Math 104A - Homework 2

Due 6/30

Section 1.3 - 1, 7a, 7d, 11
Section 2.1 - 6, 8, 20
Section 2.2 - 5, 8, 14
Section 2.3 - 5, 33

1.3.1a Use three-digit chopping arithmetic to compute the sum $\sum_{i=1}^{10} \frac{1}{i^2}$ first by $\frac{1}{1} + \frac{1}{4} + \cdots + \frac{1}{100}$, and then by $\frac{1}{100} + \frac{1}{81} + \cdots + \frac{1}{1}$. Which method is more accurate, and why?

Using three-digit chopping arithmetic,

$$\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \cdots + \frac{1}{100} = (\cdots((1.00 + 0.250) + 0.111) + \cdots) + 0.0100) = 1.53,$$

while

$$\frac{1}{100} + \frac{1}{81} + \frac{1}{64} + \cdots + \frac{1}{1} = (\cdots((0.0100 + 0.0123) + 0.0156) + \cdots) + 1.00) = 1.54.$$

The actual value is 1.5498. The first sum is less accurate because the smaller numbers are added last, resulting in significant round-off error.

1.3.1b Write an algorithm (pseudocode) to sum the finite series $\sum_{i=1}^{N} x_i$ in reverse order. (Here, the input is $N, x_1, \ldots, x_N$, and the output is the sum).

INPUT: $N, x_1, \ldots, x_N$
OUTPUT: $\sum_{i=1}^{N} x_i$

Step 1: Set $i = N, s = 0$.
Step 2: While $i > 0$ do Steps 3-4
  Step 3: Set $s = s + 1/i^2$.
  Step 4: Set $i = i - 1$.
Step 5: OUTPUT s.

1.3.7a Find the rate of convergence of $\lim_{h\to 0} \frac{\sin h}{h} = 1$ (Hint: use Taylor series).

$$\left| \frac{\sin(h)}{h} - 1 \right| = \left| \frac{h - \frac{h^3}{6}\cos(\xi)}{h} - 1 \right| = \left| -\frac{h^2}{6}\sin(\xi) \right| = O(h^2).$$

1.3.7b Find the rate of convergence of $\lim_{h\to 0} \frac{1-e^h}{h} = -1$.

$$\left| \frac{1 - e^h}{h} + 1 \right| = \left| \frac{1 - 1 - h - \frac{h^2}{2}e^{\xi}}{h} + 1 \right| = \left| -\frac{h}{2}e^{\xi} \right| = O(h).$$

1.3.11 Construct an algorithm (pseudocode) that has as input an integer $n \geq 1$, numbers $x_0, x_1, \ldots, x_n$, and a number x that produces as output the product $(x - x_0)(x - x_1) \cdots (x - x_n)$.

INPUT: $n, x_0, \ldots, x_n, x$.
OUTPUT: $\prod_{i=0}^{n}(x - x_i)$.

Step 1: Set $i = 0, p = 1$.
Step 2: While $i \leq n$ do Steps 3-4
  Step 3: Set $p = p * (x - x_i)$.
  Step 4: Set $i = i + 1$.
Step 5: OUTPUT p.

2.1.6 Use the Bisection method to find solutions accurate to within 10^{-5} for the following problems:

a $3x - e^x = 0, x \in [1, 2]$.

Using the attached code (`bisection_method.m`), we got

```
>> bisection_method('3*x-exp(x)',1,2,1000,10^-5)
ans =
   1.512138366699219
```

b $x + 3\cos x - e^x = 0, x \in [0, 1]$.

Using the attached code (`bisection_method.m`), we got

```
>> bisection_method('x+3*cos(x)-exp(x)',0,1,1000,10^-5)
ans =
   0.976768493652344
```

c $x^2 - 4x + 4 - \ln x = 0, x \in [1, 2]$ and $x \in [2, 4]$.

