## 9.4 Exercise 4

As discussed in section 4.4 MATLAB interprets a linear least square problem as a special version of solving linear sets of equation, i.e. as the solution to an overdetermined set of linear equations. The best solution will be calculated minimizing the Euclidean distance.
Your first job: Check for the MATLAB-HELP, find a corresponding example and copy it into a small function:

```
function fit_test
  t = [0 .3 .8 1.1 1.6 2.3]';
  y = [.82 .72 .63 .60 .55 .50]';
  E = [ones(size(t)) exp(-t)];
  c = E\y;
  T = (0:0.1:2.5)';
  Y = [ones(size(T)) exp(-T)]*c;
  plot(T,Y,'-',t,y,'o');
end
```

Discuss the commands in this function. Obviously the main job of a user is just to provide the matrix $E$ corresponding to the matrix $M$ in section 4.4. The command `c = E\y;` is doing the whole job calculating the best fitting parameter `c(1)` and `c(2)`.
Your next job is to improve the fitting by adding new test functions.
Up to now we have just performed linear fitting; now we will successively modify our program to additionally perform a non-linear fit with respect to a time constant `tau` in `exp(-t./tau)`. So your next job is to create a nested function within the above program which does the linear fitting. This function should have `tau` as input and `Chi2` (i.e. $\Xi^2$) output.

```
function Chi2 = fit_test
  t = [0 .3 .8 1.1 1.6 2.3]';
  y = [.82 .72 .63 .60 .55 .50]';
  Chi2 = do_linear_optimization(1);
  T = (0:0.1:2.5)';
  Y = [ones(size(T)) exp(-T)]*c;
  plot(T,Y,'-',t,y,'o');

  function Chi2 = do_linear_optimization(tau)
     E = [ones(size(t)) exp(-t./tau)];
     c = E\y;
     yfit = E*c;
     ydiff = y-yfit;
     Chi2 = ydiff'*ydiff;
  end
end
```

Just searching for the minimum of `Chi2` with respect to `tau` will result in the optimal value for `tau` and `c`, thus finding the best fitting results in one non-linear and two linear fitting parameter.
Numerical strategies for finding minima are discussed in chapter 4. A standard routine for finding minima in MATLAB is `fminbnd`. Check in the MATLAB HELP and modify the above function to add the non-linear optimization.

```
function [Chi2, tau_opt] = fit_test_non_linear
  t = [0 .3 .8 1.1 1.6 2.3]';
  y = [.82 .72 .63 .60 .55 .50]';
  tau_opt = fminbnd(@do_linear_optimization,0.3,3);
  Chi2 = do_linear_optimization(tau_opt);
  T = (0:0.1:2.5)';
  Y = [ones(size(T)) exp(-T./tau_opt)]*c;
  plot(T,Y,'-',t,y,'o');

  function Chi2 = do_linear_optimization(tau)
```

```
        E = [ones(size(t)) exp(-t./tau)];
        c = E\y;
        yfit = E*c;
        ydiff = y-yfit;
        Chi2 = ydiff'*ydiff;
     end
   end
```

### HOMEWORK 5

Improve the above fit by using the extended fitting function $f(a, b, c, \tau) = a + b\,x + c\,\exp(-t/\tau)$.
You must be able to discuss the details of your function!