Math 5601 Homework 1

Jacob Hauck

September 10, 2023

Problem 1.

(a) See `bisect.m` – also copied here for convenience.

```
function result = bisect(f, a, b, epsilon, epsilon_f, max_it)
    for k = 0:max_it
        xk = (a + b) / 2;
        fk = f(xk);
        max_error = (b - a) / 2;

        fprintf( ...
            'k = %d, x_k = %.5g, max error = %.5g, f(x_k) = %.5g\n', ...
            k, xk, max_error, fk ...
        );

        if max_error < epsilon || abs(fk) < epsilon_f
            break;
        elseif f(a) * fk < 0  % root lies in [a, x]
            b = xk;
        else  % if root is not in [a, x], it must be in [x, b]
            a = xk;
        end
    end

    result = xk;
```

(b) **(1)** The following are snippets from the relevant parts of `outputs.txt`, which contains the output from the MATLAB console. For $\varepsilon = 10^{-2}$:

```
>> bisect(@(x) atan(x), -4.9, 5.1, 1e-2, 0, 50)
k = 0, x_k = 0.1, max error = 5, f(x_k) = 0.099669
k = 1, x_k = -2.4, max error = 2.5, f(x_k) = -1.176
k = 2, x_k = -1.15, max error = 1.25, f(x_k) = -0.85505
k = 3, x_k = -0.525, max error = 0.625, f(x_k) = -0.48345
k = 4, x_k = -0.2125, max error = 0.3125, f(x_k) = -0.20939
k = 5, x_k = -0.05625, max error = 0.15625, f(x_k) = -0.056191
k = 6, x_k = 0.021875, max error = 0.078125, f(x_k) = 0.021872
k = 7, x_k = -0.017188, max error = 0.039062, f(x_k) = -0.017186
k = 8, x_k = 0.0023437, max error = 0.019531, f(x_k) = 0.0023437
k = 9, x_k = -0.0074219, max error = 0.0097656, f(x_k) = -0.0074217

ans =

   -0.0074
```

For $\varepsilon = 10^{-4}$:

```
>> bisect(@(x) atan(x), -4.9, 5.1, 1e-4, 0, 50)
k = 0, x_k = 0.1, max error = 5, f(x_k) = 0.099669
k = 1, x_k = -2.4, max error = 2.5, f(x_k) = -1.176
k = 2, x_k = -1.15, max error = 1.25, f(x_k) = -0.85505
k = 3, x_k = -0.525, max error = 0.625, f(x_k) = -0.48345
k = 4, x_k = -0.2125, max error = 0.3125, f(x_k) = -0.20939
k = 5, x_k = -0.05625, max error = 0.15625, f(x_k) = -0.056191
k = 6, x_k = 0.021875, max error = 0.078125, f(x_k) = 0.021872
k = 7, x_k = -0.017188, max error = 0.039062, f(x_k) = -0.017186
k = 8, x_k = 0.0023437, max error = 0.019531, f(x_k) = 0.0023437
k = 9, x_k = -0.0074219, max error = 0.0097656, f(x_k) = -0.0074217
k = 10, x_k = -0.0025391, max error = 0.0048828, f(x_k) = -0.0025391
k = 11, x_k = -9.7656e-05, max error = 0.0024414, f(x_k) = -9.7656e-05
k = 12, x_k = 0.001123, max error = 0.0012207, f(x_k) = 0.001123
k = 13, x_k = 0.0005127, max error = 0.00061035, f(x_k) = 0.0005127
k = 14, x_k = 0.00020752, max error = 0.00030518, f(x_k) = 0.00020752
k = 15, x_k = 5.4932e-05, max error = 0.00015259, f(x_k) = 5.4932e-05
k = 16, x_k = -2.1362e-05, max error = 7.6294e-05, f(x_k) = -2.1362e-05

ans =

  -2.1362e-05
```

