Homework Project #2

Math 365, Fall 2017

1. **Fixed point method.** Use the fixed-point method to solve

$$xe^x = c$$

for $c = 1$ and for $c = 3$. Does the fixed point method converge for both values of c?

Hint : Solve the problem $x = ce^{-x}$ and use a starting guess of $x_0 = 1$. For each value of c, report your results of the first 15 iterations in a table. See Figure 1.

2. **Newton's Method.** Use Newton's Method to solve the equation

$$xe^x = c$$

for values $c = 1$ and $c = 3$. For each value of c, report your results in a table and show convergence to a tolerance of $\tau = 1 \times 10^{-10}$.

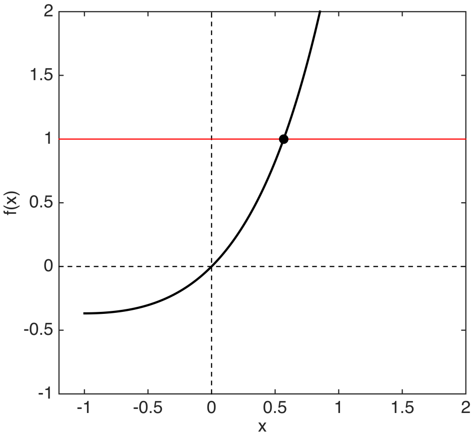

Figure 1: Plot of function $f(x) = xe^x$ from Problem 1 and Problem 2. Also shown is the line $y = 1$. The solid dot shows the solution of the equation $xe^x = 1$.

3. **Inverse of $f(x) = xe^x$.** For this problem, you will write a function $y = L(x)$, where y is *implicitly* defined by the equation $ye^y = x$. The idea is to use Newton's Method to solve $ye^y = x$ for y, given input value x.

Once you have written your function, do the following :

(a) *Vectorize* your function. This means that you should write it so that it takes with an array x of values. The easiest way to do this is to just write an outer loop that loops over each value x_i in the input array.

(b) Create a plot of the function $L(x)$ over the interval $x \in [-1/e, e]$. On the same graph, plot your function $f(x) = xe^x$ for values of $x \in [-1, 1]$. Add the reference line $y = x$, and lines indicating the x and y axis. Add labels, a title and a legend to your plot. See Figure 2 for an example of what your plot should look like.

You should see that the functions $f(x)$ and $L(x)$ are reflections of each other about the line $y = x$. In other words, these functions are *inverses* of each other.

Here is a sample of what code for your function $L(x)$ should look like :

```
function y = L(xvec)
     yvec = zeros(size(xvec);
     for k = 1:length(x)
         x = xvec(k);
         % check that the input value 'x' is a valid input

         % Use Newton's Method to solve y*exp(y) = x to get y

         yvec(k) = y;
     end
end
```

Hint : Test your function $L(x)$ by checking that you get $L(f(3)) = 3$, where $f(x) = xe^x$.

The function $L(x)$ you created is called the *Lambert W* function and in fact is the inverse of the function $f(x) = xe^x$. In MATLAB, this function is implemented as `lambertw`.

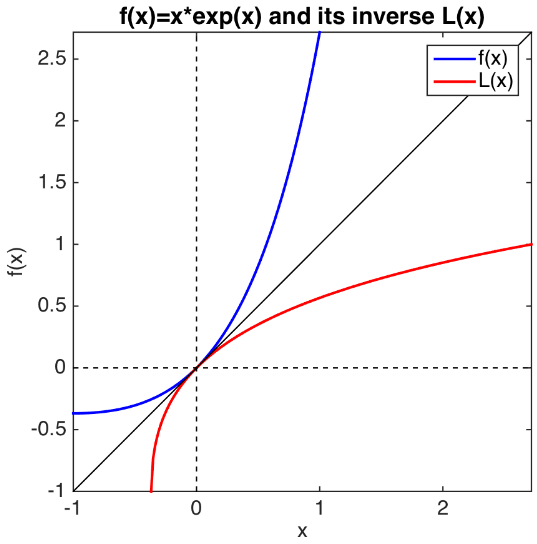

Figure 2: Plot of function $f(x) = xe^x$ from Problem 3 and its inverse, $L(x)$, the *Lambert W* function.

4. **Air Resistance.** The equation

$$h(t) = -33t + 784(1 - e^{-0.3t})$$

models the height of a rocket in the air at time t, subject to air resistance. See Figure 3 for a plot of the height $h(t)$ of the rocket as a function of time.

(a) Find the time at which the rocket hits its maximum height and compute the maximum height. Report your results using `fprintf`, and descriptive text.

