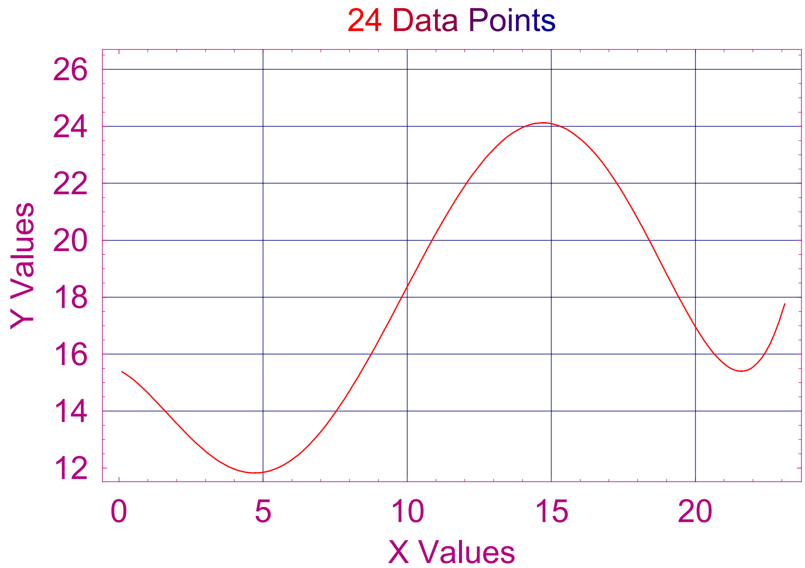
```
fitfunc[x_] = NonlinearFit[data5,
      (a x^5 + b x^4 + c x^3 + d x^2 + e x^1 + f),
   (* model 3 rd order *)
      {x}, {a, b, c, d , e, f} (* parameters *)
   ] // Chop[#, 10^-5] &;
```

Input ▹

```
{start, stop} = {Min[#], Max[#]} &[First@Transpose@data5];
p1 = MDPlotFitData[data5, fitfunc[x] , {x, start, stop},
    FrameLabel → {"X Values", "Y Values"},
   PointStyle → {Red, PointSize[0.02]},
   PlotStyle → {Red, Thickness[0.01]}];
```

$\sum (y_i - \hat{y}_i)^2$ Sum of Squared Error : 38.3985

$15.4351 - 0.491482\,x - 0.394687\,x^2 + 0.0988542\,x^3 - 0.00642266\,x^4 + 0.000126882\,x^5$



Here you can see the calculation of the points

Input ▹

```
f[x_] = 15.435092312721808` - 0.4914816885911674` x -
  0.39468663433757684` x^2 + 0.09885419453045054` x^3 -
  0.006422664641486226` x^4 + 0.00012688173457131225` x^5
```

$15.4351 - 0.491482\,x - 0.394687\,x^2 + 0.0988542\,x^3 - 0.00642266\,x^4 + 0.000126882\,x^5$

Input ▹

```
f''[x]
NSolve[f''[x] == 0]
```

$-0.789373 + 0.593125\,x - 0.077072\,x^2 + 0.00253763\,x^3$

{{x → 1.67556}, {x → 9.85182}, {x → 18.8442}}

Input ▹
```
NSolve[f'[x] == 0]
```

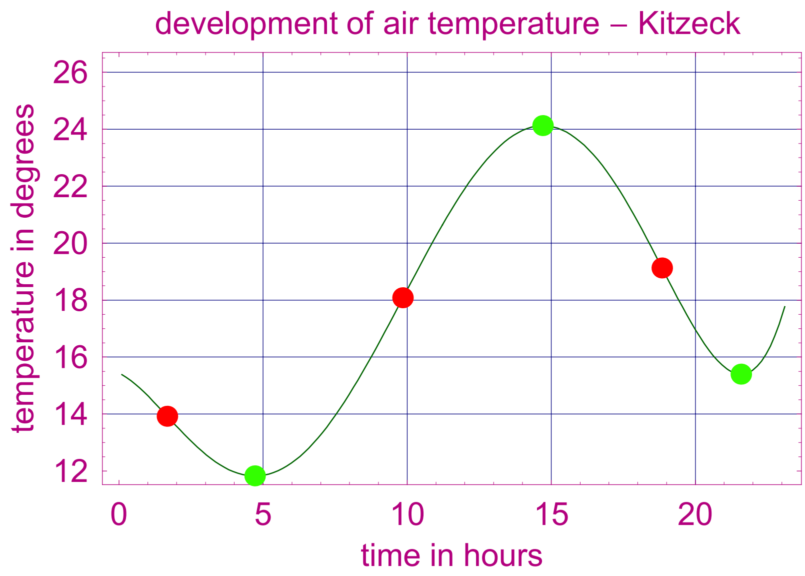
{{x → -0.517459}, {x → 4.71462}, {x → 14.7085}, {x → 21.5898}}

