

MIT OpenCourseWare
<http://ocw.mit.edu>

18.02 Multivariable Calculus
Fall 2007

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.

18.02 Problem Set 4

Due Thursday 10/4/07, 12:45 pm.

Part A (15 points)

Hand in the underlined problems only; the others are for more practice.

Lecture 10. Thu Sept. 27 Maxima and minima. Least squares.

Read: 13.5 pp. 878–881, 884–885; Notes LS

Work: 2F/ 1ab, 2; 2G/ 1ab, 4.

Lecture 11. Fri Sept. 28 Second derivative test. Boundaries and infinity.

Read: 13.10 through the top of p. 930; Notes SD.

Work: 2H/ 1ac, 3, 4, 6; 13.10/ 32.

Lecture 12. Tue Oct. 2 Differentials. Chain rule.

Read: 13.6 pp. 889–892[†]; 13.7.

Work*: 2C/ 1abcd, 2, 3, 5ab; 2E/ 1abc, 2bc, 5, 8ab.

[†] WARNING: *Don't mix differentials like df with differences like Δx and Δy . For instance, equations (5), (7), (9) do not make sense. Instead, use (6), (8), (10).*

* *Some of the problems are written so as to depend on the notation for gradient. Look ahead at the definition of gradient in 13.8 (top of p. 910) to know what it is before you do them.*

Part B (25 points)

Directions: Attempt to solve *each part* of each problem yourself. If you collaborate, solutions must be written up independently. It is illegal to consult materials from previous semesters. With each problem is the day it can be done.

Write the names of all the people you consulted or with whom you collaborated and the resources you used.

Problem 1. (Thursday, 11 points: 2+0+3+2+3+1) – Least squares and data analysis.

Parts (b)-(f) of this problem involve the use of Matlab. You may optionally use any other software with similar features, or even a calculator. In that case, indicate what you used, and describe how you proceeded. You must carry out the actual calculations rather than rely on the statistical functions that may be built into the software you are using.

a) Before going to the terminal, read Notes LS and do the following. Consider the row vectors $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]$, $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_n]$ and $\mathbf{u} = [1 \ 1 \ \dots \ 1]$ (n ones). Let $y = ax + b$ be the best-fitting line for the n points (x_i, y_i) . Translate the formula (4) in LS into a single 2×2 matrix equation

$$A\mathbf{z} = \mathbf{r}, \quad \mathbf{z} = \begin{bmatrix} a \\ b \end{bmatrix}$$

Write the entries of A and \mathbf{r} in Matlab-ready form. Don't use summations, instead use, for example, $\mathbf{x} * \mathbf{u}'$ for $\sum x_i$. You will be able to confirm that your formulas are correct by testing them on a concrete example using Matlab in part (c).

b) The worldwide sales of iPods (in million units) for each year from 2001 to 2005 are given below (sources: Wikipedia, Apple quarterly reports):

Years (x_i)	'01	'02	'03	'04	'05
Sales (y_i)	0.125	0.470	1.451	8.263	31.960

(To make the numerical answers easier to read, we take the variable x to be the year minus 2000, so x_i ranges from 1 to 5 for the given data points).

Look at the Matlab directions for plotting at the end of this problem set and make a scatter plot of these data points marked with *'s. (Nothing to hand in; you will do this over again with part (d).)

c) Use Matlab, the data from (b) and the formulas you found in part (a) to find the best line $y = ax + b$ fitted to the points. Compute the difference between the actual value of the data y and the predicted value $y = ax + b$. Report a , b , and the worst case (largest error).

(Optional, but recommended: check that your answer for a and b agrees with the Matlab operation `polyfit(x,y,1)`, which computes the coefficients of the best polynomial of degree 1 fitting the data \mathbf{x} and \mathbf{y} . If $\mathbf{c} = \text{polyfit}(\mathbf{x}, \mathbf{y}, 1)$ then $\mathbf{c} = [a, b]$ is the transpose of the column vector \mathbf{z} in part (a). In this way you can confirm that you did part (a) correctly.)

d) When a new product is launched, in the initial period the sales tend to grow exponentially rather than linearly. Use Matlab to find the best fit of the form $\ln(y) = a_1x + b_1$. (Note: Matlab uses the notation `log` for natural log, and `log10` for log in base 10. So, in Matlab notation, you will be using `log(y)`.) Report your values of a_1 , b_1 .

If you exponentiate this equation, you get

$$y = e^{b_1} e^{a_1x}$$

Compute the difference between y and the predicted value according to this formula and report the worst case (largest error).

e) Hand in a printout that shows on the same plot: the scatter plot of (x, y) labelled with *'s, the straight line fit, $(x, ax + b)$ as a dashed line, and the curve $(x, e^{b_1} e^{a_1x})$, connected by an ordinary line.

f) According to the exponential best fit, how many iPods were sold in 2006? how many will be sold in 2015? (for comparison, the total world population is about 6.6 billions). In fact, the growth is no longer exponential: "only" 46.4 million iPods were sold in 2006.

