CS 322: Assignment 3

Due: Monday, March 15, 2004 at 4pm.

Do not submit work unless you have adhered to the principles of academic integrity as described on the course website:


Points will be deducted for poorly commented code, redundant computation that seriously affects efficiency, and failure to use features of MATLAB that are part of the course syllabus. In particular, use vector operations whenever possible. Pay attention to the course website for news that relates to this assignment.

Problem A (10 pts) The Maximum Value of a Spline

Complete the following function so that it performs as specified:

```matlab
function [zStar,sMax] = maxSpline(S)
% S is the piecewise polynomial form of a cubic spline s(z).
% Let x(1) and x(n) be the leftmost and rightmost breakpoint.
% sMax is the maximum value of s on [x(1),x(n)] and zStar has the
% property that x(1) <= zStar <= x(n) and s(zStar) = sMax.
% Start by “taking apart” S using unmkpp:
[x,rho,L,K] = unmkpp(S)
% This assigns the breakpoints to x, the number of subintervals to L,
% the degree of the local polynomials plus one to k, and the
% coefficients of the local cubics to rho. rho = (ρij) is an L-by-k
% matrix and its i-th row houses the coefficients of the i-th
% local cubic. In particular, if z satisfies x(i) ≤ z ≤ x(i+1), then
% s(z) = q_i(z) = \rho_{i4} + \rho_{i3}(z - x_i) + \rho_{i2}(z - x_i)^2 + \rho_{i1}(z - x_i)^3
To find the overall maximum of s(z) on [x_1,x_n] you must look at the maximum value of each q_i on [x_i,x_{i+1}], for i = 1:L. This requires you to compare q_i(x_i), q_i(x_{i+1}), and the value of q_i at any point r where q_i'(r) = 0 and x_i ≤ r ≤ x_{i+1}. Submit your implementation of maxSpline. We’ll test it with the script A3A which is available on the website.
```

Problem B (10 pts) The Brightest Crescent

To an observer on Earth, Venus goes through phases like the moon. The “amount” of crescent that you see depends on the Earth-Sun-Venus angle. Run the script ShowCrescent to appreciate this point. Study the function Crescent to understand the underlying geometry. Pay special attention to

- The elongation angle φ. As θ increases from 0 to π, φ increases and then decreases. The maximum elongation of Venus is around 45°.
- How the crescent is specified. For 0 ≤ θ ≤ π, the crescent is bounded by the left edge of the disk and the terminator or “shadow” line. For π ≤ θ ≤ 2π, the crescent is bounded by the right edge of the disk and the terminator. You will be integrating an energy function over the crescent and the limits of integration are determined by the left and right edges of the disk and the terminator.
- The moving coordinate system. Venus is at (0,0,0) and Earth is located at (0, -d_{EV}, 0) where d_{EV} is the Earth-Venus distance (which depends upon θ).
- In the right window we are looking at Venus from the Earth. The disk is defined by the circle x^2 + z^2 = R^2. The y-axis (going into the screen) is perpendicular to this.
Venus would get brighter and brighter as the crescent enlarges except for the fact that it is also getting further away and the inverse square law “kicks in.” In this problem we explore the brightness of Venus as a function of a polar angle $\theta$ that defines its position. At what value of $\theta$ is Venus the brightest?

We start by reasoning about the path of a light ray as it bounces off the surface of Venus and proceeds on its way to the Earth observer. A very important angle that lurks behind the geometry is the elongation angle $\phi$:

$$\phi = \arctan \left( \frac{\sin(\theta) d_V}{(d_E - \cos(\theta) d_V)} \right)$$

Here, $d_E$ and $d_V$ are the radii of the Earth and Venus orbits, which we assume are circular. The distance between the Earth and Venus is given by

$$d_{EV} = \frac{d_E - d_V \cos(\theta)}{\cos(\phi)}.$$  

The vectors

$$S_{move} = \begin{bmatrix} -d_E \sin(\phi) \\ -d_{EV} + d_E \cos(\phi) \\ 0 \end{bmatrix}$$  

$$E_{move} = \begin{bmatrix} 0 \\ -d_{EV} \\ 0 \end{bmatrix}$$

specify the location of the Sun and Earth in the moving coordinate system. A point $(x, z)$ in the crescent corresponds to the Venus surface point

$$\sigma = \begin{bmatrix} x \\ -\sqrt{R^2 - x^2 - z^2} \\ z \end{bmatrix}$$

(Think of the right window as a projection of Venus onto the xz-plane with the y axis pointing into the screen.)

The outward unit normal $g$ at this point is given by $g = \sigma / R$.

Let $u$ be a unit vector in the direction of $S_{move} - \sigma$. In our model of reflection the cosine of the angle between $u$ and $g$ is important. If a ray has unit strength before it strikes the surface at $\sigma$, then after it bounces off Venus it has strength $|u^T g|$, which is the cosine of the angle between $u$ and $g$. After the reflection the ray heads towards the Earth observer, who because of the inverse square law, detects this level of intensity:

$$\text{PointEnergy}(x, z) = \frac{|u^T g|}{d_{EV}^2}$$

To quantify the overall brightness of the Venus crescent we integrate $\text{PointEnergy}(x, z)$ over the crescent, i.e.,

$$\text{CrescentEnergy}(\theta) = \int_{-R}^{R} \text{LineEnergy}(z) dz$$

where

$$\text{LineEnergy}(z) = \int_{Left(z)}^{Right(z)} \text{PointEnergy}(x, z) dx$$

Regarding the limits of integration for the inner integral, if $\theta < \pi$, then $(Left(z), z)$ is on the left edge of the disk and $(Right(z), z)$ is on the terminator. If $\theta \geq \pi$, then $(Left(z), z)$ is on the terminator and $(Right(z), z)$ is on the right edge.

Use QUAD with default tolerances to compute these integrals. You have to pass the parameters $\theta, d_V, d_E, R$ so you’ll have to encapsulate the above ideas with functions

function tau = PointEnergy(x,z,theta,dV,dE,R)  
function rho = LineEnergy(z,theta,dV,dE,R)  
function E = CrescentEnergy(theta,dV,dE,R)

Note that QUAD expects the integrand function to be vectorized. That is, if you pass a vector of values to the integrand function then it should return the corresponding vector of function evaluations.

Submit each of these three functions. We’ll confirm that they work by running the script A3B which is available on the website. It actually looks at all inner planet/ outer planet pairs that involve Mercury, Venus, Earth, and Mars. Note: A3B uses maxSpline from Part A.