

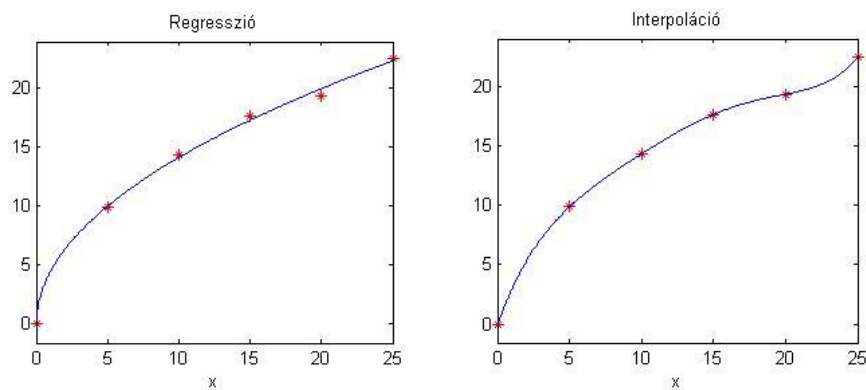
## 8. REGRESSION

---

In engineering practice, countless physical phenomena are measured and examined using digital instruments and the measurement values are stored at discrete points. A typical civil engineering example from solid mechanics could be the tensile test, when a certain material or structural element is tested to substantiate its tensile strength. Another example from geodesy, the measurements of a full-wave laser scanner which are stored at discrete values.

In the case of regression, we already have knowledge about the function (or model) of the physical phenomenon examined and we would like to determine the parameters of the function from our measured values. For example, if the phenomenon can be modelled using a quadratic function in the form  $ax^2 + bx + c$ , we can use the measured values to determine the parameters  $a, b$  and  $c$ . Usually, we have more measurements than parameters, so these problems lead to overdetermined systems which are also many times linear. The overdetermined nature also means that our resulting function will not go through any of the measurement points, but it will try to get as close to all of them as possible.

As for interpolation, we would like to have values where we have no measurements. This can mean values between measured points (interpolation) or values beyond our measurement interval (extrapolation). In these cases, we are trying to find a function that has the measured values as its function value at the measured points and adequately describes the behavior between the points.



### THE QUALITY OF REGRESSION

---

In the case of regression, we seek for the 'best' fitting function. How can we define which is the best fitting one? For this we should determine the differences of each measured point, and the function value of the fitted function at the same location. These are called the residuals ( $r_i$ ):

$$r_i = y_i - f(x_i)$$

$i=1..n$ , where  $y_i$  is the measurement value at  $x_i$  and  $f(x_i)$  is the function value of the resulting regression model.

The residuals should be minimized in some way by applying a single metric to all points. However, only summing the residuals might not work well in all cases. If we

have similar positive and negative residuals, they will cancel out in the summation which will result in a very high goodness of fit, but falsely. We could instead use the absolute value of the residuals, but this would result in an ambiguous solution, that is, more than one parameter set of a certain model would have the same best fit. The best solution is to use the sum of the squares of the residuals, as this way the solution will be unique (only one set of parameters provides the best fit for a model). The sum of squares of the residuals must be minimized:

$$S = \sum_{i=1}^n r_i^2 = \sum_{i=1}^n (y_i - f(x_i))^2$$

Local goodness of fit can be measured with the individual residuals, while globally we can use the empirical standard deviation of the residuals:

$$\sigma = \sqrt{\frac{S}{n - np}} = \sqrt{\sum_{i=1}^n \frac{(y_i - f(x_i))^2}{n - np}} = \sqrt{\sum_{i=1}^n \frac{r_i^2}{f}}$$

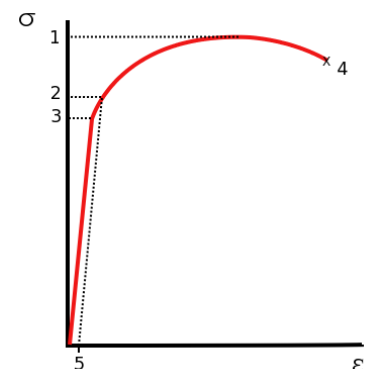
where  $n$  is the number of measurements and  $np$  is the number of parameters in the regression model. Sometimes,  $n - np$  is also called the number of redundant measurements, surplus measurements or degree of freedom ( $f$ ). Which means the number of measurements beyond what is absolutely needed to solve the problem.

### LINE FITTING

Let's look at an example from solid mechanics! Measurements from a tensile test on concrete reinforcing steel bar must be processed (a highest strength steel). During the test, the stress continuously increases up to a maximum value, then decreases, and then the material is destroyed (breaks). See the stress-strain curve below!