For $\varepsilon = 10^{-8}$:

```
>> bisect(@(x) atan(x), -4.9, 5.1, 1e-8, 0, 50)
k = 0, x_k = 0.1, max error = 5, f(x_k) = 0.099669
k = 1, x_k = -2.4, max error = 2.5, f(x_k) = -1.176
k = 2, x_k = -1.15, max error = 1.25, f(x_k) = -0.85505
k = 3, x_k = -0.525, max error = 0.625, f(x_k) = -0.48345
k = 4, x_k = -0.2125, max error = 0.3125, f(x_k) = -0.20939
k = 5, x_k = -0.05625, max error = 0.15625, f(x_k) = -0.056191
k = 6, x_k = 0.021875, max error = 0.078125, f(x_k) = 0.021872
k = 7, x_k = -0.017188, max error = 0.039062, f(x_k) = -0.017186
k = 8, x_k = 0.0023437, max error = 0.019531, f(x_k) = 0.0023437
k = 9, x_k = -0.0074219, max error = 0.0097656, f(x_k) = -0.0074217
k = 10, x_k = -0.0025391, max error = 0.0048828, f(x_k) = -0.0025391
k = 11, x_k = -9.7656e-05, max error = 0.0024414, f(x_k) = -9.7656e-05
k = 12, x_k = 0.001123, max error = 0.0012207, f(x_k) = 0.001123
k = 13, x_k = 0.0005127, max error = 0.00061035, f(x_k) = 0.0005127
k = 14, x_k = 0.00020752, max error = 0.00030518, f(x_k) = 0.00020752
k = 15, x_k = 5.4932e-05, max error = 0.00015259, f(x_k) = 5.4932e-05
k = 16, x_k = -2.1362e-05, max error = 7.6294e-05, f(x_k) = -2.1362e-05
k = 17, x_k = 1.6785e-05, max error = 3.8147e-05, f(x_k) = 1.6785e-05
k = 18, x_k = -2.2888e-06, max error = 1.9073e-05, f(x_k) = -2.2888e-06
k = 19, x_k = 7.2479e-06, max error = 9.5367e-06, f(x_k) = 7.2479e-06
k = 20, x_k = 2.4796e-06, max error = 4.7684e-06, f(x_k) = 2.4796e-06
k = 21, x_k = 9.5367e-08, max error = 2.3842e-06, f(x_k) = 9.5367e-08
k = 22, x_k = -1.0967e-06, max error = 1.1921e-06, f(x_k) = -1.0967e-06
k = 23, x_k = -5.0068e-07, max error = 5.9605e-07, f(x_k) = -5.0068e-07
k = 24, x_k = -2.0266e-07, max error = 2.9802e-07, f(x_k) = -2.0266e-07
k = 25, x_k = -5.3644e-08, max error = 1.4901e-07, f(x_k) = -5.3644e-08
k = 26, x_k = 2.0862e-08, max error = 7.4506e-08, f(x_k) = 2.0862e-08
k = 27, x_k = -1.6391e-08, max error = 3.7253e-08, f(x_k) = -1.6391e-08
k = 28, x_k = 2.2352e-09, max error = 1.8626e-08, f(x_k) = 2.2352e-09
k = 29, x_k = -7.0781e-09, max error = 9.3132e-09, f(x_k) = -7.0781e-09
```

```
ans =

  -7.0781e-09
```

(2) The maximum error M_k after k iterations of the bisection method is given by

$$M_k = \frac{b-a}{2^{k+1}} \tag{1}$$

To obtain a maximum error less than $\varepsilon > 0$, we need that k satisfies the inequality

$$M_k < \varepsilon \iff \frac{b-a}{\varepsilon} < 2^{k+1} \tag{2}$$

Thus, we need

$$k > \log_2\left(\frac{b-a}{2\varepsilon}\right) \tag{3}$$

Since k must be an integer, the least number of iterations needed to guarantee an error no greater than ε is given by the ceiling of the left side of (3), that is, the smallest integer greater than or equal to RHS(3):

$$k = \left\lceil \log_2\left(\frac{b-a}{2\varepsilon}\right) \right\rceil \tag{4}$$

For $[a, b] = [-4.9, 5.1]$ and $\varepsilon = 10^{-2}$, this gives $k = \lceil 8.9658 \rceil = 9$; for $\varepsilon = 10^{-4}$, it gives $k = \lceil 15.6096 \rceil = 16$; and for $\varepsilon = 10^{-8}$ it gives $k = \lceil 28.8974 \rceil = 29$. These are exactly the number of iterations that were executed in the numerical experiments.

Problem 2.

