

Selected Solutions for Exercises in  
Numerical Methods with MATLAB:  
Implementations and Applications

Gerald W. Recktenwald

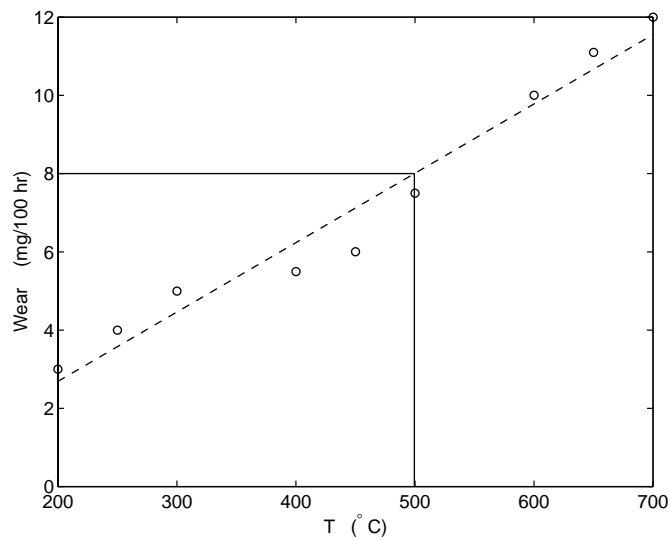
Chapter 9  
Least Squares Fitting  
of a Curve to Data

The following pages contain solutions to selected end-of-chapter Exercises from the book *Numerical Methods with MATLAB: Implementations and Applications*, by Gerald W. Recktenwald, © 2001, Prentice-Hall, Upper Saddle River, NJ. The solutions are © 2001 Gerald W. Recktenwald. The PDF version of the solutions may be downloaded or stored or printed only for noncommercial, educational use. Repackaging and sale of these solutions in any form, without the written consent of the author, is prohibited.

The latest version of this PDF file, along with other supplemental material for the book, can be found at [www.prenhall.com/recktenwald](http://www.prenhall.com/recktenwald).

**9.3** Lipson and Sheth (*Statistical Design and Analysis of Engineering Experiments*, 1973, McGraw-Hill, p. 385) give data on the wear of a particular journal bearing as a function of operating temperature. Their data are contained in the `bearing.dat` file in the `data` directory of the NMM Toolbox. Use the `linefit` function to fit bearing wear as a function of temperature. Plot a comparison of the data and the curve fit on the same axes. From this data what is the limit on bearing temperature if the wear should not exceed  $8\text{ mg}/100\text{hr}$  of operation?

**Numerical Answer:** A plot of the fit is shown below. The temperature limit is  $500^\circ\text{C}$  ( $T = 499.5206^\circ\text{C}$ ).



Plot of solution to  
**Exercise 9-3.**



**9.6** The `xyline.dat` file contains two columns of numbers ( $x$  and  $y$ ) that are reasonably well-fitted by a line. Write an m-file function to perform the following tasks:

- (a) Find the slope and intercept of the line, storing these in a vector called  $c$ . Compute the norm of the residual of the fit,  $r = \|y - Ac\|_2$
- (b) Create two vectors with values within  $\pm 40$  percent of the  $c_1$  and  $c_2$  obtained from the preceding step; that is, create

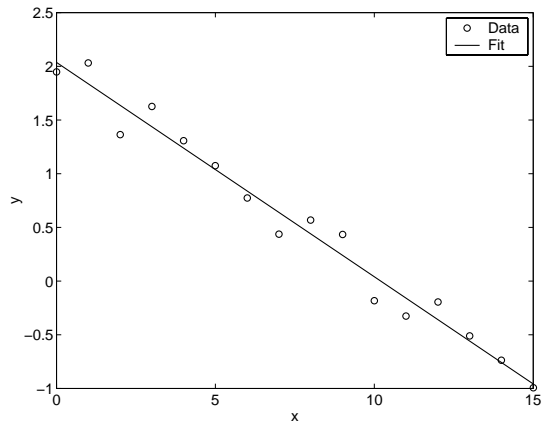
```
c1t = c(1)*linspace(0.6,1.4,20);  
c2t = c(2)*linspace(0.6,1.4,20);
```

- (c) Create a matrix,  $R$ , having elements equal to the norms of the residuals of the fits using all combinations of  $c1t$  and  $c2t$ , i.e.