Now you see these points plotted

Input ▹
```
fitfunc[x_] = NonlinearFit[data5,
        (a x^5 + b x^4 + c x^3 + d x^2 + e x^1 + f),
   (* model 3 rd order *)
        {x}, {a, b, c, d , e, f} (* parameters *)
    ] // Chop[#, 10^-5] &;

{start, stop} = {Min[#], Max[#]} &[First@Transpose@data5];
MDPlotFitData[data5, fitfunc[x] , {x, start, stop},
    FrameLabel → {"time in hours", "temperature in degrees"},
  PlotLabel → "development of air temperature - Kitzeck ",
  PointStyle → {Red, PointSize[0.02]},
  PlotStyle → {DarkGreen, Thickness[0.01]},


  Epilog → {PointSize[0.03], Hue[1],
    Point[{1.675556506582579`, fitfunc[1.675556506582579`]}],
    {PointSize[0.03], Hue[1],
     Point[{9.851818774093662`, fitfunc[9.851818774093662`]}]},
    {PointSize[0.03], Hue[1], Point[
      {18.844204977992668`, fitfunc[18.844204977992668`]}]},

    {PointSize[0.03], Hue[0.3],
     Point[{4.714617065981598`, fitfunc[4.714617065981598`]}],
     {PointSize[0.03], Hue[0.3], Point[
       {14.708452733847954`, fitfunc[14.708452733847954`]}]},
     {PointSize[0.03], Hue[0.3], Point[
       {21.589829147881765`, fitfunc[21.589829147881765`]}]}}}];
```

$\sum (y_i - \hat{y}_i)^2$ Sum of Squared Error : 38.3985

$15.4351 - 0.491482\,x - 0.394687\,x^2 + 0.0988542\,x^3 - 0.00642266\,x^4 + 0.000126882\,x^5$



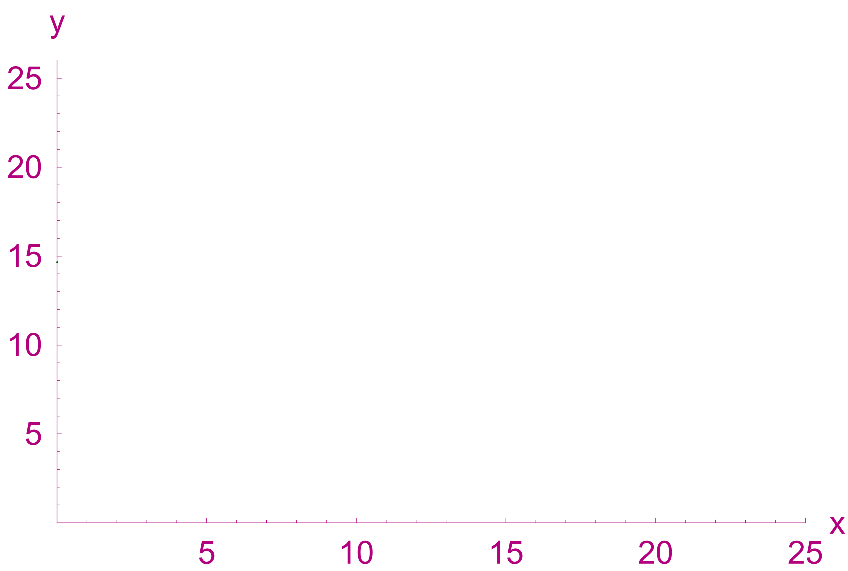
6. Movie

Input ▹
```
T[t1_] = 14.648303365446791` + 0.6450070833886103` x -
  0.7052687113299626` x^2 + 0.11878900938514672` x^3 -
  0.006434215080994428` x^4 + 0.00011114061727570432` x^5
```

$14.6483 + 0.645007\, x - 0.705269\, x^2 + 0.118789\, x^3 - 0.00643422\, x^4 + 0.000111141\, x^5$

Input ▹
```
{start, end, step, k} = {0.001, 25, 0.1, 1.5};
```

Input ▹
```
MDHMovie[MDPlot[T[x], {x, 0, t1}, PlotRange → {{0, 25}, {0, 26}}],
 {t1, start, end, step}]
```



8

Temperature of Hollenegg on the 13th June:

Open / Close Print

Development of the air temperature in Hollenegg within 1 day:

Open / Close

1. The measurepoints shown in a table.

```
More... ;
Clear[f, x];     f[x_] := 10 ; (* enter your f *)
start = 0.23;
stop = 23.23;
step  = 1;

tablevalues = Table[{x, f[x]}, {x, start, stop, step}] // Chop;
MDShowTable[tablevalues,
  {"time in hours", "temperature in degrees"}];
```

Input ▹

time in hours	temperature in degrees
0.23	15.2
1.23	14.1
2.23	13.2
3.23	12.6
4.23	12.5
5.23	12.4
6.23	11.2
7.23	10.9
8.23	12.0
9.23	15.8
10.23	17.0
11.23	19.1
12.23	20.7
13.23	22.5
14.23	24.1
15.23	24.7
16.23	25.6
17.23	25.9
18.23	25.6

19.23	24.3
20.23	21.8
21.23	17.7
22.23	17.4
23.23	16.8

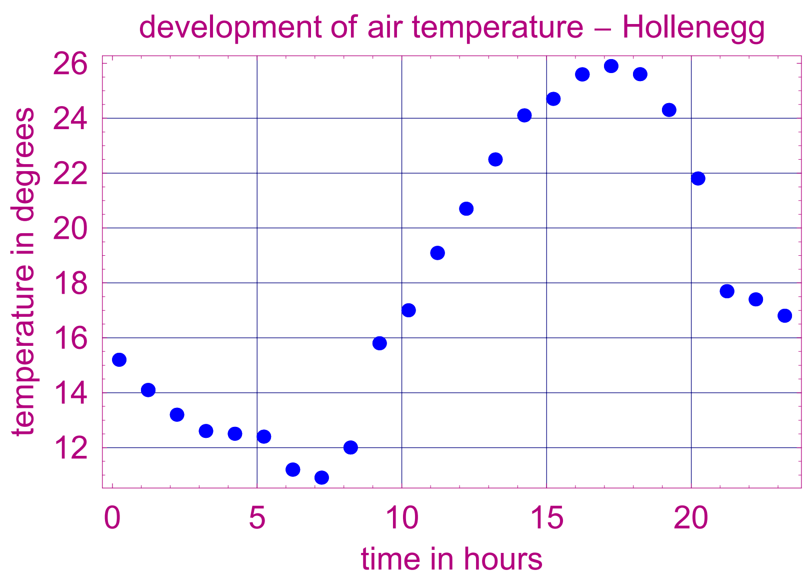
2. The data source

Input ▹
```
data6 = {{0.23, 15.2}, {1.23, 14.1}, {2.23, 13.2}, {3.23, 12.6},
   {4.23, 12.5}, {5.23, 12.4}, {6.23, 11.2}, {7.23, 10.9},
   {8.23, 12.0}, {9.23, 15.8}, {10.23, 17.0}, {11.23, 19.1},
   {12.23, 20.7}, {13.23, 22.5}, {14.23, 24.1}, {15.23, 24.7},
   {16.23, 25.6}, {17.23, 25.9}, {18.23, 25.6}, {19.23, 24.3},
   {20.23, 21.8}, {21.23, 17.7}, {22.23, 17.4}, {23.23, 16.80}};
```

The graphical display
Open / Close

3. The measurepoints plotted

Input ▹
```
MDPlotData[data6,
    FrameLabel →
   {"time in hours", "temperature in degrees"},
  PlotStyle → {Blue, PointSize[.02]},
  PlotLabel -> "development of air temperature - Hollenegg"];
```