(a) See `fixed.m` – also copied here for convenience.

```
function result = fixed(g, x0, epsilon, max_it)
    x_next = x0;

    for k = 0:max_it
        xk = x_next;
        x_next = g(xk);
        cauchy_error = abs(x_next - xk);

        fprintf(...
            'k = %d, x_k = %.5g, Cauchy error = %.5g, f(x_k) = %.5g\n',...
            k, xk, cauchy_error, x_next...
        )

        if cauchy_error < epsilon
            break;
        end
    end

    result = xk;
```

The following are relevant snippets from `outputs.txt`, which contains the output from the MATLAB console. For $x_0 = 5$:

```
>> fixed(@(x) x - atan(x), 5, 0, 0, 10)
k = 0, x_k = 5, Cauchy error = 1.3734, f(x_k) = 3.6266
k = 1, x_k = 3.6266, Cauchy error = 1.3017, f(x_k) = 2.3249
k = 2, x_k = 2.3249, Cauchy error = 1.1646, f(x_k) = 1.1603
k = 3, x_k = 1.1603, Cauchy error = 0.85945, f(x_k) = 0.30082
k = 4, x_k = 0.30082, Cauchy error = 0.29221, f(x_k) = 0.008611
k = 5, x_k = 0.008611, Cauchy error = 0.0086108, f(x_k) = 2.1282e-07
k = 6, x_k = 2.1282e-07, Cauchy error = 2.1282e-07, f(x_k) = 3.2028e-21
k = 7, x_k = 3.2028e-21, Cauchy error = 3.2028e-21, f(x_k) = 0
k = 8, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 9, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 10, x_k = 0, Cauchy error = 0, f(x_k) = 0

ans =

     0
```

For $x_0 = -5$:

```
>> fixed(@(x) x - atan(x), -5, 0, 0, 10)
k = 0, x_k = -5, Cauchy error = 1.3734, f(x_k) = -3.6266
k = 1, x_k = -3.6266, Cauchy error = 1.3017, f(x_k) = -2.3249
k = 2, x_k = -2.3249, Cauchy error = 1.1646, f(x_k) = -1.1603
k = 3, x_k = -1.1603, Cauchy error = 0.85945, f(x_k) = -0.30082
k = 4, x_k = -0.30082, Cauchy error = 0.29221, f(x_k) = -0.008611
k = 5, x_k = -0.008611, Cauchy error = 0.0086108, f(x_k) = -2.1282e-07
k = 6, x_k = -2.1282e-07, Cauchy error = 2.1282e-07, f(x_k) = -3.2028e-21
k = 7, x_k = -3.2028e-21, Cauchy error = 3.2028e-21, f(x_k) = 0
k = 8, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 9, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 10, x_k = 0, Cauchy error = 0, f(x_k) = 0

ans =

     0
```

For $x_0 = 1$:

```
>> fixed(@(x) x - atan(x), 1, 0, 0, 10)
k = 0, x_k = 1, Cauchy error = 0.7854, f(x_k) = 0.2146
k = 1, x_k = 0.2146, Cauchy error = 0.2114, f(x_k) = 0.0032063
k = 2, x_k = 0.0032063, Cauchy error = 0.0032063, f(x_k) = 1.0987e-08
k = 3, x_k = 1.0987e-08, Cauchy error = 1.0987e-08, f(x_k) = 0
k = 4, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 5, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 6, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 7, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 8, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 9, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 10, x_k = 0, Cauchy error = 0, f(x_k) = 0

ans =

     0
```

For $x_0 = -1$:

```
>> fixed(@(x) x - atan(x), -1, 0, 0, 10)
k = 0, x_k = -1, Cauchy error = 0.7854, f(x_k) = -0.2146
k = 1, x_k = -0.2146, Cauchy error = 0.2114, f(x_k) = -0.0032063
k = 2, x_k = -0.0032063, Cauchy error = 0.0032063, f(x_k) = -1.0987e-08
k = 3, x_k = -1.0987e-08, Cauchy error = 1.0987e-08, f(x_k) = 0
k = 4, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 5, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 6, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 7, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 8, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 9, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 10, x_k = 0, Cauchy error = 0, f(x_k) = 0

ans =

     0
```

