CS 322: Assignment P5

Due: Wednesday, May 1, 2002 in lecture or Friday, May 3, in Upson 4130 by 4pm.

You may work in pairs. Follow the course rules for the submission of assignments. 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.

Part A (10 pts) Season Lengths and the Eccentric Model

Suppose the Sun moves counterclockwise along the circle \( x^2 + y^2 = 1 \) at a uniform angular rate. Assume that the Earth is located at \((0,0)\) and has a 23.5\(^\circ\) axis tilt with the tilt being towards the summer solstice point \(S\) in the following figure:

The points \(A\), \(W\), and \(V\) are respectively the autumnal equinox point, the winter solstice point, and the vernal equinox point. It follows that spring, summer, fall, and winter begin when the Sun passes through \(V\), \(S\), \(A\), and \(W\). As reported in the figure, we have a problem because the model predicts that each season has length \(365.2425/4 = 91\) days + 7 hours + 21 minutes when in fact they have rather different durations:

<table>
<thead>
<tr>
<th>Season</th>
<th>Start</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spring</td>
<td>19:02 March 20, 2002</td>
<td>92d 18h 08m</td>
</tr>
<tr>
<td>Summer</td>
<td>13:07 June 21, 2002</td>
<td>93d 15h 37m</td>
</tr>
<tr>
<td>Autumn</td>
<td>04:29 September 23, 2002</td>
<td>88d 20h 13m</td>
</tr>
<tr>
<td>Winter</td>
<td>01:01 December 22, 2002</td>
<td>88d 23h 51m</td>
</tr>
<tr>
<td>Spring</td>
<td>00:31 March 21, 2003</td>
<td>- - - - - - - - -</td>
</tr>
</tbody>
</table>

This will be the case no matter how we “tilt” the equinox/solstice “crosshair”.

In order to improve the predictive power of the model we continue to move the Sun uniformly along the circle. However, we now position the Earth at the point \((d, 0)\). Assume that \(V = (\cos(\alpha), \sin(\alpha))\). With this
2-parameter model we can vary the lengths of the predicted seasons. For example, by setting \( d = .2 \) and \( \alpha = \pi/6 \) we obtain

\[
\phi(d, \alpha) = \max\{ |T_V - \bar{T}_V|, |T_S - \bar{T}_S|, |T_A - \bar{T}_A|, |T_W - \bar{T}_W| \}
\]

i.e., the maximum discrepancy between the actual season lengths \( T_V, T_S, T_A, \) and \( T_W \) and their predicted counterparts \( \bar{T}_V, \bar{T}_S, \bar{T}_A, \) and \( \bar{T}_W \).

Our approach will be to use the MATLAB one-dimensional minimizer \( \text{fmin} \) alternately applied to the functions

\[ f_\alpha(d) = \phi(d, \alpha) \quad \alpha \text{ fixed} \]

and

\[ f_d(\alpha) = \phi(d, \alpha) \quad d \text{ fixed} \]

In particular, write a script P5A that initializes both \( d_0 \) and \( \alpha_0 \) to zero and then repeats the following for \( i = 1:6 \):

- Use \( \text{fmin} \) to compute \( d_1 \) so that \( f_{\alpha_{i-1}}(d_1) \) is the minimum value of \( f_{\alpha_{i-1}} \) across the interval \([-1, 1]\).
- Display the results using \( \text{DrawEccentric} \). (See below.)
- Use \( \text{fmin} \) to compute \( \alpha_1 \) so that \( f_d(\alpha_1) \) is the minimum value of \( f_d \) across the interval \([0, 2\pi]\).
- Display the results using \( \text{DrawEccentric} \). (See below.)

