CS 4220: Assignment 5

Due: Wednesday, April 15, 2009 (In Lecture)

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. For each problem 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. Transit of Earth From Mars

Every so often from our vantage point on Earth, the planet Venus moves across the face of the Sun. This is called a "transit of Venus." It doesn't happen that often because the two orbits are not co-planar: 1874, 1882, 2004, 2012, 2117, 2125, etc. They come in pairs with 8-year spacings. The 2004 transit lasted about 6 hours.

In this problem you are to estimate the duration of a transit of Earth as seen from Mars. To simplify matters we will assume that the location of Earth is given by $(x_E(t), y_E(t))$ where

$$x_E(t) \;=\; \frac{P_E - A_E}{2} \;+\; \frac{P_E + A_E}{2}\cos\left(\frac{2\pi(t+t_E)}{\rho_E}\right) \qquad\qquad y_E(t) \;=\; \sqrt{P_E A_E}\,\sin\left(\frac{2\pi(t+t_E)}{\rho_E}\right)$$

with $P_E = 147.098$ (perihelion in millions of kilometers), $A_E = 152.097$ (aphelion in millions of kilometers), $\rho_E = 365.24$ (period of revolution in days), and $t_E = 70$. Likewise, for Mars we will assume

$$x_M(t) \;=\; \frac{P_M - A_M}{2} \;+\; \frac{P_M + A_M}{2}\cos\left(\frac{2\pi(t+t_M)}{\rho_M}\right) \qquad\qquad y_M(t) \;=\; \sqrt{P_M A_M}\,\sin\left(\frac{2\pi(t+t_M)}{\rho_M}\right)$$

with $P_M = 206.669$, $A_M = 249.209$, $\rho_M = 686.971$ (period of revolution in days), and $t_M = 0$. We situate the Sun at (0,0) and assume that it has radius $r = 695500$ km. Thus, we have coplanar elliptical orbits. It is easy to show that (0,0) is a focus for each ellipse so the orbits are in the Kepler tradition. However, the motion doesn't quite follow Kepler's equal area law.

During the transit, Earth will be seen to move left to right across the Sun's equator. Design a function `fStart(t)` that is zero whenever a transit begins and a function `fEnd(t)` that is zero whenever a transit ends. (You may assume that Earth is a point, which simplifies the start/stop computations.) Using these functions and the MATLAB `fzero` function, compute the length of the first three Earth transits that follow $t = 0$. The computed durations must be accurate to within one minute. Explain how you determined the initial bracketing interval for the `fzero` calls. Submit listings of all the functions/scripts that you needed to solve this problem. Submit relevant output including exploratory plots (if any). Points will be deducted for insufficiently commented code. **Extra Credit**. Assume that the Earth is a disk and assume that the transit is "on" as long as the Earth's disk and the Sun's disk intersect. FYI, the Earth's radius is 3188 km. How do the transit durations change when we model Earth as a disk instead of as a point?

P2. Brightness Local Max

Let $A(\theta)$ be the radian measure of angle Sun-Earth-Mars. If $\theta = \pi/2$, then the Martian sees a "half Earth" similar to a half-moon. Ifθ is very small, then the Earth is very full in the Martian sky. If θ is very close to π, then the Earth is appears as a crescent. Using the orbits given in the previous problem, compute the t-values associated with the first three local maximums for the "brightness" function

$$B(t) \;=\; \frac{f(t)}{d(t)^2}$$

Here, $f(t)$ is the fraction of the Earth's disk that is illuminated at time t (to a Martian) and $d(t)$ is Earth-to-Mars distance. Make effective use of the MATLAB function `fminbnd`. Explain clearly the objective function that you are using and how you determined the initial bracketing intervals. It is OK to approximate f. To that end you may want to use the MATLAB function `quad`. Submit all output, plots, scripts, and functions.