Curve fitting with Linear Least Squares

The objective of curve fitting is to find a mathematical equation that describes a set of data and that is minimally influenced by random noise. The most common approach is the "linear least squares" method, a well-known mathematical procedure for finding the coefficients of (polynomial) equations that are a "best fit" to a set of X,Y data. A polynomial equation expresses the dependent variable Y as a polynomial in the independent variable X, for example as a straight line (Y = **a** + **b**X, where **a** is the *intercept* and **b** is the *slope*), or a quadratic (Y = **a** + **b**X + **c**X^2), or a cubic (Y = **a** + **b**X + **c**X^2 + **d**X^3), or higher-order polynomial. (Other functions can also appear instead of polynomials!) In all these cases, Y is a linear function of the parameters **a**,**b**,**c**, and **d**. This is why this is called "linear" least-squares fit, *not* because the plot of X vs Y is linear. Only for the first-order polynomial is the plot of X vs Y linear. (Sometimes this type of curve fitting is called "curvilinear").

We first provide a brief tutorial on curve fitting using LS. Afterwards, it is your turn to try to solve some curve-fitting problems using Matlab.

TUTORIAL

In this lab we are going to look at taking a function of the form

$$c_1 + c_2 \sin(x) + c_3 \cos(x) + \cdots + c_{2m} \sin(mx) + c_{2m+1} \cos(mx)$$

and fitting it to a given set of data for some value of m. We'll first do an example by hand;

Let's suppose we have the following 8 data points:

(0, 6)	(1, 4)	(2, 3)	(3, 5)
(4, 3)	(5, 4)	(6, -1)	(7, 2)

The first thing we need to do is enter this data into Matlab:

```
x = 0:7
y = [6 4 3 5 3 4 -1 2]
```

We're going to fit this data to a function of the form:

$$y = c_1 + c_2 \sin(x) + c_3 \cos(x)$$

which corresponds to using $m = 1$ in the general form. For each of our (x, y) pairs, we have an equation with the unknowns c_1, c_2 and c_3. For example, for the point (1,4) we have this equation:

$$4 = c_1 + c_2 \sin(1) + c_3 \cos(1)$$

or

$$4 = c_1 + 0.8415\, c_2 + 0.5403\, c_3$$

We have 8 points, so we have 8 equations. Since we have only 3 unknowns (c_1, c_2, and c_3), this system of equations probably does not have a solution. So, we will try to find the values of c_1, c_2 and c_3 that approximate the solution to the system in the least-squares sense.

The first thing we have to do is enter the equations into Matlab. For each equation, the coefficient of c_1 is always going to be 1, the coefficient of c_2 is the sine of the x value for the point, and the coefficient of c_3 is the cosine of x. The right-hand side is the y data value. So, we can create the matrix of coefficients and the right-hand side vector using these Matlab commands:

```
A = [ones(8,1), sin(x'), cos(x')]
b = y'
```

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0.8415 & 0.5403 \\ 1 & 0.9093 & -0.4161 \\ 1 & 0.1411 & -0.9900 \\ 1 & -0.7568 & -0.6536 \\ 1 & -0.9589 & 0.2837 \\ 1 & -0.2794 & 0.9602 \\ 1 & 0.6570 & 0.7539 \end{bmatrix}$$

Look carefully at the matrix `A`. Where is the equation corresponding to the point $(1, 4)$ that we looked at before?

Notice that we took the transpose of `x` and `y` so that the row vectors would be switched to column vectors.

We're solving this equation:

