

# ASSIGNMENT 7 (count as 2 assignments)

Due May 3, 2005 (before start of class)

## Problem 8

This problem is adopted from computer problem 3.5 in Heath's book.

A planet follows an elliptical orbit, which can be represented in a Cartesian  $(x, y)$  coordinate system by the equation

$$ay^2 + bxy + cx + dy + e = x^2.$$

- (a) Write a Matlab program, using Matlab's backslash operator for linear least squares to determine the orbital parameters  $a, b, c, d, e$ , given the following observations of the planet's position:

$x$	1.02	0.95	0.87	0.77	0.67	0.56	0.44	0.30	0.16	0.01
$y$	0.39	0.32	0.27	0.22	0.18	0.15	0.13	0.12	0.13	0.15

In addition to printing the values of the orbital parameters, plot the resulting orbit and the given data points in the  $(x, y)$  plane. Use the "hold on" command to incrementally add drawings to a plot. Remember to use the command "hold off" at the start of the program. The orbit can be plotted using the function `conicSectionPlot.m` which is provided with this assignment. The first input parameter of this function is a vector of orbit parameters  $[a \ b \ c \ d \ e]$ . Read the comments to see how to use this function.

- (b) This least squares problem is nearly rank-deficient. To see what effect this has on the solution, perturb the input data slightly by adding to each coordinate of each data point a random number uniformly distributed on the interval  $[-0.005, 0.005]$  and solve the least squares problem with the perturbed data. Compare the new values for the parameters with those previously computed. What effect does this difference have on the plot of the orbit? Can you explain this behavior?

In generating the randomized data, make sure that you set the seed of the random number generator to value 17 so that I can readily compare your results with mine. This can be accomplished in Matlab using the command `rand('state', 17)`. However you should run your program 10 or more times with that command removed in order to obtain proper valuable insights into the effects of the perturbation.

- (c) Compute the singular value decomposition of the  $10 \times 5$  least squares matrix.
- (d) Use the singular value decomposition to compute the solution to the least squares problem. With the singular values in order of decreasing magnitude, compute the solution using the first  $k$  singular values,  $k = 1, \dots, 5$ . For each of the five solutions obtained, print the values of the orbital parameters and also plot the resulting orbits along with the given data points in the  $(x, y)$  plane. In order to distinguish the various curves, you may want to define a vector of strings such as

```
PlotStyle = ['b--'; 'r--'; 'g--'; 'c--'; 'm--'];
```

If  $k$  is a loop index, then different line style can be obtained using `PlotStyle(k,:)` within the loop.

- (e) Perturb the input data slightly by adding to each coordinate of each data point a random number uniformly distributed on the interval  $[-0.005, 0.005]$ . Compute the singular value decomposition of the new least squares matrix, and solve the least squares problem with the perturbed data as in part *d*. Compare the new values for the parameters with those previously computed for each value of  $k$ . What effect does this difference have on the plot of the orbits? Can you explain this behavior? Which solution would you regard as better: one that fits the data more closely, or one that is less sensitive to small perturbations in the data? Why?
- (f) For simplicity, we have used ordinary least squares in this problem, but in fact all of the data are equally subject to observational errors (indeed  $x$  appears on both sides of the equation), which makes the applicability of ordinary least squares questionable. Reformulate this problem as a total least squares problem and solve the latter using the singular value decomposition as described in Section 3.6.6.

Your program will have several parts. You should write it incrementally. Make sure that the first part is correct before writing the next part. Submit a hard copy of the program with relevant output results. Also submit an electronic copy of your program using the same file naming convention as we have done in the past.

To help you deal with the graphics part of the program (plotting the orbits with roughly the right scales, etc.) a skeleton of the program (name `ellipticalOrbit.m`) is provided for your use. Do not be too surprised if some low-ranking approximations using SVD do not give elliptical orbits.