CS 422: Assignment 6

Due: Friday, May 2, 2008

Scoring for each problem is on a 0-to-5 scale (5 = complete success, 4 = overlooked a small detail, 3 = good start, 2 = right idea, 1 = germ of the right idea, 0 = missed the point of the problem.) One point will be deducted for insufficiently commented code. Unless otherwise stated, you are expected to utilize fully MATLAB's vectorizing capability subject to the constraint of being flop-efficient. Test drivers and related material are posted on the course website `http://www.cs.cornell.edu/courses/cs422/2008sp/`. For P1 and P2 submit output and a listing of all scripts/functions that *you* had to write in order to produce the output. You are allowed to discuss *background* issues with other students, but the codes you submit must be your own.

P1. (Supernovae Luminosity)

Type I supernovae have been found to follow a specific pattern of luminosity (brightness). Beginning a few days after the maximum luminosity this pattern may be described by

$$L(t) = C_1 \exp(-t/\alpha_1) \; + \; C_2 \exp(-t/\alpha_2)$$

where t is the time in days after the maximum luminosity and $L(t)$ is the luminosity relative to the maximum luminosity. The function `[t,Lvals] = LuminData` returns actual data associated with the supernova SN1939A. In particular, `Lvals(i)` is the luminosity at time `t(i)` for `i=1:21`.

By making effective use of MATLAB's nonlinear least squares solver `LSQNONLIN`, produce estimates of C_1, C_2, α_1 and α_2 so that

$$\phi(C_1, C_2, \alpha_1, \alpha_2) = \sum_{i=1}^{21} \left(L(t_i) - Lvals(i)\right)^2$$

is minimum. FYI, α_1 is about 5 and α_2 is about 60. But you may want to experiment with different starting values. Strive for 5 significant digits of accuracy. Also submit a plot that displays the optimal fitting function and the data.

P2. (Intersecting Orbits)

The parametric representation of an ellipse with center (x_c, y_c), x-semiaxis a, y-semiaxis b, and tilt θ is given by

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} x_c \\ y_c \end{bmatrix} + \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} a\cos(t) \\ b\sin(t) \end{bmatrix} \qquad 0 \le t \le 2\pi$$

Write a function `[u,v] = Intersections(E1,E2)` that returns in column vectors `u` and `v` the xy coordinates of those points that reside on both ellipse `E1` and ellipse `E2`. Thus, `u` and `v` may have length 0, 1, 2, 3, or 4. (Empty vectors returned if there are no intersection points.) The input parameters `E1` and `E2` must be structures with five fields: `xc`, `yc`, `a`, `b`, and `theta`. Thus,

```
E = struct('xc',3,'yc',2,'a',1,'b',7,'theta',pi/6)
```

encodes an ellipse that is centered at (3,2), has x-semiaxis 1, y-semiaxis 7, and tilt angle 30 degrees.

Approach this problem as an $n = 2$ nonlinear system. In particular, if $(x_1(t_1), y_1(t_1))$ and $(x_2(t_2), y_2(t_2))$ define `E1` and `E2` respectively, then to find an intersection point you must find $\tilde{t}_1$ and $\tilde{t}_2$ so that $(x_1(\tilde{t}_1), y_1(\tilde{t}_1))$ $= (x_2(\tilde{t}_2), y_2(\tilde{t}_2))$. Use Newton's method with exact Jacobian. Strive for absolute error $\le 10^{-6}$.

Starting values should be obtained graphically. In particular, `Intersections` should display the two ellipses in the same window. Use `ginput` to generate an initial guess as follows. If the mouseclick is inside the intersection region of `E1` and `E2`, then determine the t_1^{opt} value that identifies the nearest point on `E1` and the t_2^{opt} value that identifies the nearest point on `E2`. (Refer to A5.) Use (t_1^{opt}, t_2^{opt}) as the starting value. If the mouse click is outside the intersection region, then `Intersections` should terminate. Thus, your implementation should keep going after intersection *points* until you mouseclick outside the intersection *region.* You may assume an intelligent mouseclicker, i.e., someone who will not mouseclick near a previously determined intersection point. An ellipse "fact sheet" is on the website–it'll help you figure out whether or not a mouseclick is inside an ellipse.

Submit your implementation of `Intersections` by email. Just one file–use subfunctions as required.

P3. (Mars from Earth According to Ptolemy)

Assume the availability of functions $x_E(t)$, $y_E(t)$, $x_M(t)$, and $y_M(t)$ that return the "exact" location of Earth and Mars at time t (days). (Sun at (0,0).) Assume $y_E(0) = y_M(0) = 0$ and $0 < x_E(0) < x_M(0)$, i.e., the planets are aligned along the positive x-axis at the start. Your job is to determine epicycle parameters P_1, r_2, P_2, r_3 and P_3 so that if

$$\begin{aligned} x(t) &= \cos(2\pi t/P_1) + r_2\cos(2\pi t/P_2) + r_3\cos(2\pi t/P_3) \\ y(t) &= \sin(2\pi t/P_1) + r_2\sin(2\pi t/P_2) + r_3\sin(2\pi t/P_3) \end{aligned}$$

then the direction of the vector

$$Q(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$$

is "as close as possible" to the direction of

$$R(t) = \begin{bmatrix} x_M(t) - x_E(t) \\ y_M(t) - y_E(t) \end{bmatrix}$$

over the time interval $[0, 1000]$.

In this epicycle model, think of the Earth at the center of a unit disk whose period of revolution is P_1. Centered on the rim of this disk is a second disk whose radius is r_2 and whose period of revolution is P_2. Finally, centered on the rim of the second disk is a third disk whose radius is r_3 and whose period of revolution is P_3. On the rim of the third disk is Mars and so $Q(t)$ is a vector that points to Mars from the "observation point" (0,0).

For an objective function, use

$$\phi(P_1, r_2, P_2, r_3, P_3) = \sum_{i=1}^{1000} \sin(\theta_i)^2$$

where θ_i is the angle between the 2-vectors $Q(i)$ and $R(i)$. Vector cross-products can be used to compute the sine of an angle between two vectors. Make effective use of `fminsearch`. Getting a good initial guess is important. It might be handy to use the fact that the two planets have approximate periods 365 and 697 (days) and that average Sun-Mars distance is about 1.524 times the average Sun-Earth distance.

Submit all the code (nicely commented) that you use to solve the problem together with output and (as necessary) written comments that explain how you solved the problem.