CS4220 Assignment 4 Due: 3/10/14 (Mon) at 11pm

You must work either on your own or with one partner. You may discuss background issues and general solution strategies with others, but the solutions you submit must be the work of just you (and your partner). If you work with a partner, you and your partner must first register as a group in CMS and then submit your work as a group. Each problem is worth 5 points. One point may be deducted for poor style.

Topics: Finding zeros of a single nonlinear equation. `fzero`. Minimizing a function of a single variable. `fminbnd`.

1 Roots of $f(x) = \sin(\alpha x) - x$

Complete the following function so that it performs as specified:

```
  function xRoots = FindZeros(alpha)
% alpha is a positive real number.
% xRoots is a column that houses all the roots (in left-to-right order) of the function
%
%                             f(x) sin(alpha*x) - x
%
% that occur in the interval [-1,1].
```

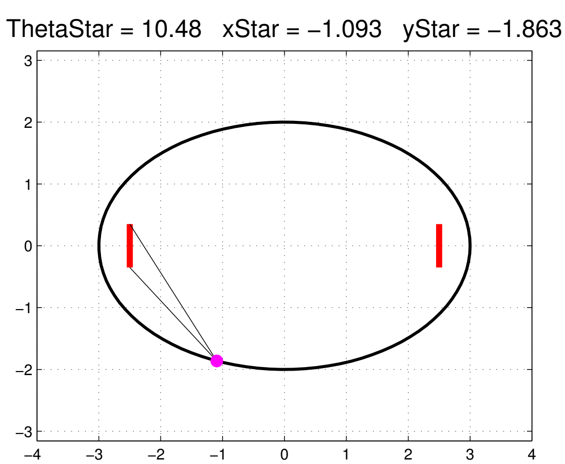
Make effective use of `fzero` and use default tolerances. The script `P1` plots f for various α and it should get your intuition rolling! Submit `FindZeros` to CMS.

2 Maximum Angle

Elliptical soccer is played on a pitch whose boundary is given by

$$\begin{aligned} x(t) &= a\cos(t) \\ y(t) &= b\sin(t) \end{aligned}$$

where $0 \le t \le 2\pi$. The goals have width w and are centered at $(-d, 0)$ and $(d, 0)$ respectively. Goal A has "endpoints $P_1 = (-d, -w/2)$ and $P_2 = (-d, w/2)$. Assume that these goal endpoints are within the boundary of the pitch. Where on the boundary does Goal A "look the biggest"?



To answer this question, complete the following function so that it performs as specified:

```
   function  [xStar,yStar,ThetaStar] = Sideline(a,b,d,w)
% a, b, c, d are positive real numbers with  the property that goalpost locations
% P1 = (-d,-w/2) and P2 = (-d,w/2) are inside the ellipse (x/a)^2 + (y/b)^2 = 1.
% Q = (xStar,yStar) is a point on the ellipse  with the property  that
% angle P1-Q-P2 is maximized subject to the constraint that a player standing at Q
% is looking into the goal. ThetaStar is the maximum angle in degrees.
```

Make effective use of `fminbnd` with default tolerances. You may use any built-in trig function. Do not worry about the nonuniqueness of the optimum sideline point. Submit `Sideline` to CMS.

3 Transit Times

A transit of Venus occurs when that planet is "in between" the Earth and the Sun. During the transit, Venus appears as a small dot on the face of the Sun. This problem is about transits of Mercury (as viewed from Venus, Earth and Mars), transits of Venus (as viewed from Earth and Mars) and transits of Earth (as viewed from Mars. We assume throughout that the orbits of these 4 planets are co-planar and that the planets themselves are points.

Run the simulation `ShowTransits` that can be downloaded from the website. It helps us reason about transits as viewed from Mars. Notice the left and right tangent lines that connect Mars to the left and right "edges" of the Sun. A transit is "going on" when one of the inner three planets is in between these two lines and is in between Mars and the Sun (which is centered at (0,0).) For example, when Earth crosses the left tangent line (cyan) the transit starts and when it crosses the right tangent line (magenta) it ends.

Take a look at `ShowTransit`. Note that the subfunction `[L,R] = TP(E,r)` is used to compute the tangent points. In the simulation, `r = 15` so that we can see how the tangent lines "work." The actual radius of the Sun is `r = .6955`. (All distances in this problem are in millions of kilometers.) Also note that the subfunction `E = Location(t,k)` is used to compute the location of planet `k` at time `t`. Time is in days, $k = 1$ corresponds to Mercury, $k = 2$ corresponds to Venus, $k = 3$ corresponds to Earth, and $k = 4$ corresponds to Mars.

Implement a function `[StartTime,Duration] = Transits(i,j,T)` that reports on the transits of planet i as seen from planet j. Here, $1 \leq i < j \leq 4$. Regarding the output, if m transits start during the time interval $[0, T]$, then `StartTime` and `Duration` are column m-vectors where the p-th transit starts at StartTime(p) (measured in days) and has duration `Duration(p)` (measured in minutes). It is OK for a transit to end after $t = T$. If no transits start during $[0, T]$, then empty vectors should be returned in `StartTime` and `Durartion`.

Transit start times occur when the vector from the outer to the inner planet makes an angle of zero degrees with the vector from the outer planet to the left tangent point *and* the inner planet is between the Sun and the Outer planet. Likewise, transit end times occur when the vector from the outer to the inner planet makes an angle of zero degrees with the vector from the outer planet to the right tangent point. The sines of these angles are functions of t and they should be zeroed using `fzero` (with default tolerances). For your information, if `x` and `y` are nonzero 2-vectors, then `(x(1)*y(2)-x(2)*y(1))/(norm(x)*norm(y))` is the sine of the angle between them.

You'll have to develop a method for computing the initial bracketing intervals. You may use the fact that all transits viewed from Venus, Earth, and Mars are less than 12 hours in length. To fix the time domain of interest, you may also assume that $T <= 5000$. Be careful about "untransits", the situation when the inner planet is in between the tangent lines but behind the Sun.

Submit `Transits.m` to CMS. (No need to include `TP` or `Location` as a subfunctions.)