CS 322: Assignment 5

Due: Friday, April 23, 2004 (5 pm)

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

`http://www.cs.cornell.edu/Courses/cs322/2004sp/`

Points will be deducted for poorly commented code, redundant computation that seriously effects 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 (4 pts) Fitting Conic Sections

The set of points in the plane that satisfy $Q(x,y) = ax^2+bxy+cy^2+dx+ey+1 = 0$ define a *conic section*. The circle, the parabola, the ellipse, and the hyperbola are examples. Given the points $(x_1,y_1),\ldots,(x_n,y_n)$, what's the "best" fitting conic section? One idea is to minimize the sum of squares $Q(x_1,y_1)^2+\cdots+Q(x_n,y_n)^2$:

```
  function [a,b,c,d,e] = ConicFit(xVals,yVals)
% xVals and yVals are column n-vectors, n>=5.
% a,b,c,d, and e are scalars chosen so that if
%
%         Q(x,y)= ax^2 + bxy + cy^2 + dx + ey + 1
%
% then the sum of squares
%
%         Q(xVals(1),yVals(1))^2 + ... + Q(xVals(n),yVals(n))^2
%
% is minimized.
```

You are allowed to use the "\" operator to solve least squares problems. We will test your implementation with the script `A5A` which is available on the website.

Problem B (8 pts) Solving Rational Equations

Suppose $d, v \in \mathbb{R}^{n-1}$ and $\alpha \in \mathbb{R}$ are given. Assume that $d_1 > d_2 > \cdots > d_{n-1} > 0$ and that $v_i \neq 0$, $i = 1{:}n-1$. The function

$$f(\lambda) = \lambda + \sum_{i=1}^{n-1} \frac{v_i^2}{d_i - \lambda} \; - \; \alpha$$

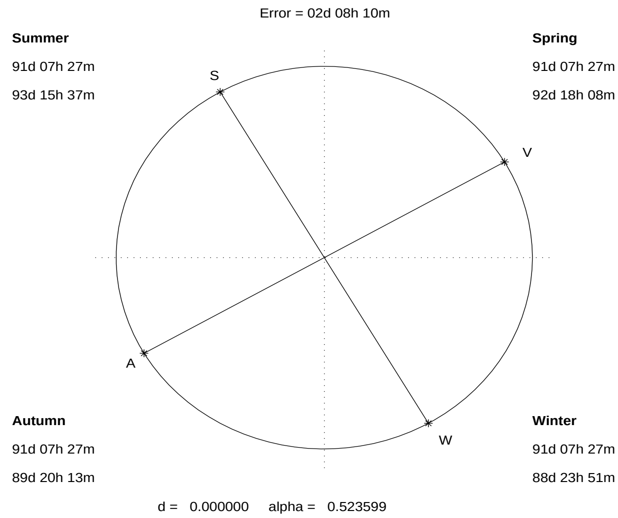
has n distinct roots. Complete the following function so that it performs as specified

```
  function lambda = RationalSolver(d,v,alpha,tol)
% d and v are column (n-1)-vector with d(1) > d(2) > ... > d(n-1), and
% assume each v(i) is nonzero.
% alpha is a scalar and tol is a positive scalar.
% Define f(z) = z + v(1)^{2}/(d(1)-z) + ... + v(n-1)^2/(d(n-1)-z) - alpha
% lambda is a column n-vector whose components satisfy lambda(1) > ... > lambda(n)
% and are roots of f with absolute error less than or equal to tol.
```

Use `fzero` to find these roots. In particular, to compute λ_i, find a bracketing interval $[u_i, v_i]$ that contains λ_i (and only that root) and then execute `lambda(i) = fzero('MyF',[u(i),v(i)],optimset('TolX',tol))`. We are not supplying a test script–so make sure that you try out your code on a wide range of examples. In particular, make sure your bracketing intervals are reliable.

Part C (8 pts) Season Lengths and the Eccentric Model

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

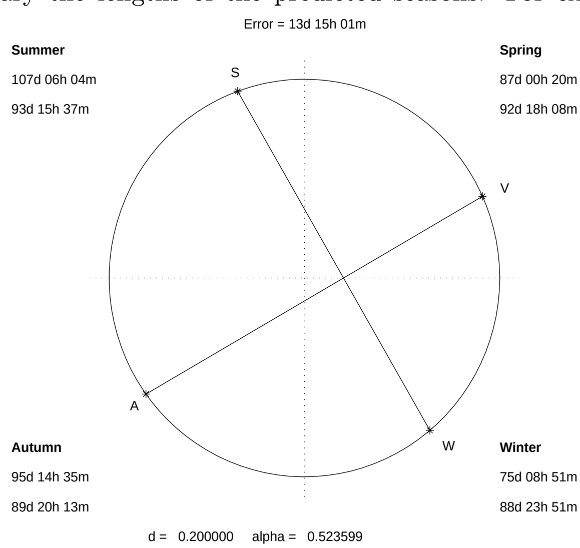


The points A, W, and V are respectively the autumnal equinox point, the winter solstice point, and the vernal equinox point. It follows that spring, summer, fall, and winter begin when the Sun passes through V, S, A, and W. As reported in the figure, we have a problem because the model predicts that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

Season	Start	Length
Spring	19:02 March 20, 2002	92d 18h 08m
Summer	13:07 June 21, 2002	93d 15h 37m
Autumn	04:29 September 23, 2002	89d 20h 13m
Winter	01:01 December 22, 2002	88d 23h 51m
Spring	00:31 March 21, 2003	- - - - - -

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



By moving the Earth to (.2,0) we have increased the predicted length of summer (good) and autumn (not good) and decreased the predicted length of winter (good) and spring (not good). The question before us is how to best choose d and α so as to minimize

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

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

Our approach will be to use the MATLAB one-dimensional minimizer `fmin` alternately applied to the functions

$$f_\alpha(d) = \phi(d,\alpha) \qquad \alpha \text{ fixed}$$

and

$$f_d(\alpha) = \phi(d,\alpha) \qquad d \text{ fixed}$$

Write a function `[dnext,alphanext] = Update(d,alpha)` that

- Uses `fmin` to compute d_{next} so that $f_\alpha(d_{next})$ is the minimum value of $f_\alpha(d)$ as d ranges across the interval $[-1,1]$.
- Uses `fmin` to compute α_{next} so that $f_{d_{next}}(\alpha_{next})$ is the minimum value of $f_{d_{next}}(\alpha)$ as α ranges across the interval $[0,2\pi]$.

A function

```
  function [x,y] = CirclePoints(d,alpha)
% Determines the equinox and solstice points for the eccentric model
% with displacement d and rotation alpha.
%
% x and y are row 4-vectors with the property that
%
%  (x(1),y(1)) = coordinate of the Vernal   Equinox  point
%  (x(2),y(2)) = coordinate of the Summer   Solstice point
%  (x(3),y(3)) = coordinate of the Autumnal Equinox  point
%  (x(4),y(4)) = coordinate of the Winter   Solstice point
```

encapsulates a lot of the underlying trigonometry and is provided on the website. You may find it handy to write a function `sE = SeasonLength(d,alpha)` that returns a 4-vector of season lengths. This makes it easy write f_α and f_d. Use

$$T_V = 92.7556 \qquad T_S = 93.6507 \qquad T_A = 89.8424 \qquad T_W = 88.9938$$

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