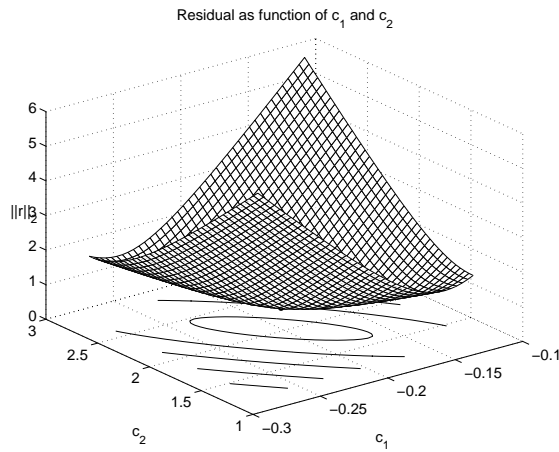
```
R(i,j) = norm( y - (c1t(i)*x + c2t(j)) );
```

- (d) Use the built in `meshc` and `contour` functions to create surface and contour plots with  $R$  on the  $z$  axis,  $c1t$  on the  $x$  axis, and  $c2t$  on the  $y$  axis.
- (e) Explain how the data displayed in the surface and contour plots is consistent with the theory used to develop the  $c$  for the least squares fit.

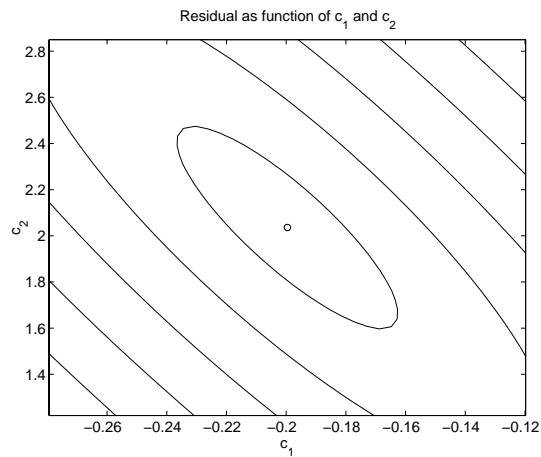
**Partial Solution:** Carrying out the computations in steps (a)–(d) gives the plots on the next page.



**Exercise 9–6.** Plot (a):  
Curve fit.



**Exercise 9–6.** Plot (b):  
Surface plot of residuals.



**Exercise 9–6.** Plot (c):  
Contour plot of residuals.



**9.7** Least squares fitting problems with only one undetermined coefficient lead to particularly simple computational procedures. Derive the equations for finding the (scalar)  $c$  coefficient of the following equations assuming that  $x$  and  $y$  are known vectors of data

(a)  $y = cx$

(b)  $y = cx^2$

(c)  $y = x^c$

For each of these fit equations, write a one-line MATLAB expression to compute  $c$ . Assume that  $x$  and  $y$  are column vectors. *Hint:* Write the overdetermined system and then form the normal equation.

**Solution (a):** For  $m$  data pairs the overdetermined system is

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} c = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

The column vector  $x$  corresponds to the coefficient matrix in Equation (9.15). Form the (only one) normal equation by multiplying both sides by  $x^T$  on the left.

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} c = \begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \iff x^T x c = x^T y$$

Since  $x^T x$  and  $x^T y$  are both scalars we can directly solve for  $c$ .

$$c = \frac{x^T y}{x^T x}$$

Translating the preceding expression to MATLAB gives

```
>> x = ...           (x and y are column vectors)
>> y = ...
>> c = (x'*y)/(x'*x)
```

Does  $c = (x'*x) \setminus (x'*y)$  return the same result?

\_\_\_\_\_  $\diamond$  \_\_\_\_\_

**9.11** 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 an `xlinxFit` function that calls `linefit` to fit data to  $y = x/(c_1 + c_2x)$ . 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

**Typographical Error:** The correct form of the transformed linear relationship is  $z = c_1 + c_2w$ .

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

————— ◇ —————

**9.19** The temperature dependence of the viscosity of a liquid can be modeled with

$$\ln\left(\frac{\mu}{\mu_0}\right) = c_1 + c_2\frac{T}{T_0} + c_3\left(\frac{T}{T_0}\right)^2$$

where  $\mu$  is the viscosity,  $T$  is the absolute temperature,  $\mu_0$  and  $T_0$  are reference values of  $\mu$  and  $T$ , and  $c_i$  are curve fit constants. Use the data in the `H20Visc.dat` file in the `data` directory of the NMM Toolbox to find values of the  $c_i$  for water with  $T_0 = 0^\circ\text{C}$ . The first column in `H20Visc.dat` is the temperature in  $^\circ\text{C}$ , the second column is the viscosity in  $\text{kg}/(\text{m}\cdot\text{s})$ . Convert the temperature to kelvins before performing the fit.