Problem 2. (Friday, 8 points: 2+2+2+2)

Consider a triangle inscribed in the unit circle in the plane, with one vertex at $(1, 0)$ and the two other vertices given by polar angles θ_1 and θ_2 , in that order counterclockwise.

a) Express the area A of the triangle in terms of θ_1 and θ_2 . What is the set of possible values for θ_1 and θ_2 ?

b) Find the critical points of the function A in this region.

c) By computing the values of A at the critical points and its behavior on the boundary of the region where it is defined, find the maximum and the minimum of A (justify your answer). Describe the shapes of the triangles corresponding to these two situations.

d) Use the second derivative test to confirm the nature of the critical points you found in (b).

Problem 3. (Tuesday, 6 points: 1+2+2+1) (see also 2E/5)

a) Let $w = f(x, y)$, and suppose we change from rectangular to polar coordinates: $x = r \cos \theta$, $y = r \sin \theta$. Using the chain rule, derive the change of variables formula in matrix form:

$$\begin{bmatrix} w_r \\ w_\theta \end{bmatrix} = A \begin{bmatrix} w_x \\ w_y \end{bmatrix},$$

writing the entries of the 2×2 matrix A as functions of r and θ .

b) Use the formulas $r = \sqrt{x^2 + y^2}$, $\theta = \tan^{-1} y/x$ to similarly derive the converse formula:

$$\begin{bmatrix} w_x \\ w_y \end{bmatrix} = B \begin{bmatrix} w_r \\ w_\theta \end{bmatrix},$$

writing the entries of the 2×2 matrix B as functions of x and y .

c) Check that $B = A^{-1}$ by computing the product AB and changing variables.

d) If $w_r = 2$ and $w_\theta = 10$ at the point of polar coordinates $r = 5$, $\theta = \pi/2$, compute w_x and w_y at the same point.

Matlab Directions

You can reach Matlab in MIT Server by clicking on the main menu (lower left icon with the footprint), then Math/Plotting, then Matlab.

Be patient – it takes Matlab several seconds to load. Wait for the prompt symbol: `>>`. You can leave Matlab by typing `exit` [return], or by closing the window. Matlab calculates with matrices and vectors and draws graphs in 2D and 3D. Skip the Introduction and Help documents; as preliminary practice, just read and carry out the following.

Entering matrices and vectors. In Matlab the variables represent matrices and vectors. The symbol `=` assigns the value on the right side of the equation to the symbol on the left. Type each of these lines in order, and see what you get. (Always hit [return] to end a line or command.)

```
A = [1 2 3; 4 5 6; 7 8 9]      (you can use commas instead of spaces: 1,2,3;)
b = [5 2 1]
b'      (transpose: gives the column vector which Matlab calls [5;2;1])
```

Try making a mistake: `C = [1,2,3; 4,5]`. To edit the mistake, press any of the four arrow keys to get the line back. (You can also prepare your commands in a text editor and copy them with the mouse onto the Matlab command line.)

Operations with matrices and vectors

Sum, difference `A+B, A-B` (matrices must be same size)
Product `A*B` (matrices must be compatibly sized)
Powers `A^n` (A times itself n times; A must be square)
Transpose `A'`
Inverse `inv(A)` (or `A^-1`)

Try typing (use the values of A and b above): `A*b` `A*b'` `b*A`

Array operations. Recall that `*` and `^` are product and power operations *for matrices*. Adding a dot before `*` or `^` makes these operations act component-wise. So, if $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]$, then

```
exp(x) = [exp(x1) ... exp(xn)]    (similarly with sin, cos, log, etc.)
x+y = [x1 + y1 ... xn + yn]    (similarly with -)
x.*y = [x1y1 ... xnyn]
x.^m = [x1m ... xnm]    (m can be zero)
```

Colon operator. This generates a vector with equally spaced entries; for example,

```
[0 : 2 : 12] = [0 2 4 6 8 10 12];    [2 : -.1 : 1.6] = [2.0 1.9 1.8 1.7 1.6]
```

Two-dimensional plots in Matlab. Given $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]$, $\mathbf{y} = [y_1 \ \dots \ y_n]$,

```
plot(x,y)    plots the  $n$  points  $(x_i, y_i)$ , joined by solid line segments.
plot(x,y,'--')    plots the  $n$  points, joined by dashed line segments.
plot(x,y,'*')    plots the  $n$  points as individual stars (or dots or circles, etc).
print    gives a print-out of the current screen plot.
```

Try in order (press [return] after each command):

```
x=[0:.1:2]
plot(x,sin(x))
plot(x,4*x.^3)    (this plots  $y = 4x^3$ ; note the need for the array operator)
```

You can also combine two plot commands: for example,

```
plot(x,sin(x),'*',x,cos(x),'--')
```