From the examination, we determine the following:

- 1) Total elongation (strain) at fracture (4 in the figure)
- 2) Ultimate tensile strength (1 in the figure) (UTS or  $R_m$ )
- 3) Limit of elastic deformation (3 in the figure)<sup>1</sup>
- 4) Modulus of Elasticity (Young-Modulus=slope) (E)
- 5) 0.2% offset yield strength (2 in the figure), (YS or  $R_{p0.2}$ )



Yielding occurs where the initial linear region transitions to the non-linear portion. This transition does not occur always at a clearly visible well-defined point. The most common method is to draw a line parallel to the modulus line at an offset strain of 0.2% (5 in the figure). The intersection stress becomes what is defined at the “0.2% offset yield strength”.

To solve the problem, first load our measurement data from the **tensiletest.txt** file. In this file, the first column contains the measured deformations, the strain ( $\epsilon$  - %), and the second column contains the corresponding stress ( $\sigma$  - Mpa) values.

<sup>1</sup> With no permanent (plastic) deformation, the metal returns to its original shape.

```

> % Tensile test
> clc; clear all; close all;
> data = load('tensiletest.txt');
> x = data(:,1); % strain, epsilon
> y = data(:,2); % stress, sigma
> figure(1)
> plot(x,y, '.')

```

It is easy to answer to the first two questions. To determine where the **total elongation** where the material breaks, we have to take the last measurement result, which is also the maximum strain value. To determine the **ultimate tensile strength**, we need to find the maximum stress ( $\sigma$ ).

```

> disp('Total elongation at fracture:')
> x(end) % same as max(x) => 0.2644 %
> disp('Ultimate tensile strength:')
> max(y) % 668.3606 Mpa

```


So, the material broke at 26.4% relative deformation, and the maximum stress, the ultimate tensile strength, was 668 MPa. To determine the **modulus of elasticity**, we must find the **limit of elastic deformation section** where the relationship between the strain and the stress is linear (Hook's law), and we must fit a straight line to it. The slope of this line will be the value of the modulus of elasticity. To do this, let's zoom in on the figure a bit, up to 0.6% deformation and 400 MPa stress!

```

> axis([0 0.006 0 400])

```

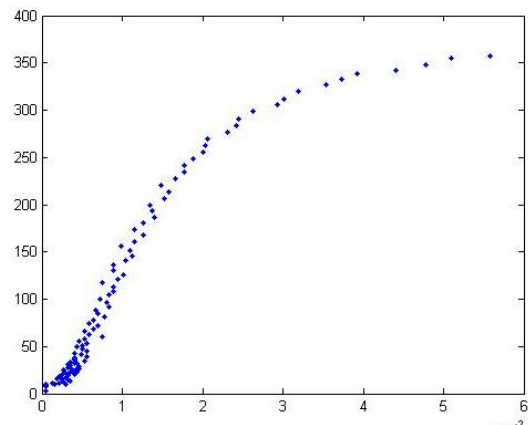
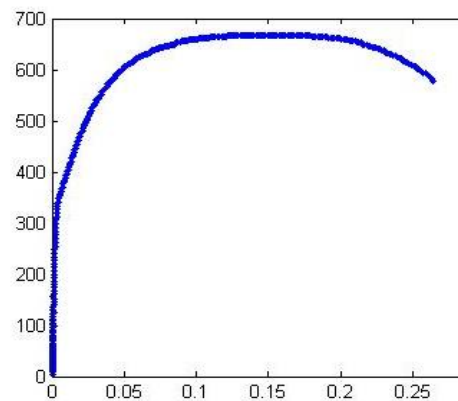
The figure above shows that at the beginning of the measurement, near the origin, the measurement cannot be considered linear due to the uncertainty of the instrument, so for a straight fit, the beginning of the data must be cut off. Now we cut it, which is less than 0.02 %. The upper end of the flexibility limit (proportionality limit) must also be found, let's take this as 0.15%.