(b) Find the time at which the rocket hits the ground, i.e. when $h(t) = 0$. Report your results using `fprintf`, and descriptive text.

(c) Produce a plot like the one in Figure 3, with your maximum height and the time the rocket hits the ground clearly marked.

Hints : You will be able to find the maximum height using just what you know from Calculus. To find the time that the rocket hits the ground, you will need to solve an expression like

$$At + e^{Bt} = 1 \tag{1}$$

for t. To do this, use the transformation

$$t = -\frac{1}{B}u + \frac{1}{A}$$

Substitute this into your expression (1) and solve for u, using your function $L(x)$. Then use the transformation to solve for t, the time at which the rocket hits the ground.

You might ask yourself why you would use $L(x)$ to solve this problem, rather than just solve (1) directly. After all, the transformation adds extra step of algebra that you might rather avoid. One reason for doing the algebra is that you don't know that (1) will successfully converge for all values of A and B. But you know that you can always evaluate $L(x)$, at least for values of $x > -1/e$. Moreover, you can optimize your $L(x)$, which may lead to a more robust algorithm than the one that is customized for (1) only.

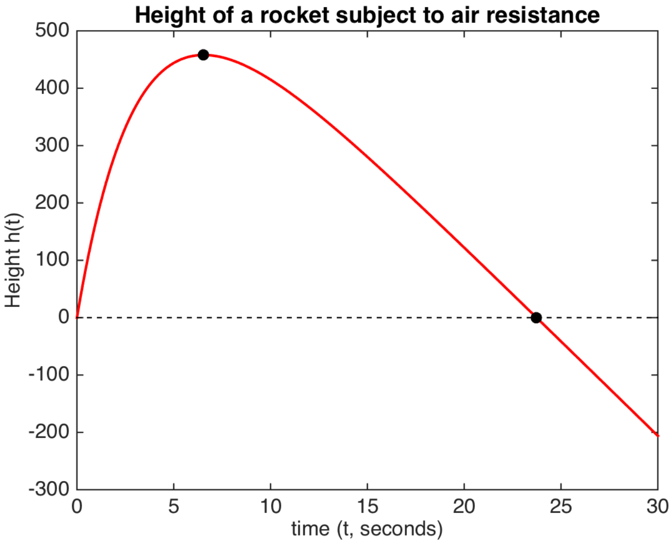

Figure 3: Height of a rocket in the air, subject to air resistance. The solid dots show the maximum height reached by the rocket, and the time at which the rocket hits the ground.

5. (**Newton's method**) The almost universally used algorithm to compute $\sqrt{a}$, where $a > 0$, is the recursion

$$x_{n+1} = \frac{1}{2}\left(x_n + \frac{a}{x_n}\right) , \tag{2}$$

easily obtained by means of Newton's method for the function $f(x) = x^2 - a$. One potential problem with this method is that it requires a floating point division, which not all computer processors support, or which may too expensive for a particular application.

For these reasons, it is advantageous to devise a method for computing the square root that does not require any floating point divisions, (except division by 2, which can be easily done by shifting the binary representation one bit to the right), but only addition, subtraction and multiplication. The trick for doing this is to use Newton's method to compute $\frac{1}{\sqrt{a}}$, and then obtain $\sqrt{a}$ by multiplying by a.

Your goal is to write an algorithm for computing the square root using this trick above. Division by 2 is allowed, but no other floating point division. Your algorithm should work for any input value $a > 0$.

- Write your algorithm as a function so you can call it any starting value.

  ```
  function y = squareroot(x)
       % Your algorithm goes here
  end
  ```

  Set the tolerance for the relative error in your solution to $\tau = 10^{-12}$. The *relative error* can be computed as

  $$e_k = \frac{|x_k - x_{k+1}|}{x_k}.$$

  where x_k and x_{k+1} are successive iterates in your Newton iteration.

- Use your function to compute $\sqrt{35}$, $\sqrt{2.3 \times 10^{-6}}$, and $\sqrt{17^{15}}$. Write out your three solutions to a file `squareroot.out`.

To see where this sort of software assisted accelaration is used in gaming, see the course webpage for a link to the article : *Origin of Quake3's Fast InvSqrt().*

6. (**Using `fzero`**) Consider the Colebrook equation for the friction factor in a fully developed pipe flow

$$\frac{1}{\sqrt{f}} = -2\log_{10}\left(\frac{\epsilon/D}{3.7} + \frac{2.51}{\mathrm{Re}_D\sqrt{f}}\right)$$

where f is the Darcy friction factor, ϵ/D is the relative roughness of the pipe material, and Re is the Reynolds number based on a pipe diameter d. use `fzero` to find the friction factor f corresponding to parameter values $\epsilon/D = 0.0001$ and $\mathrm{Re}_D = 3 \times 10^5$. Use a tolerance of 1×10^{-8} and write the root you find to the file `frictionfactor.out`. **hint :** use the function `optimset` to set up the tolerance `tolx`.

