as follows, `p = polyfit(x,y,2)`. After this the vector `p` contains the coefficients. The number 2 tells Matlab to fit a polynomial of order two.

When using the output of the `polyfit()` function it is important to keep in mind that the coefficient are organized in the vector p with the coefficient of the highest power first. Thus if we write the polynomial as

$$y(t) = \sum_{n=0}^{N} a_n x^n$$

with N the highest power, then

$$a_n = p_{N-n+1} \qquad \text{OR} \qquad p_\ell = a_{N-\ell+1}$$

▷ Problem 5.4
The date file `carPos.m` contains a stream of data which is the measured position of a rocket powered car. The sample rate is 10Hz. Fit the data with a quadratic function and from this fitting estimate the acceleration of the car.

▷ Problem 5.5
Show that
(a) $a_n = p_{N+1-n}$.
(b) Show that $p_\ell = a_{N+1-\ell}$ is equivalent to $a_n = p_{N+1-n}$.

§ 5.4 Least squares fitting

In the previous section we did not discuss how we know if we have found the 'best fit'. There is no single answer for this, but the most commonly used criteria for the best fit is the function that minimizes the squares of the difference between the fit function and the data. To state this mathematically, suppose you have a set of data points (x_n, y_n), and that you wish to fit the data with a function $y = f(x, \vec{p})$ where $\vec{p}$ is a list of parameters for the function. We define the *mean squared error* as,

$$\epsilon(\vec{p}) = \frac{1}{N} \sum_{n=1}^{N} \left[f(x_n, \vec{p}) - y_n \right]^2 .$$

The *least squares fit* is the $\vec{p}$ that minimizes the mean squared error.

Using `fminsearch` to find the least squares fit

There is a Matlab function called `fminsearch` that can be used to find the minimum of a function. As an example consider the cosine function If we type

```
>> fminsearch(@cos,1)
ans =
      3.1416
```

we see that the fminsearch function successfully found that 3.1416 is a local minimum of the function. While typing

```
>> fminsearch(@cos,-1)
ans =
      -3.1416
>> fminsearch(@cos,7)
ans =
      9.4248
```

find different local minima. The first argument of fminsearch is the name of the function you wish to minimize and the second argument is an initial guess for the location of the minimum.

We can use fminsearch to find the set of parameters that minimize the mean squared error. The code below will fit a quadratic function to a set of data.

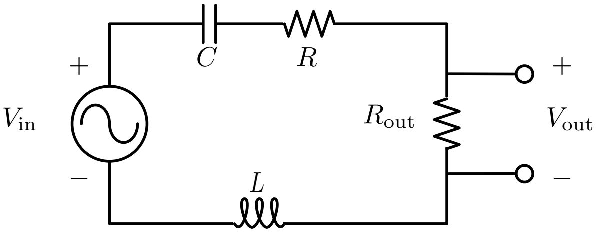
```
function p_fit = fitQuadratic(x,y)
% Fits the data to a function y = a*x^2 + b*x + c.
% p(1) = a
% p(2) = b
% p(3) = c
p_init = [1,1,0]; % Initial guess for the parameters.
p_fit = fminsearch(@compMSE,p_init);

% Here we define the mean squared error function.
% Notice that it is defined inside of fitQuadratic.
    function mse = compMSE(p)
        yfit = p(1)*x.^2 + p(2)*x + p(3);
        mse = mean((y - yfit).^2);
    end
end

```

▷ Problem 5.6

The drag force on an object caused by wind is measured at different velocities of the wind. The file drag.mat is an array with the velocity as the first column of the array and the drag force as the second column. Write a least squares program that fits a function $F(v) = av^b$ for the two fitting parameters a and b. Use this program to determine the parameters a and b that lead to the best fit to the data.

▷ Problem 5.7

Consider the following circuit.



The gain G of the circuit is the ratio $\frac{v_{\text{out}}}{v_{\text{in}}}$ and can be rewritten as

$$G = \left|\frac{v_{\text{out}}}{v_{\text{in}}}\right| = \frac{1}{\sqrt{\alpha^2 + \beta^2\left(\frac{f}{f_0} - \frac{f_0}{f}\right)^2}}$$

where $\alpha = 1 + \frac{R}{R_{\text{out}}}$ and $\beta = \frac{1}{R_{\text{out}}}\sqrt{\frac{L}{C}}$ and $f_0 = \frac{1}{2\pi}\sqrt{\frac{1}{LC}}$. We can see that the maximum of G is $G_{\max} = \frac{1}{\alpha}$ and occurs when $f = f_0$.
(a) Write a least squares fitting function that takes a set of data (f_n, G_n) and estimates the value of α, β and f_0 and then using these values computes the values of R, L and C. In order to make the initial estimate of the parameters find the peak in G and use the peak value and location to estimate f_0 and α. Assume that the value of R_{out} is given.
(b) The data file `RLC.mat` contains an array. The first column of the array is frequency and the second column is gain. The value of $R_{\text{out}} = 2.0\Omega$ for this data set. Use your code to determine the value of R, L and C.

§ 5.5 Discrete Fourier Transform

Theorem: **Discrete Fourier Transform**
Let h_k be a sequence of N data points with k going from 0 to $N-1$. Then we can write h_k as follows.

$$h_k = \frac{1}{N}\sum_{n=0}^{N-1} H_n e^{i2\pi nk/N}$$

where the values of the constants H_n are computed as

$$H_n = \sum_{k=0}^{N-1} h_k e^{-i2\pi nk/N}$$

The H_n are said to be the *discrete Fourier transform* (DFT) of the h_k, while the h_k are said to be the *inverse discrete Fourier transform* (iDFT) of the H_n. There is an algorithm to compute the DFT and iDFT that is fast and so called the Fast Fourier Transform or FFT. So usually the DFT is referred to as the FFT and the iDFT is referred to as the iFFT. We will do so too.