Note: the end of the linear section was determined based on the figure. By clicking on the 'data cursor' button , we can query the data of each point. Using logical indexing, we can select the points of the linear section where the x-coordinate is greater than 0.0002 and less than 0.0015!

```

> cond1=and(x>0.0002,x<0.0015); % condition
> x1 = x(cond1);

```

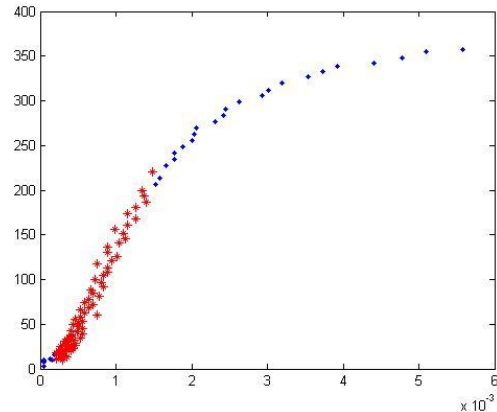


```
> y1 = y(cond1);
> hold on;
> plot(x1,y1,'r*');
```

Let's check whether there really is a linear relationship between the selected points. For this, we calculate the linear correlation coefficient:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

```
> xs = x1-mean(x1)
> ys = y1-mean(y1)
> r = sum(xs.*ys)/sqrt(sum(xs.^2)*sum(ys.^2)) % 0.9805
```



Or the same using Matlab's built-in **corr2** command (from Image Processing Toolbox).

```
> r = corr2(x1,y1) % 0.9805
```

The closer the absolute value of the correlation coefficient is to 1, the more linear the relationship between the two variables. Since this value is now 0.98, we can consider the relationship linear and fit a straight line to it. For this, let's look at the equation of the line:  $y = m \cdot x + b$ . There are two unknowns  $m$ , the slope of the line and the y-intercept  $b$ , where the line intersects the y axis. Based on the selected measurements, we have 75 related x,y values, based on which we can write 75 equations, which are linear with regard to the unknowns ( $m,b$ ):

$$\begin{matrix} m \cdot x_1 + b = y_1 \\ m \cdot x_2 + b = y_2 \\ \vdots \\ m \cdot x_{75} + b = y_{75} \end{matrix} \quad \text{In matrix form} \quad (A \cdot x = B): \quad A = \begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \vdots \\ x_{75} & 1 \end{pmatrix}; B = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{75} \end{pmatrix}$$

In the equation system above, there are two unknowns ( $m,b$ ) and 75 equations, that is, we have 73 extra measurements. This is an overdetermined system of linear equations, where we want to obtain the solution with the smallest error by minimizing the sum of the squares of the remaining errors, the residuals. Based on the previous classes, we can use, for example, the  $x=A \setminus B$  command, which uses QR decomposition in overdetermined cases, or  $x=\text{pinv}(A)*B$ , which uses SVD decomposition.

First, we need to generate the matrix A, with the coefficients of the unknowns. In the first column, there will be the coefficients of  $m$ , in this case the values of  $x_i$  ( $x_i^1$ ), and in the second column, the coefficient of  $b$ , which is always 1, for the sake of simplicity, this can also be written as  $x_i^0$ .

```
> A = [x1.^1 x1.^0]
> B = y1;
> mb = A \ B
> % parameters of line
> m = mb(1)
> b = mb(2)
> % equation of line
> f = @(x) m*x+b
> hold on; fplot(f,[0 0.0015],'g','Linewidth',3);
> axis([0 0.006 0 400])
```

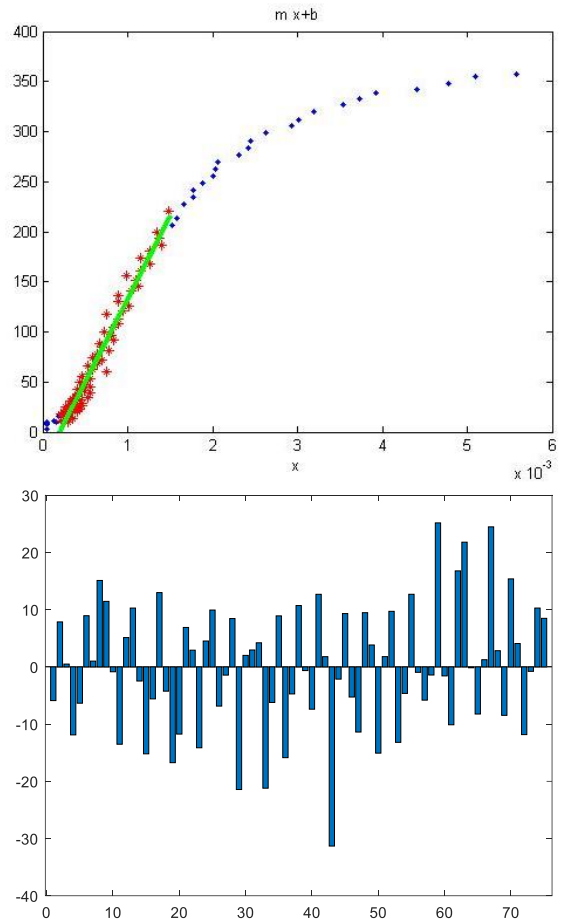
The slope of the fitted line will be the value of the modulus of elasticity or Young-modulus:

```
> format long;
> E = m % E = 165626.174495478
```

In other words, the value of the elasticity modulus  $E$  determined is: 165 626 N/mm<sup>2</sup> (MPa).

