Coding Challenge 2 - Part 1

Zach Swain, 4/2/18, All files available at https://www.github.com/zswain/MEEG332

```
clear all

initGuess = .5;           %let initial guess be .5 as in c

n = 0:.1:10;              %same n range and step
y0 = [0 0 initGuess];     %same y0 definition as in c
[nSol,ySol] = ode45(@(n,y) lamBoundLayerVeloODE(n,y),n,y0); %evaluate
 the ODE as in c
y2 = ySol(:,2);           %define y2 as all rows in column 2 of ySol,
 f'

diff = y2(100)-1;         %evaluate the difference between y2@infinity
 and 1
if diff > 0               %if initial guess gives y2@infinity greater
 than 1
    guess = initGuess;    %for cohesion in while
    while diff > 0        %until y2@infinity gets just barely below 1
        y0 = [0 0 guess]; %redefine y0 with current guess value
        [nSol,ySol] = ode45(@(n,y)
 lamBoundLayerVeloODE(n,y),n,y0); %update evaluation of ODE
        y2 = ySol(:,2);   %update y2 def as current y2 column 2
        diff = y2(100)-1; %update new difference from 1
        if diff > .002    %originally had just guess-=.0000001 but
 runtimes were absurd
            guess = guess-.001; %blunt if statements are a poor man's
 optimization
        end
        if diff <= .002 && diff > .00014
            guess = guess-.0001; %more precise guess steps
        end
        if diff <= .00014 && diff > .000035
            guess = guess-.00001; %more precise guess steps
        end
        if diff <= .000035 && diff > .00001
            guess = guess-.000001; %more precise guess steps
        end
        if diff <= .00001
            guess = guess-.0000001; %actual guess step precision
 wanted
        end
    end
    alpha = guess         %define alpha as latest guess value
end

y0 = [0 0 initGuess];     %redefine y0 to have original guess not
 latest guess value from while
[nSol,ySol] = ode45(@(n,y) lamBoundLayerVeloODE(n,y),n,y0); %redefine
 using original guess
```

```
y2 = ySol(:,2);           %redefine using original guess
                          %redefs to ensure cant go through both
 whiles
diff = y2(100)-1;         %redefine diff to original
if diff < 0               %if inital guess gives y2 less than 1
    guess = initGuess;    %for cohesion in while
    while diff < 0        %until y2@infinity gets just barely above 1
        y0 = [0 0 guess]; %redefine y0 with current guess value
        [nSol,ySol] = ode45(@(n,y)
 lamBoundLayerVeloODE(n,y),n,y0); %update evaluation of ODE
        y2 = ySol(:,2);   %update y2 def as current y2 column 2
        diff = y2(100)-1; %update new difference from 1
        if diff < -.002   %ifs to avoid long runtimes as previous
 while
            guess = guess+.001; %makeshift optimizaiton as previous
 while
        end
        if diff >= -.002 && diff < -.00014
            guess = guess+.0001; %more precise guess steps
        end
        if diff >= -.00014 && diff < -.000035
            guess = guess+.00001; %more precise guess steps
        end
        if diff >= -.000035 && diff < -.00001
            guess = guess+.000001; %more precise guess steps
        end
        if diff >= -.00001
            guess = guess+.0000001; %more precise guess steps
        end
    end
    alpha = guess         %define alpha as latest guess value
end
```

```
alpha =

    0.3320
```

Published with MATLAB® R2017b