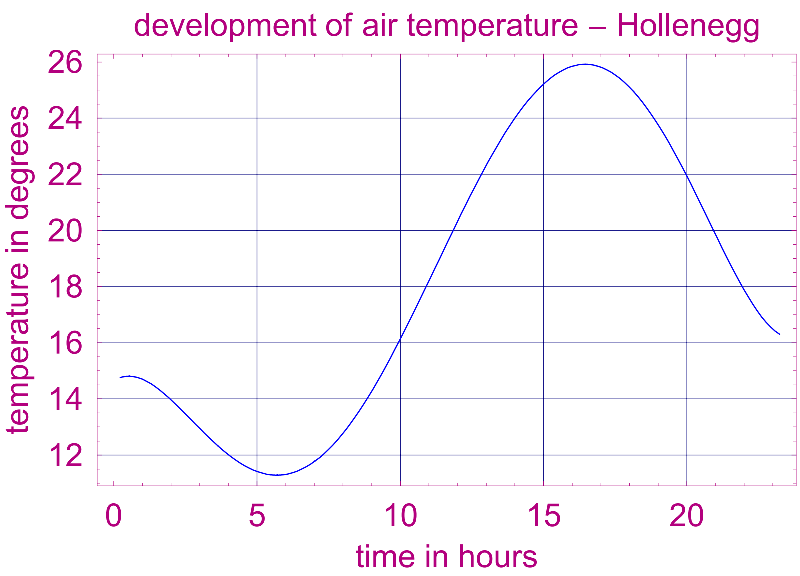
4. Data fitted

Input ▹
```
data6;

fitfunc[x_] = NonlinearFit[data6,
        (a x^5 + b x^4 + c x^3 + d x^2 + e x^1 + f),
    (* model 3 rd order *)
        {x}, {a, b, c, d , e, f} (* parameters *)
    ] // Chop[#, 10^-5] &;
```

Input ▹
```
{start, stop} = {Min[#], Max[#]} &[First@Transpose@data6];
p2 = MDPlotFitData[data6, fitfunc[x] , {x, start, stop},
    FrameLabel → {"time in hours", "temperature in degrees"},
   PointStyle → {Red, PointSize[0.02]},
   PlotStyle → {Blue, Thickness[0.01]},
   PlotLabel -> "development of air temperature - Hollenegg"];
```

$\sum (y_i - \hat{y}_i)^2$ Sum of Squared Error : 11.5489

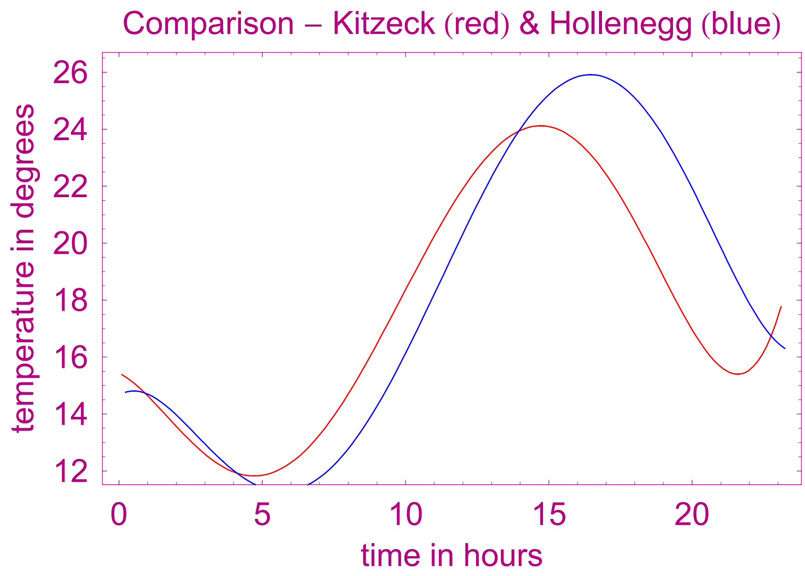
$14.6483 + 0.645007\,x - 0.705269\,x^2 + 0.118789\,x^3 - 0.00643422\,x^4 + 0.000111141\,x^5$



Comparison:
Open / Close

Here we compared the two curves

Input ▹
```
Show[{p1, p2},
  PlotLabel -> "Comparison - Kitzeck (red) & Hollenegg (blue)",
  FrameLabel → {"time in hours", "temperature in degrees"},
  Frame → True, GridLines → False];
```



9 Our team

Open / Close Print

Participants

Open / Close

My partner Martin Fasching and I both are students in the same class.

This picture was taken from http://lyska.net/MathProject/index.html

Martin Fasching
Weissweg 75
8454 Arnfels
Austria
e-mail: martin6866@gmx.at

Dietmar Kappel
Steinriegel 25
8442 Kitzeck
Austria
e-mail: dietmar.kappel@gmx.at

Jens Schatten
Bettina von Arnim Gymnasium
Dormagen
Germany

Our Experience with the Project

Open / Close

We've had, we have and I'm sure we will have great experiences with the project. This programm and project made us become enthusiatic for what we'd never had thought before. The project brings us to places we never been to before and it shows how good people can work together from different regions. We stayed in contact with our partners in Dormagen and we learned a lot of new things which we wouldn't have learned without this project. The project made especially me become enthusiastic for the programm '*Mathematica*' and '*M@th Desktop*' and I think my partner also. I didn't know about the possibilities which you have with '*Mathematica*' and I'm sure I'll have great fun with it in the future.

Dietmar Kappel,

Martin Fasching

New Chapter