**Numerical Answer:**  $c_1 = 15.5613$ ,  $c_2 = -23.4078$ ,  $c_3 = 7.8189$

————— ◇ —————

**9.23** (Adapted from P.W. Atkins, *Physical Chemistry*, 2nd ed., 1982, W.H. Freeman, San Francisco, problem 27.15, p. 964) The temperature dependence of the reaction rate coefficient of a chemical reaction is often modeled by the Arrhenius equation

$$k = A \exp(-E_a/RT)$$

where  $k$  is the reaction rate,  $A$  is the *pre-exponential factor*,  $E_a$  is the activation energy,  $R$  is the universal gas constant, and  $T$  is the absolute temperature. Experimental data for a particular reaction yield the following results:

$T$ (K)	773.5	786	797.5	810	810	820	834
$k$	1.63	2.95	4.19	8.13	8.19	14.9	22.2

Use a least squares fit of this data to obtain values for  $A$  and  $E_a$  for the reaction. Take  $R = 8314 \text{ J/kg/K}$ .

**Solution:** Vectors of  $k$  and  $T$  are given, the scalars  $A$  and  $E_a$  are unknown. The Arrhenius equation can be written

$$k = c_1 \exp(c_2 x) \quad (\star)$$

where  $c_1 = A$ ,  $c_2 = -E_a$ , and  $x = 1/(RT)$ . Therefore, by fitting Equation  $(\star)$  to the  $(k, T)$  data we obtain  $A$  and  $E_a$ . The curve fit is easily performed with the `expfit` function developed as the solution to Exercise 8.

The curve fit for the given data is computed in the `demoArrhenius` function listed below. Note that the `fitArrhenius` subfunction could be moved to a separate m-file so that it could be called by other m-files or invoked directly from the command window.

Running `demoArrhenius` gives  $A = 1.8717 \times 10^{13}$ ,  $E_a = 2.3803 \times 10^8$ , and the plot on the next page.

```

function demoArrhenius
% demoArrhenius Demonstrate least squares fit to Arrhenius reaction rate equation
%               Uses the fitArrhenius function.   Exercise 9-23

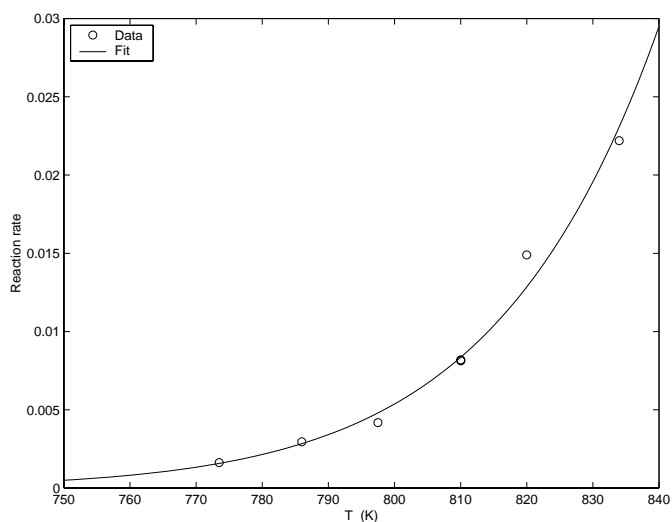
% --- Store data
r = 8314;                % gas constant, J/(K*kg-mol)
t = [773.5 786 797.5 810 810 820 834];
k = [1.63 2.95 4.19 8.13 8.19 14.9 22.2]*1e-3;

% --- Obtain fit, print parameters, and plot fit along with data
[a,ea] = fitArrhenius(k,t,r)
tf = linspace(750,840);
kf = a*exp(-ea./(r*tf));
plot(t,k,'o',tf,kf,'-');
xlabel('T (K)');        ylabel('Reaction rate');    legend('Data','Fit',2);

% =====
function [a,ea] = fitArrhenius(k,t,r)
% fitArrhenius Least squares fit to parameters of arrhenius rate equation
%
% Synopsis: [a,ea] = fitArrhenius(k,t)
%           [a,ea] = fitArrhenius(k,t,r)
%
% Input:   k = vector of reaction rate constants
%          t = vector of temperatures
%          r = (optional) gas constant; Default: r = 8314 J/(K*kg-mol)
%
% Output:  a,ea = parameters in the arrhenius rate equation, k = a*exp(-ea/r/t)
%          "a" is the pre-exponential factor. ea is the activation energy.

if nargin<3, r = 8314; end % default gas constant J/(K*kg-mol)
c = expfit(1./(r*t),k);
a = c(1);
ea = -c(2);