A function

\[
\text{function } [x, y] = \text{CirclePoints}(d, \alpha) \\
% \text{Determines the equinox and solstice points for the eccentric model} \\
% \text{with displacement } d \text{ and rotation } \alpha. \\
% \\
% x \text{ and } y \text{ are row 4-vectors with the property that} \\
% \\
% (x(1), y(1)) = \text{coordinate of the Vernal Equinox point} \\
% (x(2), y(2)) = \text{coordinate of the Summer Solstice point} \\
% (x(3), y(3)) = \text{coordinate of the Autumnal Equinox point} \\
% (x(4), y(4)) = \text{coordinate of the Winter Solstice point} 
\]
is provided on the website. You may find it handy to write a function \( sE = \text{SeasonLength}(d, \alpha) \) that returns a 4-vector of season lengths. This makes it easy write \( f_a \) and \( f_d \). Use

\[
T_V = 92.7556 \quad T_S = 93.6507 \quad T_A = 89.8424 \quad T_W = 88.9938
\]

and a total year length \( Y = 365.2425 \). Note that if \( VS \) is the arclength from \( V \) to \( S \) then \( \tilde{T}_V = Y - VS/(2\pi) \). To compute the arc length you might want to work with the arcsec function asin and the isosceles triangle defined by \((0,0), V, A\). Obtain similar expressions for the other predicted season lengths.

Your script P5A should call

```matlab
function DrawEccentric(d, alpha, sE, sA)
% Displays the eccentric model with displacement d and rotation
% alpha.
% sE is a row 4-vector of estimated season lengths (in days).
% sA is a row 4-vector of actual season lengths (in days).
% Both sE and sA specify these lengths in Spring-Summer-Fall-Winter order.
```

after each \( d \) update and \( \alpha \) update. Submit the figures associated with \((d_1, \alpha_1), (d_2, \alpha_2), \) and \((d_6, \alpha_6)\). The process pretty much converges after six iterations. Also submit a listing of all scripts and functions that you had to write to produce these figures.

**Part B (10 pts) Apollo 13**

Consider the following IVP:

\[
\begin{align*}
\dot{x}(t) &= 2y(t) + x(t) - \frac{\mu(x(t) + \mu)}{r_1^3}, & x(0) = 1.2, & \dot{x}(0) = 0, \\
\dot{y}(t) &= -2x(t) + y(t) - \frac{\mu y(t)}{r_2^3}, & y(0) = 0, & \dot{y}(0) = -1.0493575,
\end{align*}
\]

where \( \mu = 1/82.45, \mu_+ = 1 - \mu, \) and

\[
\begin{align*}
r_1 &= \sqrt{(x(t) + \mu)^2 + y(t)^2} \\
r_2 &= \sqrt{(x(t) - \mu_+)^2 + y(t)^2}
\end{align*}
\]

It describes the orbit of a spacecraft that starts behind the Moon (located at \((1 - \mu, 0)\)), swings by the Earth (located at \((-\mu, 0)\)), does a large loop, and returns to the vicinity of the Earth before returning to its initial position behind the Moon at time \( T_0 = 6.19216933 \). Here, \( \mu = 1/82.45 \).

(a) Apply \texttt{ode45} with \( t_{\text{initial}} = 0, t_{\text{final}} = T_0, \) and \( tol = 10^{-6} \). Plot the orbit twice, once with the default "pen" and once with "", so that you can see how the time step varies. (b) Using the output from the \texttt{ode45} call in part (a), plot the distance of the spacecraft to Earth as a function of time across \([0, 75]\). Use spline to fit the distance "snapshots." To within a mile, how close does the spacecraft get to the Earth's surface? Assume that the Earth is a sphere of radius 4000 miles and that the Earth-Moon separation is 228,000 miles. Use \texttt{fmin} with an appropriate spline for the objective function. Note that the IVP is scaled so that one unit of distance is 228,000 miles. (c) To the nearest minute, compute how long the spacecraft is hidden to an observer on earth as it swings behind the Moon during its orbit. Assume that the observer is at \((-\mu, 0)\) and that the Moon has diameter 2160 miles. Make intelligent use of \texttt{fzero}.

Submit output and all the scripts and functions that were required to produce it.