Let's determine the accuracy of the fitting and the empirical standard deviation! For this, the residuals must first be calculated, and then the sum of squares of the residuals. Afterwards, the number of redundant measurements must also be determined (degree of freedom), i.e. the difference between the number measurements and the number of parameters to be determined (which is now 2:  $m, b$ ). Let's plot the residuals on a bar chart!

```
> % residuals
> r = y1 - f(x1);
> figure(2); bar(r)
> % sum of squares of the residuals
> format shortG;
> S = sum(r.^2) % 8935.9
> % degree of freedom, f = n-np
> n = length(x1) % 75 - number of
measurements
> np = length(mb) % 2 - number of
parameters
> f = n-np % degree of freedom
> % empirical standard deviation
> estd = sqrt(S/f) % 11.064 Mpa
```

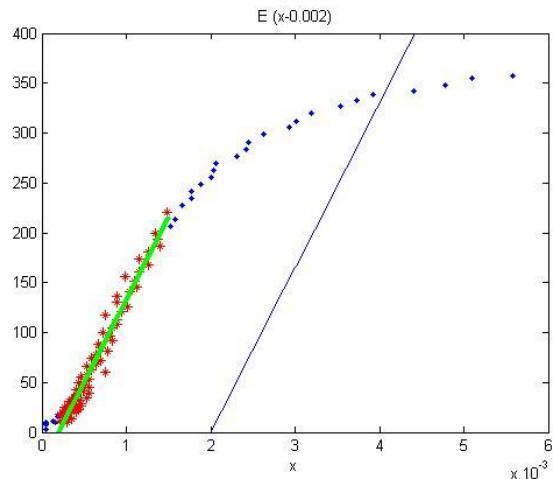


### PARABOLA FITTING

The determination of the yield strength in this case is not clear from the figure, this material does not have a clearly visible yield strength point. In such cases, it is customary to use the conventional  $R_{p0.2}$  yield strength, which is the stress value corresponding to a 0.2% permanent deformation. This can be determined from the stress-strain diagram by drawing a line parallel to the modulus line at an offset strain of 0.2%. The intersection stress becomes what is defined at the “0.2% offset yield strength”. Let's define this line and plot it in the figure!

```
> % 0.2% offset yield strength
> line2 = @(x) E*(x-0.002)
> figure(1); fplot(line2,[0 0.006])
> axis([0 0.006 0 400])
```

How do we determine the intersection of this line and the measurements of the stress-strain diagram? A good solution is to fit a function to the non-linear section, where the intersection point is also located, and find the intersection point of the two functions! To do this, select the points of the non-linear section shown in the figure, i.e. the points between  $x > 0.0015$  and  $x < 0.006$ !



```
> % Section between 0.0015 and 0.006
> cond2=and(x>0.0015,x<0.006);
> xp = x(cond2);
> yp = y(cond2);
> plot(xp,yp, 'm*')
```

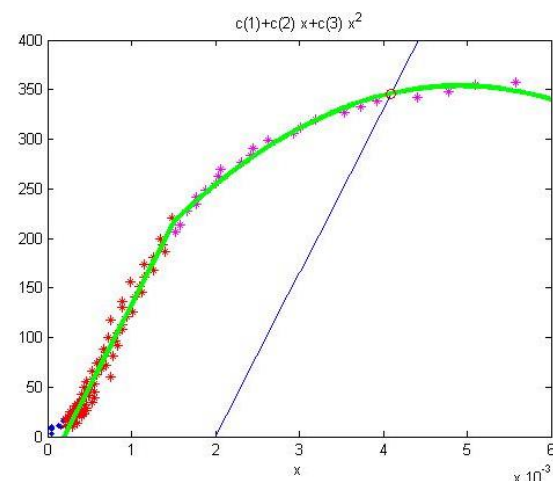
A total of 23 points meet the above condition. Now we fit a quadratic polynomial, a parabola to the points, in the form  $y = c_0 + c_1 \cdot x + c_2 \cdot x^2$ !

$$\begin{matrix}
 y_1 = c_0 + c_1 x_1 + c_2 x_1^2 \\
 y_2 = c_0 + c_1 x_2 + c_2 x_2^2 \\
 \vdots \\
 y_n = c_0 + c_1 x_n + c_2 x_n^2
 \end{matrix}
 \quad \text{In matrix form:}
 \quad
 A = \begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 \end{pmatrix};
 b = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}$$