For $x_0 = 0.1$:

```
>> fixed(@(x) x - atan(x), 0.1, 0, 0, 10)
k = 0, x_k = 0.1, Cauchy error = 0.099669, f(x_k) = 0.00033135
k = 1, x_k = 0.00033135, Cauchy error = 0.00033135, f(x_k) = 1.2126e-11
k = 2, x_k = 1.2126e-11, Cauchy error = 1.2126e-11, f(x_k) = 0
k = 3, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 4, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 5, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 6, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 7, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 8, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 9, x_k = 0, Cauchy error = 0, f(x_k) = 0
k = 10, x_k = 0, Cauchy error = 0, f(x_k) = 0

ans =

     0
```

(b) It appears that the algorithm converges to 0 from all the initial guesses that I experimented with. The ones that start closer to 0 get closer to 0 in fewer iterations than the ones that start farther from 0.

First, set $G = [-R, R]$, where $R > 0$ is large enough that $x_0 \in [-R, R]$. By the Fundamental Theorem of Calculus, if $x \in G$, then

$$|g(x)| = \left|x - \tan^{-1}(x)\right| = \left|\int_0^x \left(1 - \frac{1}{1+t^2}\right)\, \mathrm{d}t\right| \leq |x| \leq R \tag{5}$$

Therefore, $g(G) \subseteq G$. Furthermore, g is L-Lipschitz on $[-R, R]$ with $L = 1 - \frac{1}{1+R^2} < 1$ because

$$g'(x) = 1 - \frac{1}{1+x^2} \leq 1 - \frac{1}{1+R^2} \tag{6}$$

if $x \in G$. Therefore, g is a contraction on G, so the fixed point method must converge for any $x_0 \in G$. Since $G = [-R, R]$, and $R > 0$ was arbitrary, it follows that the fixed point method should converge for all initial guesses.

Second, if the initial guess x_0 is farther from the fixed point $z = 0$, then the error bound

$$|x_k - z| \leq \frac{L^k}{1-L}|x_1 - x_0| \tag{7}$$

is looser as L gets bigger, and we need to choose a bigger L when x_0 is farther from 0 because we need to choose R large enough so that $x_0 \in [-R, R]$ in order for the fixed point theorem to apply with the initial guess x_0. The looser bound for x_0 farther from 0 suggests that the algorithm will require more iterations when x_0 is farther from 0.

Problem 3.

First, we need to show that $g(G) \subseteq G$. Note that

$$g'(x) = \frac{1}{3}\left(x^2 - 2x - \frac{5}{4}\right), \qquad g''(x) = \frac{2}{3}(x-1) \tag{8}$$

The roots of g' are $1 \pm \frac{1}{2}\sqrt{4+5} = 1 \pm \frac{3}{2}$, and the only root of g'' is 1. Since $1 \pm \frac{3}{2} \notin [0, 2]$, the Extreme Value Theorem implies that

$$\max_{x\in G} g(x) = \max\{g(0), g(2)\} = \max\left\{\frac{4}{3}, \frac{1}{18}\right\} = \frac{4}{3} \tag{9}$$

$$\min_{x\in G} g(x) = \min\{g(0), g(2)\} = \min\left\{\frac{4}{3}, \frac{1}{18}\right\} = \frac{1}{18} \tag{10}$$

Therefore, $g(G) \subseteq \left[\frac{1}{18}, \frac{4}{3}\right] \subseteq G$. Furthermore, the Extreme Value Theorem also implies that

$$\max_{x\in G} g'(x) = \max\{g'(0), g'(2), g'(1)\} = \max\left\{-\frac{5}{12}, -\frac{9}{12}\right\} = -\frac{5}{12} \tag{11}$$

$$\min_{x\in G} g'(x) = \min\{g'(0), g'(2), g'(1)\} = \min\left\{-\frac{5}{12}, -\frac{9}{12}\right\} = -\frac{9}{12} \tag{12}$$

Therefore, $|g'| \leq \frac{9}{12}$ on G, so g is L-Lipschitz on G with $L = \frac{9}{12} < 1$. By the Contraction Mapping Theorem, there is a unique fixed point z of g on G, and for any $x_0 \in G$, the sequence $\{x_k\}_{k=0}^{\infty}$ defined recursively by $x_{k+1} = g(x_k)$ converges to z as $k \to \infty$.