Using the attached code (`bisection_method.m`), we got

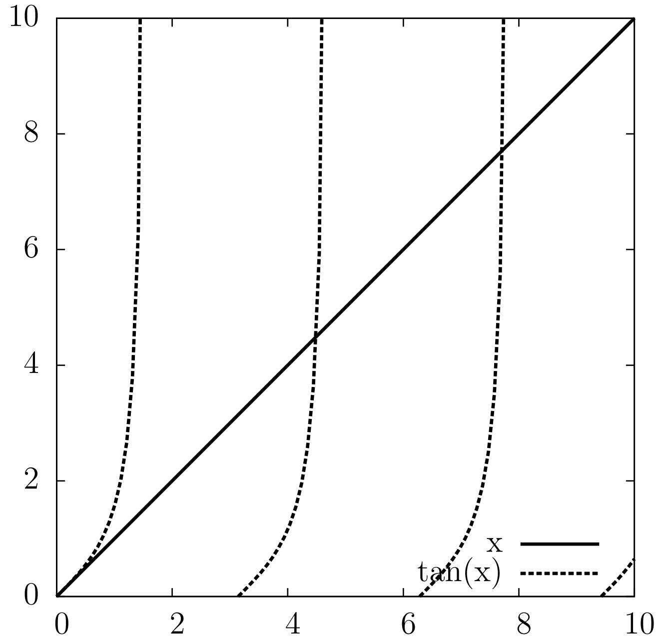
```
>> bisection_method('x^2-4*x+4-log(x)',1,2,1000,10^-5)
ans =
   1.412391662597656
>> bisection_method('x^2-4*x+4-log(x)',2,4,1000,10^-5)
ans =
   3.057106018066406
```

d $x + 1 - 2\sin(\pi x) = 0, x \in [0, 0.5]$ and $x \in [0.5, 1]$.

Using the attached code (`bisection_method.m`), we got

```
>> bisection_method('x+1-2*sin(pi*x)',0,0.5,1000,10^-5)
ans =
   0.206031799316406
>> bisection_method('x+1-2*sin(pi*x)',0.5,1,1000,10^-5)
ans =
   0.681968688964844
```

2.1.8a Sketch the graphs of $y = x$ and $y = \tan x$.



2.1.8b Use the bisection method to find an approximation to within 10^{-5} to the first positive value of x with $x = \tan x$.

From the sketch, we see that the first positive fixed-point occurs somewhere between 4 and 5. Using the attached code (`bisection_method.m`), we got

```
>> bisection_method('x-tan(x)',4,5,1000,10^-5)
ans =
   4.493415832519531
```

2.1.20 A particle starts at rest on a smooth inclined plane whose angle θ is changing at a constant rate $\frac{d\theta}{dt} = \omega < 0$. At the end of t seconds, the position of the object is given by

$$x(t) = -\frac{g}{2\omega^2}\left(\frac{e^{\omega t} - e^{-\omega t}}{2} - \sin(\omega t)\right).$$

Suppose the particle has moved 1.7 ft in 1 s. Find, to within 10^{-5}, the rate ω at which θ changes. Assume that $g = 32.17$ ft/s^2.

Substituting the appropriate values, we find ω by finding the root of

$$f(x) = -\frac{32.17}{2x^2}\left(\frac{e^x - e^{-x}}{2} - \sin(x)\right) - 1.7.$$

Using the attached code (`bisection_method.m`), we got

```
>> bisection_method('-32.17/(2*x^2)*((exp(x)-exp(-x))/2...
                       -sin(x))-1.7',-1,-0.1,1000,10^-5)
ans =
  -0.317055511474609
```

2.2.5 Use a fixed-point iteration method to determine a solution accurate to within 10^{-2} for $x^4 - 3x^2 - 3 = 0$ on $[1, 2]$. Use $p_0 = 1$.

After first rearranging the equation to get $(3x^2+3)^{1/4} = x$, we use attached code (`fixed_point_method.m`) to get

```
>> fixed_point_method('(3*x^2+3)^(1/4)',1,1000,10^-2)
Took 6 iterations.
ans =
   1.943316929898677
```

2.2.8 Use theorem 2.2 to show that $g(x) = 2^{-x}$ has a unique fixed point on $[\frac{1}{3}, 1]$. Use fixed-point iteration to find an approximation to the fixed point accurate to within 10^{-4}. Use corollary 2.4 to estimate the number of iterations required to achieve 10^{-4} accuracy, and compare this theoretical estimate to the number actually needed.

Since g is decreasing, we know that

$$\begin{aligned} \max g(x) &= g(\frac{1}{3}) \approx 0.793700526 < 1, \\ \min g(x) &= g(1) = 0.5 > \frac{1}{3}, \end{aligned}$$

and so $g(x) \in [\frac{1}{3}, 1]$. Since $g'(x) = -2^{-x}\ln(2)$ is negative and increasing, we know that $|g'(x)| \leq |g'(\frac{1}{3})| \approx 0.550151282 = k < 1$. Then g satisfies the conditions of theorem 2.2, and so g has a unique fixed point on the interval $[\frac{1}{3}, 1]$.