7. (**Using `fzero`**) David Peters (SIAM Review, 1997) obtains the following equation for the optimum damping ratio of a spring-mass-damper system designed to minimize the transmitted force when an impact is applied to the mass:

$$\cos\left[4\zeta\sqrt{1-\zeta^2}\right] = -1 + 8\zeta^2 - 8\zeta^4$$

Use `fzero` to find the $\zeta \in [0, 0.5]$ that satisfies this equation. Use a tolerance of 1×10^{-8} and write the solution to the file `damping.out`.

8. (**Using `fminbnd`**) Consider the curve traced out by the set of parametric equations

$$x(\theta) = e^{\pi\cos(\theta)}\cos(\pi\sin(\theta))$$
$$y(\theta) = e^{\pi\cos(\theta)}\sin(\pi\sin(\theta))$$

over the domain $-\pi \le \theta \le \pi$. The curve is shown in Figure 4.

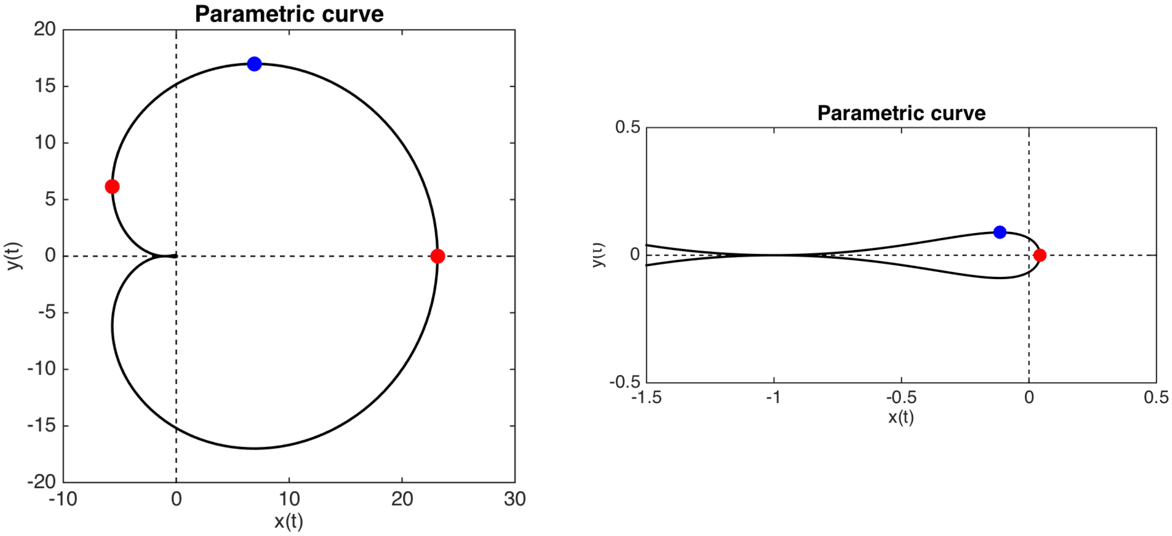

Figure 4: Parametric curve $(x(\theta), y(\theta))$ for Problem 8. The three red dots are the extreme values for the $x(\theta)$ and the two blue dots are the extreme values of the $y(\theta)$ coordinates. The figure at the right is a zoom of the left plot near the origin.

(a) Find the location of each of the five extrema shown as solid red or blue dots in the plots in Figure 4. Print out a table indicating the values of each θ value at the extrema, and corresponding $x(\theta)$ and $y(\theta)$ values. List the θ values in order of increasing θ and show at least 8 digits of accuracy in your table. Your θ values should be in the interval $[0, \pi]$.

(b) Plot the curve and each of the five extrema on a single graph.

For this problem, you will use the MATLAB function `fminbnd`, a function which finds the minimum value of a function in an interval $[a, b]$. For this problem, set the tolerance `TolX` to $tol = 1 \times 10^{-8}$.

Hint : To get an idea of where the extrema are, plot curves $x(\theta)$ and $y(\theta)$ as functions of θ. Use these curves to isolate the minimum and maximum values to an interval $[a, b]$ which you can then pass as an argument to the function `fminbnd`.

This problem comes from a branch of mathematics called *Complex Analysis*. The curve represents the graph, in the complex plane, of the function $f(z) = e^z$ for values of $z = \pi(\cos(\theta) + i\sin(\theta))$, where $i = \sqrt{-1}$.