CS 422: Assignment 5

Due: Monday, April 7, 2008

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. Test drivers and related material are posted on the course website `http://www.cs.cornell.edu/courses/cs422/2008sp/`. For P1 and P2 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. (Newton Applied to Complex f(z))

Newton's method is also defined for complex-valued functions. The same recipe holds:

$$z_+ = z_c - \frac{f(z_c)}{f'(z_c)}$$

Here, z_c and z_+ are complex numbers. This problem is about the behavior of Newton's method if we use it to find a zero of $f(z) = z^4 - 1$. There are four roots: $r_1 = 1$, $r_2 = i$, $r_3 = -1$ and $r_4 = -i$. Find a positive real number ρ such that if a Newton iterate z_c satisfies $|r_i - z_c| \leq \rho$, then the iteration converges to root r_i. Using your chosen value of ρ, complete the following function so that it performs as specified:

```
  function [zVals,idx] = TraceNewton(z0,nMax)
% zVals is a column n-vector where zVals(k) is the kth iterate when Newton's
% method is used to find a zero of f(z) = z^4 - 1 given that z0 is the starting
% value.
% idx is the index of the root to which the iteration is converging (1,2,3, or 4).
% n should be the smallest integer such that |zVals(n) - r_{idx}| <= rho.
% If more than nMax iterations are required, then n = nMax and idx = 0
```

In other words, we want to run Newton's method just long enough to discover which of the four roots it is converging to. Clearly, the bigger the value of your ρ, the more efficient will be your implementation.

P2. (Long Winter)

In laying water mains, utilities must be concerned with the possibility of freezing. Although soil and weather conditions are complicated, reasonable approximations can be made on the basis of the assumption that soil is uniform in all directions. In that case the temperature in degrees Celsius $T(x,t)$ at distance x (meters) below the surface, t seconds after the beginning of a cold snap, is given approximately by

$$\frac{T(x,t) - T_s}{T_i - T_s} = erf\left(\frac{x}{2\sqrt{\alpha t}}\right)$$

where T_s is the constant surface temperature during the cold period, T_i is the initial surface temperature before the cold snap, and α is the thermal conductivity of the soil (in $meters^2$ per second). For your information

$$erf(w) = \frac{2}{\sqrt{\pi}} \int_0^w \exp(-t^2)dt$$

It is a built-in MATLAB function.

Assume that $T_i = 20^o$C, $T_s = -15^o$C, and $\alpha = 0.138 \cdot 10^{-6} m^2/sec$. Use `fzero` to determine how deep a water main should be buried so that it will only freeze after 60 days exposure at this constant surface temperature.

P3. (Multiple Roots)

Complete the following function so that it performs as specified:

```
  function xRoots = TanLine(L,R)
% 0 < L < R
% xRoots is a column vector consisting of all roots to the function
%                  f(x) = tan(x) - x
% in the interval [L,R]. xRoots should be the empty vector if there are no roots.
```

Make effective use of `fZero`.

P4. (Point-to-Ellipse Distance)

Complete the following function so that it performs as specified:

```
  function [d,tStar] = minDist(u,v,a,b,tol)
% Assume a> 0 and b> 0 and let E be the ellipse defined by
%              (x(t),y(t)) = (a*cos(2*pi*t),b*sin(2*pi*t))
% d is the minimum distance between (u,v) and a point on E and
% (x(tStar),y(tStar)) is a nearest point.
% tol > eps and tStar should be within tol of an exact minimizing t.
```

Your implementation should make use of Golden Section Search.

P5. (Best Inbound Shot Location)

A soccer pitch has dimensions 100-by-50 (meters) with a 10-meter goals centered on the end lines. At what point on the sideline does the left-to-right width of the goal appear widest? Make effective use of `fminbnd`.