To fit the parabola (quadratic polynomial), we have to create the appropriate coefficient matrix again (the coefficients of the unknown  $c_0$ ,  $c_1$  and  $c_2$ ) and solve an overdetermined system of linear equations! We have 3 unknowns and we can write 23 equations, so we have a total of 20 extra measurements.

```
> A = [xp.^0 xp.^1 xp.^2]
> b = yp;
> % solution of an overdetermined system of linear equations
> c = A\b
> % equation of the fitted parabola
> f2 = @(x) c(1) + c(2).*x + c(3).*x.^2
> fplot(f2,[0.0015 0.006], 'g', 'Linewidth', 3)
> axis([0 0.006 0 400])
```

Now we just need to find the intersection point! Based on our previous studies, we can do this easily. At the intersection point:  $f(x)=g(x)$ . By rearranging the equation to zero, we can find the roots of the nonlinear equation  $h(x) = f(x)-g(x) = 0$ , for example using the **fzero** built-in function!



```
> % find 0.2% offset yield strength
> % line2(x) = f2(x), so
> h(x) = line2(x) - f2(2)=0
> h = @(x) line2(x) - f2(x)
> sectionp = fzero(h,0.004)
> YS = f2(sectionp) % 345.82 MPa
> plot(sectionp, YS, 'ro')
```

The conventional  $R_{p0.2}$  yield strength, which is the stress value corresponding to a 0.2% permanent deformation was found to be 346 MPa in our measurement.

Similarly, we can fit polynomials of a higher degree (third, fourth, etc.) to our data. However, we have to be careful with higher degree polynomials, because our coefficient matrix will be badly conditioned and our solution will be uncertain. The polynomial may fit perfectly to our measurement points, but oscillations may occur between them. We will see an example of this with interpolation.

Let's check the goodness of fitting in this case too, by calculating the empirical standard deviation!

```
> % empirical standard deviation for parabola fitting
> rp = yp-f2(xp);
> Sp = sum(rp.^2);
> n = length(xp) % 23
> np = length(c) % 3
> fp = n-np % 20
> estdp = sqrt(Sp/fp) % 6.519 MPa
```

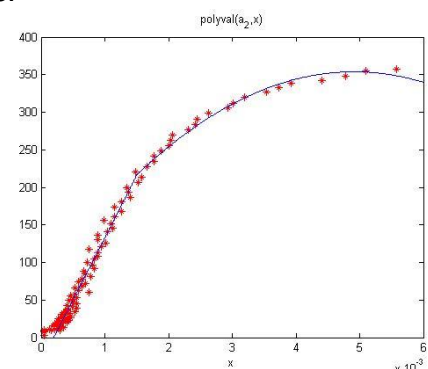
### POLYNOMIAL FITTING WITH MATLAB'S BUILT-IN FUNCTIONS (POLYFIT, POLYVAL)

In order to determine the elasticity modulus, in the first part of the task, a line fitting was required, which corresponds to a first-degree polynomial, and in the second part of the task, we fitted a second-degree polynomial.

In Matlab, there is a command (**polyfit**) that can be used to fit a polynomial of any degree to the measured points using the method of least squares. The result of the command will be a vector containing the coefficients of the polynomial from the term of the highest degree back to the constant term.

There is another related command, **polyval**, which calculates the value of a polynomial at an arbitrary point, given the vector containing its coefficients. The **polyval** command can be called with a specific x coordinate, but it is often advisable to define a function with an independent variable x for further use. Let's see how we could have solved the polynomial fits from the previous task with these commands!

```
> % line fitting (first degree polynomial)
> a1 = polyfit(x1,y1,1)
> p1 = @(x) polyval(a1,x)
> % fitting a parabola (quadratic polynomial)
> a2 = polyfit(xp,yp,2)
> p2 = @(x) polyval(a2,x)
>
> figure(3); plot(x,y,'r*'); hold on;
> fplot(p1,[0 0.0015]);
> fplot(p2,[0.0015 0.006])
> axis([0 0.006 0 400])
```



DETERMINATION OF LINEAR PARAMETERS OF A GENERAL FUNCTIONS

Of course, not only polynomial fits can be solved in the same way, solving many other problems also leads to linear regression. This is the case, for example, if we are looking for the amplitude of a harmonic vibration movement (a relationship that can be described by a sine-cosine function), or if we need to determine the linear parameters of any other function. Let's look at two examples of this, where the unknown parameters to be determined are  $a, b, c$ .

$$y = a \cdot \sin(\pi \cdot x) + b$$

$$y = a \cdot |x - 0.5|^{\frac{3}{2}} + b \cdot \cos(x) + c \cdot e^x$$

Of course, these can also be solved with linear regression, as the parameters are in linear form! The  $x, y$  coordinates are known, so any function of them can be determined, the coefficients can be calculated. Let's look at the coefficient of unknowns in the above two cases!