```



Plot of solution to  
**Exercise 9-23.**





**9.26** Write a function to fit data to  $y = c_1x^5 + c_2x^3 + c_3x + c_4$  without calling `polyfit`, `fitnorm`, or `fitqr`. Your function should take two column vectors, `x` and `y`, as input, and return the coefficient vector. It should be self-contained, i.e., it should set up and solve the overdetermined system of equations without calling any auxiliary functions. Test your solution with the following data.

$x$	-3.0000	-1.8571	-0.7143	0.4286	1.5714	2.7143	3.8571	5.0000
$y$	-4.9262	-4.4740	-3.3136	-2.9396	-2.6697	-1.3906	-0.8669	-3.3823

**Numerical Answer:**  $c_1 = -3.116759277 \times 10^{-3}$ ,  $c_2 = 6.286454564 \times 10^{-2}$ ,  $c_3 = 0.3375296900$ ,  $c_4 = -3.181091285$ .

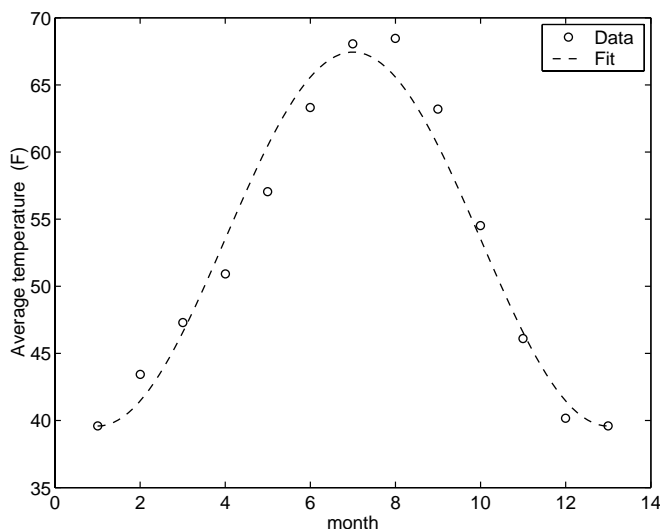


**9.32** The `pdxTemp.dat` file in the `data` directory of the NMM toolbox contains historically averaged monthly temperatures measured at the Portland, OR airport. The file contains four columns of data. The first column is the number of the month (1 through 12). The second through fourth columns are the historically averaged low, high, and average temperature, respectively, for each month. Create a fit of the average temperature to the function

$$T = c_1 + c_2 \sin^2 \left[ \frac{(m-1)\pi}{12} \right]$$

where  $m$  is the number of the month. Plot a comparison of the original data and the fit function. *Hint:* (1) Replicate the January data so that the input has 13 pairs of values with January at the beginning and end of the data set, (2) create a vector of  $\theta = (m-1)\pi/12$  values.

**Partial Solution:** The fit coefficients are  $c_1 = 39.579986$ ,  $c_2 = 27.863363$ . Below is a plot of the fit and the original data.



Plot of solution to  
**Exercise 9–32.**



**9.37** Lipson and Sheth (*Statistical Design and Analysis of Engineering Experiments*, 1973, McGraw-Hill, p. 385) give data on nitrogen oxide (NO) emission from an internal combustion engine as a function of humidity and atmospheric pressure. Their data are contained in the `emission.dat` file in the `data` directory of the NMM toolbox. Obtain a multiple linear regression of this data of the form

$$\text{NO} = c_1 + c_2h + c_3p$$

where  $h$  is the humidity in *grains* per pound of dry air, and  $p$  is the atmospheric pressure in inches of Mercury. What are the values of  $c_1$ ,  $c_2$  and  $c_3$ ?

**Numerical Answer:**  $c_1 = 5028.913856$ ,  $c_2 = -5.347260274$ ,  $c_3 = -117.4130076$ .

—————  $\diamond$  —————