Using the attached code (`fixed_point_method.m`), we get

```
>> p = 0.6411857445049859844;

>> abs(p-fixed_point_method('2^(-x)',2/3,7))
ans =
     1.951777565947221e-04

>> abs(p-fixed_point_method('2^(-x)',2/3,8))
ans =
     8.673817179283283e-05
```

so 8 iterations are necessary to be within 10^{-4} of the true fixed point $p = 0.641185744504986$. By corollary 2.4, $|p_n - p| \leq \frac{k^n}{3}$, and so

$$\begin{aligned} \frac{k^n}{3} &\leq 10^{-4} \\ \ln(k^n) &\leq \ln(3 \cdot 10^{-4}) \\ n &\geq \frac{\ln(3 \cdot 10^{-4})}{\ln(k)} \approx 13.5747058. \end{aligned}$$

2.2.14 Use a fixed-point iteration method to determine a solution accurate to within 10^{-4} for $x = \tan x$, for $x \in [4, 5]$.

Rearranging the equation so that $g(x) = \frac{1}{\tan(x)} - \frac{1}{x} + x$ and using the attached code (`fixed_point_method.m`), we get

```
>> fixed_point_method('1/tan(x)-1/x+x',4.5,1000,10^-4)
Took 2 iterations.
ans =
   4.493409457929371
```

2.3.5 Use Newton's method to find solutions accurate to within 10^{-4} for the following problems:

a $x^3 - 2x^2 - 5 = 0, x \in [1, 4]$.

Using the attached code (`newtons_method.m`), we get

```
>> newtons_method('x^3-2*x^2-5','3*x^2-4*x',2.5,1000,10^-4)
Took 4 iterations
ans =
   2.690647448028615
```

b $x^3 + 3x^2 - 1 = 0, x \in [-3, -2]$.

Using the attached code (`newtons_method.m`), we get

```
>> newtons_method('x^3+3*x^2-1','3*x^2+6*x',-2.5,1000,10^-4)
Took 5 iterations
ans =
  -2.879385241571822
```

c $x - \cos x = 0, x \in [0, \pi/2]$.

Using the attached code (`newtons_method.m`), we get

```
>> newtons_method('x-cos(x)','1+sin(x)',pi/4,1000,10^-4)
Took 3 iterations
ans =
   0.739085133215161
```

d $x - 0.8 - 0.2 \sin x = 0, x \in [0, \pi/2]$.

Using the attached code (`newtons_method.m`), we get

```
>> newtons_method('x-0.8-0.2*sin(x)','1-0.2*cos(x)',pi/4,1000,10^-4)
Took 3 iterations
ans =
   0.964333887695271
```

2.3.33 Player A will shut out (win by a score of 21-0) player B in a game of raquetball with probability

$$P = \frac{1+p}{2}\left(\frac{p}{1-p+p^2}\right)^{21},$$

where p is the probability that A will win any specific rally (independent of the server). Determine, to within 10^{-3}, the minimal value of p that will ensure that A will shut out B in at least half the matches they play.

From a sketch of the graph, we see that p is close to 0.8, so we use the bisection method (`bisection_method.m`) with an initial interval $[0.7, 0.9]$ with function $f(x) = \frac{1+x}{2}\left(\frac{x}{1-x+x^2}\right)^{21} - 0.5$:

```
>> bisection_method('(1+x)/2*(x/(1-x+x^2))^21-0.5',...
                     0.7,0.9,1000,10^-4)
ans =
   0.842968750000000
```

Code

```
%%% bisection_method.m %%%

function p = bisection_method(fstring,a,b,N,TOL)

    i=1;
    f = inline(fstring);
    FA = f(a);

    while ( i <= N )

        p = (a+b)/2;
        FP = f(p);

        if( FP == 0 || (b-a)/2 < TOL )
            return;
        end

        if( FA*FP > 0 )
            a = p;
            FA = FP;
        else
            b = p;
        end

        i = i+1;

    end

end

%%% end of bisection_method.m %%%



%%% fixed_point_method.m %%%

function p = fixed_point_method(fstring,p0,N,TOL)

    i = 1;
    f = inline(fstring);

    while i < N

        p = f(p0);

        if( nargin == 4 && abs(p-p0) < TOL )
```

```
            fprintf('Took %i iterations.',i);
            return;
        end

        p0 = p;
        i = i+1;

    end

end

%%% end of fixed_point_method.m %%%



%%% newtons_method.m %%%

function p = newtons_method(fstring,fpstring,p0,N,TOL)

    i = 1;
    f = inline(fstring);
    fp = inline(fpstring);

    while(  i <= N )

        p = p0-f(p0)/fp(p0);

        if( abs(p-p0) < TOL )
            fprintf('Took %i iterations',i);
            return;
        end

        i = i+1;
        p0=p;

    end

end

%%% end newtons_method.m %%%
```