Unknown parameters:	a	b	c
Equations:	Coefficients of parameters		
$y = a \cdot \sin(\pi \cdot x) + b$	$\sin(\pi \cdot x)$	1	-
$y = a \cdot  x - 0.5 ^{\frac{3}{2}} + b \cdot \cos(x) + c \cdot e^x$	$ x - 0.5 ^{\frac{3}{2}}$	$\cos(x)$	$e^x$

Let's look at the first example in practice, for 5 given points, with known  $x, y$  coordinates!

x	0	0.25	0.5	0.75	1
y	0	0.5	1	0.5	0

The shape of our hypothesis function is as follows:

$$y = a \cdot \sin(\pi \cdot x) + b$$

Let's determine the parameters  $a, b$  with linear regression! Based on 5 points, we can write 5 linear equations for the parameters.

$$\begin{aligned}
 y_1 &= a \cdot \sin(\pi \cdot x_1) + b \\
 y_2 &= a \cdot \sin(\pi \cdot x_2) + b \\
 y_3 &= a \cdot \sin(\pi \cdot x_3) + b \\
 y_4 &= a \cdot \sin(\pi \cdot x_4) + b \\
 y_5 &= a \cdot \sin(\pi \cdot x_5) + b
 \end{aligned}$$

In matrix form:

$$A = \begin{pmatrix} \sin(\pi \cdot x_1) & 1 \\ \sin(\pi \cdot x_2) & 1 \\ \sin(\pi \cdot x_3) & 1 \\ \sin(\pi \cdot x_4) & 1 \\ \sin(\pi \cdot x_5) & 1 \end{pmatrix}; b = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{pmatrix}$$

Solution in Matlab:

```

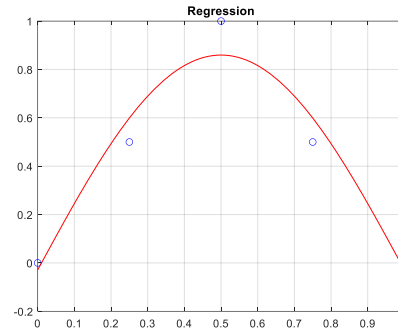
> xy = [0,0.25,0.5,0.75,1;
>       0,0.5,1,0.5,0]';
> x = xy(:,1); y = xy(:,2);
> figure(1); plot(x,y,'bo')
> grid on; title('Regression')
> % Coefficient matrix A
    
```



```

> A = [sin(pi*x), ones(size(x))]
> % parameters using the least squares
  method
> c = A\y
> a = c(1) % 0.8887
> b = c(2) % -0.0291
> % plot function
> f = @(x) a*sin(pi*x)+b
> hold on; fplot(f,xlim,'r')

```



### NONLINEAR REGRESSION TRANSFORMED INTO LINEAR FORM

In reality, there are many physical phenomena where the relationship between quantities is not linear. For example, air density ( $\rho$ ) as a function of height ( $h$ ) can be modeled with an exponential function:  $\rho = k \cdot e^{m h}$ , the speed  $v$  of a dropped object as a function of the distance  $x$  can be described with the following function:  $v^2 = 2 g x$ .

There are many nonlinear functions, but now we will only deal with those that can be transformed to find the parameters by solving a system of linear equations using the method of least squares. Such, among others, is

- power function:  $y = k x^m$
- exponential function:  $y = k e^{m x}$  or  $y = k 10^{m x}$
- reciprocal function:  $y = \frac{1}{m x + c}$

Algebraic polynomials are also like this, we have already seen their fitting in the previous example. In each of the three examples mentioned above, there are two unknowns,  $k$  and  $m$ , but these are mostly presented in non-linear form. The question is, how can we rewrite the above functions into a linear form in terms of the parameters we are looking for?

Generally, by introducing new variables, we can transform the two-variable nonlinear equation into a linear one, so that the new variables (which can be derived from the original variables) are included in the equations in a linear form. Let's look at an example of this, let's transform the **power function** into a linear form for fitting! To do this, first let's take the natural base logarithm of both sides!

$$\ln(y) = \ln(k x^m) = m \ln(x) + \ln(k)$$

Let's introduce new variables to get a linear form  $Y = c_1 X + c_2$ . The new variables should be:  $Y = \ln(y)$ ,  $X = \ln(x)$ ,  $c_1 = m$ ,  $c_2 = \ln(k)$ :

$$\begin{aligned} \ln(y) &= m \ln(x) + \ln(k) \\ Y &= c_1 X + c_2 \end{aligned}$$

The above form is suitable for linear regression to obtain the value of  $c_1, c_2$ . Then, based on  $c_1, c_2$ , the unknown parameters  $k$  and  $m$  can be easily determined:

$$m = c_1, \quad k = e^{c_2}$$

Many other non-linear equations can be linearized in a similar way. Let's look at one of the examples mentioned in the introduction!