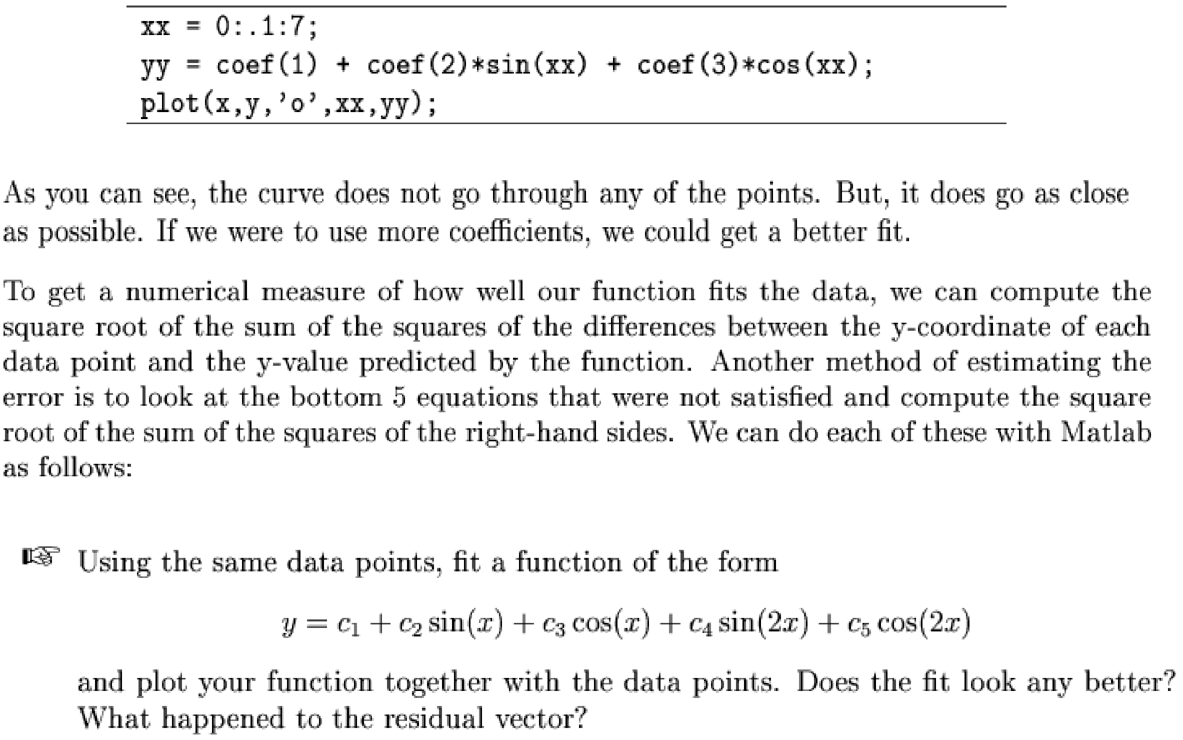
$$Ac = b$$

where c contains the coefficients c_1, c_2, and c_3.

Note that the above SOLE is inconsistent. However, we can try to fit a curve that is as close to the data points as possible, such that the squared error $\|\ Ac - b\ \|^2$ is minimal. In Matlab the least squares solution can be found as

```
coef=A\b;
```

Afterwards, we can plot the fitted curve:



```
xx = 0:.1:7;
yy = coef(1) + coef(2)*sin(xx) + coef(3)*cos(xx);
plot(x,y,'o',xx,yy);
```

As you can see, the curve does not go through any of the points. But, it does go as close as possible. If we were to use more coefficients, we could get a better fit.

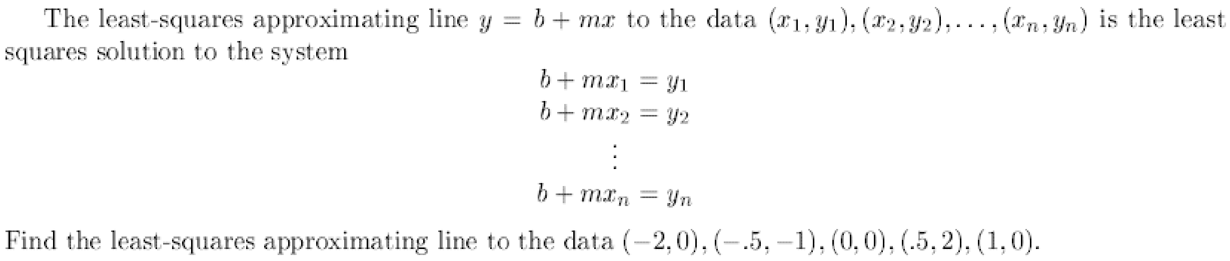
To get a numerical measure of how well our function fits the data, we can compute the square root of the sum of the squares of the differences between the y-coordinate of each data point and the y-value predicted by the function. Another method of estimating the error is to look at the bottom 5 equations that were not satisfied and compute the square root of the sum of the squares of the right-hand sides. We can do each of these with Matlab as follows:

☞ Using the same data points, fit a function of the form

$$y = c_1 + c_2 \sin(x) + c_3 \cos(x) + c_4 \sin(2x) + c_5 \cos(2x)$$

and plot your function together with the data points. Does the fit look any better? What happened to the residual vector?

Straight line fitting:



The least-squares approximating line $y = b + mx$ to the data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ is the least squares solution to the system

$$\begin{aligned} b + mx_1 &= y_1 \\ b + mx_2 &= y_2 \\ &\vdots \\ b + mx_n &= y_n \end{aligned}$$

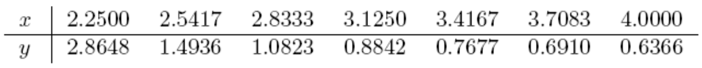
Find the least-squares approximating line to the data $(-2, 0), (-.5, -1), (0, 0), (.5, 2), (1, 0)$.

Plot all the points *(x_i,y_i)* in the plane and then plot the best fitting line on top.

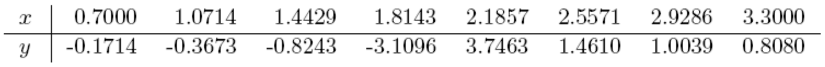
Hint: Use c = *A\b* to solve in least squares sense the system *A*c=b*, where *c* contains the two unknowns *b* and *m*. The matrix *A* has two columns (a column of ones, and a column with the *x*-values) and the vector *b* is a column with the *y*-values.

Fitting a rational function:

The function $y = x/(c_1x+c_2)$ can be transformed into a linear relationship $z = c_1 + c_2w$ with the change of variables $z = 1/y$, $w = 1/x$. Write a function that uses Linear Least Squares to fit data to $y = x/(c_1x+ c_2)$. Test your function by fitting the following sets of data.



x	2.2500	2.5417	2.8333	3.1250	3.4167	3.7083	4.0000
y	2.8648	1.4936	1.0823	0.8842	0.7677	0.6910	0.6366



x	0.7000	1.0714	1.4429	1.8143	2.1857	2.5571	2.9286	3.3000
y	-0.1714	-0.3673	-0.8243	-3.1096	3.7463	1.4610	1.0039	0.8080

Plot all the points *(x_i,y_i)* in the plane and then plot the best fitting curve $y = x/(c_1x+c_2)$ on top.

Solution: For the first data set, $c_1 = 3.1416$, $c_2 = -6.2832$. For the second data set, $c_1 = 3.1415$, $c_2 = -6.2831$.