Air density ( $\rho$ ) as a function of height ( $h$ ) can be modeled with an **exponential function**:  $\rho = k \cdot e^{m \cdot h}$ . The following table shows air density values measured at different heights

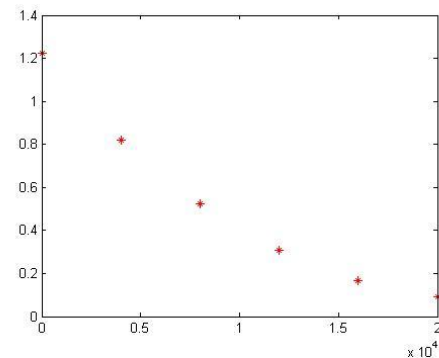
$h$ [m]	1	4000	8000	12000	16000	20000
$\rho$ [kg/m <sup>3</sup> ]	1.225	0.820	0.525	0.309	0.168	0.092

We would like to get answers to two questions:

- 1) What will be the air density on the 8850 m high Mount Everest?
- 2) At what height will the air density be 1 kg/m<sup>3</sup>?

Using linear regression, determine the coefficients  $k$  and  $m$  for the best fitting function! Let's examine locally and globally the curve fitting errors as well! For the solution, load the airdensity.txt file, which contains the above data.

```
> % air density as a function of altitude
> clear all; close all; clc;
> data = load('airdensity.txt')
> % p=k*e^(m*h) - exponential function,
> m,k = ?
> h = data(:,1) % height
> p = data(:,2) % airdensity
> figure(1)
> plot(h,p, 'r*')
```



Let's transform the  $\rho = k \cdot e^{m \cdot h}$  exponential function into linear form!

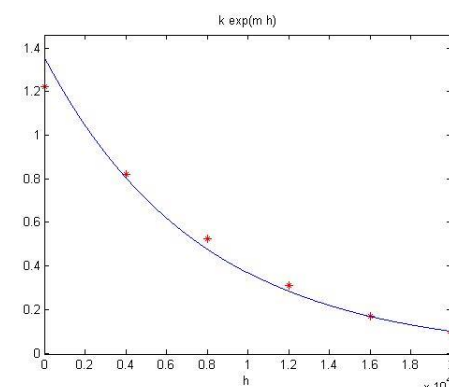
$$\ln(\rho) = m h + \ln(k)$$

$$Y = c_1 X + c_2$$

```
> Y = log(p)
> X = h
> % In the form A*x=b
> A = [X.^1 X.^0]
> b = Y
> % solutions:
> c = A\b % c1 = m, c2 = ln(k)
```

The parameters we are looking for:  $m = c_1$ ,  $k = e^{c_2}$

```
> % fitted parameters
> m = c(1)
> k = exp(c(2))
> % fitted function
> f = @(h) k*exp(m*h)
> % plot
> hold on;
> fplot(f,[0 20000])
```



Let's answer the questions. Using the equation, what will be the air density on the 8850 m high Mount Everest? At what height will the air density be 1 kg/m<sup>3</sup>? The first question can be answered with a simple substitution, for the second, the equation  $f(x)=1$  must be transformed into the form  $g(x)=f(x)-1=0$  and the roots of this nonlinear equation must

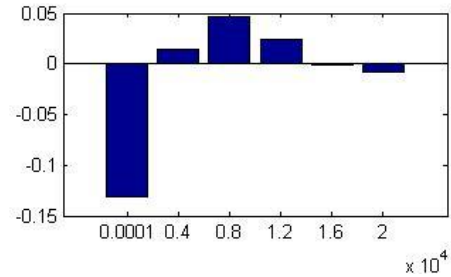
be found. For this, it is also necessary to specify a starting value, which can be taken from the figure as approximately 2000-nek (0.2x10<sup>4</sup>).

```
> % air density at 8850 m
> p8850 = f(8850) % 0.4285
> % At what height will the air density be 1 kg/m^3?
> g = @(h) f(h)-1
> h06 = fzero(g,2000) % 2341.5 m
```

So the density of the air at Mount Everest is 0.4285 kg/m<sup>3</sup>, and at 2342 m the density will be exactly 1 kg/m<sup>3</sup>.

Let's look at the size of the residuals! Plot them on a bar chart and calculate the sum of squares of the residuals and the empirical standard deviation of the residuals.

```
> % residuals
> r = p - f(h)
> % bar chart
> figure(2); bar(h,r)
> % sum of squares of the residuals
> S = sum(r.^2) % 0.0203
> % empirical standard deviation
> n = length(h) % number of data
> np = 2 % number of parameters: k, m
> estd = sqrt(S/(n-np)) % 0.0712
```



Let's summarize some nonlinear equations in a table, which could be similarly solved with linear regression!

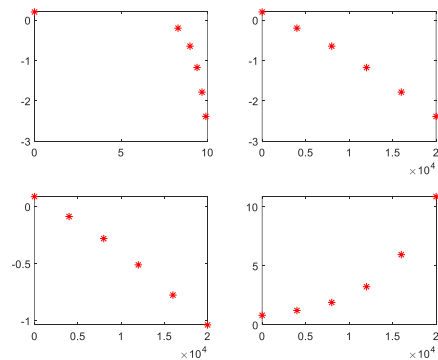
Nonlinear equation	Linearization	In linear form: $Y = c_1 X + c_2$	Searched parameters	Values for linear regression (its image is straight)
$y = k x^m$	$\ln(y) = m \ln(x) + \ln(k)$	$Y = \ln(y), X = \ln(x), c_1 = m, c_2 = \ln(k)$	$m = c_1, k = e^{c_2}$	$\ln(x), \ln(y)$
$y = k e^{m x}$	$\ln(y) = m x + \ln(k)$	$Y = \ln(y), X = x, c_1 = m, c_2 = \ln(k)$	$m = c_1, k = e^{c_2}$	$x, \ln(y)$
$y = k 10^{m x}$	$\lg(y) = m x + \lg(k)$	$Y = \lg(y), X = x, c_1 = m, c_2 = \lg(k)$	$m = c_1, k = 10^{c_2}$	$x, \lg(y)$
$y = \frac{1}{m x + k}$	$\frac{1}{y} = m x + k$	$Y = 1/y, X = x, c_1 = m, c_2 = k$	$m = c_1, k = c_2$	$x, \frac{1}{y}$
$y = \frac{m x}{x + k}$	$\frac{1}{y} = \frac{k}{m} \frac{1}{x} + \frac{1}{m}$	$Y = 1/y, X = 1/x, c_1 = k/m, c_2 = 1/m$	$m = 1/c_2, k = c_1/c_2$	$\frac{1}{x}, \frac{1}{y}$

NONLINEAR EQUATION OF UNKNOWN TYPE<sup>2</sup>

In the previous task, we had a model of what type of function can be used to describe the relationship between quantities. However, there may be cases where we do not know the type of the function describing the relationship. How can we choose the appropriate nonlinear equation to be fitted in such a case? It is advisable to plot the points first. Then we calculate the values of the new variables (X,Y in the previous table) that can be used for the linearization methods described above. Afterwards, we also plot these in separate figures. If there is one for which the points are roughly located along a straight line, then choose that function type for the regression!

Let's look at the previous example, as if we don't know the type of the function we want to fit! In Matlab, the function of the natural base logarithm is **log**, the base 10 logarithm is **log10** and the exponential function is **exp**. For plotting, we use the **subplot** command, which allows us to plot several drawings on one figure. The command can be called in the form **subplot(n,m,i)**, where *n* is the number of rows, *m* is the number of columns, and *i* is the number of the given drawing, counted from left to right and from top to bottom.

```
> x=h; y = p;
> figure(3); subplot(2,2,1)
> plot(log(x),log(y), 'r*')
> subplot(2,2,2); plot(x,log(y), 'r*')
> % approximately straight line
> subplot(2,2,3); plot(x,log10(y), 'r*')
> % approximately straight line
> subplot(2,2,4); plot(x,1./y, 'r*')
```



In the second and third images of the drawing above, the image of the points became approximately straight when  $x$  and  $\ln(y)$  or  $\log(y)$  were plotted. Looking at the table, it can be seen that even if we do not know the function relationship, it would still be worth trying to fit an exponential function.

NEW FUNCTIONS USED IN THE CHAPTER

- axis - Control axis scaling and appearance
- mean - Average or mean value of a vector
- sum - Sum of elements of a vectos
- corr2 - Linear correlation coefficient
- polyfit - Fit polynomial to data
- polyval - Evaluate polynomial
- bar - Bar graph
- subplot - Create axes in tiled positions

<sup>2</sup> Supplementary material for home study