Math 342: Project 1

Connor Emmons

Documentation: I used ChatGPT solely for looking up Latex commands. The main Project 1 MatLab script and all required dependencies are located in the Project 1 folder found here: https://github.com/Connor-Lemons/Emmons-Math-342. No other resources used.

Project 1:

Part 1: Aitken's delta-squared process

Problem 1: Consider the problem of approximating the root p of $f(x) = x + ln(x)$ using Fixed-Point iteration.

Part A: Show that the root p of $f(x)$ is also a fixed point of the function $g(x) = e^{-x}$.

Begin with the function $f(x) = x + ln(x)$. Finding the root p is equivalent to finding the value of x which satisfies the equation:

$$f(x) = 0 = x + ln(x) \tag{1}$$

Rewriting equation (1) gives:

$$-x = ln(x) \tag{2}$$

$$x = e^{-x} \tag{3}$$

Defining $g(x) = e^{-x}$ gives:

$$g(x) = x = e^{-x} \tag{4}$$

Finding the value of x which satisfies $g(x) = x$ is equivalent to finding the fixed point of $g(x)$. Because this is the same value of x that gives p, the root p of $f(x)$ is the same as the fixed point of $g(x)$.

Part B: Use the Fixed-Point Theorem to show that Fixed-Point Iteration with $g(x) = e^{-x}$ will converge to p for any $p_0 \neq p$ in the interval $\left[\frac{1}{e}, 1\right]$.

Theorem 2.4 states that if $g \in C[a, b]$ (i.e., $g(x) \in [a, b]$ for all $x \in [a, b]$), g' exists on (a, b), and there exists some constant $0 < k < 1$ such that $\left(\forall x \in (a, b)\right)(|g'(x)| \leq k)$, then for any $p_0 \in [a, b]$ the sequence $p_n = g(p_{n-1})$ converges to a unique fixed point $p \in [a, b]$.

Because $g(x)$ is a decreasing monotone function (i.e., $(\forall x \in R)(y \in R \geq x \Rightarrow g(y) \leq g(x))$, if $g(a) \in [a, b]$ and $g(b) \in [a, b]$, then $g \in C[a, b]$.

$$a = \frac{1}{e} = 0.3679 \tag{1}$$

$$b = 1 \tag{2}$$

$$g(a) = 0.6922 \Rightarrow g(a) \in [a, b] \tag{3}$$

$$g(b) = \frac{1}{e} = 0.3679 \Rightarrow g(b) \in [a, b] \tag{4}$$

Thus, both $g(a)$ and $g(b)$ are in the interval $[a, b]$.

Consider $g(x) = e^{-x}$ and its derivative $g'(x) = -e^{-x}$. Both of these functions are defined for all $x \in R$, and are continuous functions for real number inputs. Notably, this means that $g'(x)$ exists on the interval (a, b) because $(a, b) \subseteq R$. Because $g'(x)$ is a monotone increasing function, $g'(x)$ will have the largest value on the interval $[a, b]$ at a. This gives:

$$|g'(a)| = 0.6922 \tag{1}$$

Thus, $k = 0.7$ satisfies the condition

$$0 < 0.7 < 1: \big(\forall x \in (a, b)\big)(|g^{'}(x)| \leq k) \tag{2}$$

because

$$\big(\forall x \in (a, b)\big)(max(|g^{'}(x)|) = g'(a) = 0.6922 < 0.7) \tag{3}$$

Therefore, because $g(x)$ has been shown to satisfy all the conditions of Theorem 2.4 on the interval $[a, b]$, where $a = \frac{1}{e}$ and $b = 1$, $g(x)$ will converge to a unique fixed point p for any $p_0 \neq p \in \left[\frac{1}{e}, 1\right]$.

Part C: Write a MatLab script to implement Fixed-Point Iteration $p_n = g(p_{n-1})$ with $g(x) = e^{-x}$ and $p_0 = 0.4$. Use $TOL = |p_n - p_{n-1}| \leq 10^{-6}$ as the stopping criterion.

n	p_n	$\lvert p_n - p_{n-1} \rvert$
1	0.67032	0.27032
2	0.51154	0.15878
3	0.59957	0.088024
4	0.54905	0.05052
5	0.5775	0.028451
6	0.5613	0.016199
7	0.57047	0.0091664
8	0.56526	0.0052052
9	0.56821	0.00295
10	0.56654	0.0016737
11	0.56749	0.00094903
12	0.56695	0.00053831
13	0.56725	0.00030528
14	0.56708	0.00017314
15	0.56718	9.8194e-05
16	0.56712	5.5691e-05
17	0.56715	3.1584e-05
18	0.56714	1.7913e-05
19	0.56715	1.0159e-05
20	0.56714	5.7617e-06
21	0.56714	3.2677e-06
22	0.56714	1.8533e-06
23	0.56714	1.0511e-06

24	0.56714	5.9611e-07

Code can be found in Appendix A.

Part D: Show that the solution obtained using Fixed-Point Iteration converges linearly to p by showing that

$$\frac{|p_{n+1}-p|}{|p_n-p|} \approx \text{constant} \tag{0}$$

for the 4 largest values of n.

For the iteration above, use $n = 21,22,23$.

For $n = 21$:

$$\frac{|p_{n+1}-p|}{|p_n-p|} = 0.32537 \tag{1}$$

For $n = 22$:

$$\frac{|p_{n+1}-p|}{|p_n-p|} = 1.3102 \tag{2}$$

For $n = 23$:

$$\frac{|p_{n+1}-p|}{|p_n-p|} = 0 \tag{3}$$

Note that this quotient equals zero for $n = 23$ because the value of p is taken to be the value of p_{24}, which was the first p_n that satisfied the error requirement. It is difficult to determine from these points alone whether $\frac{|p_{n+1}-p|}{|p_n-p|} \approx \text{constant}$ holds. Calculating this quotient for all $n \in [1,23]$ gives a more accurate picture. When accounting for all $n \in [1,23]$, it is clearer that the quotient is approximately constant, and thus this iteration converges linearly.

n	$\frac{\lvert p_{n+1} - p\rvert}{\lvert p_n - p\rvert}$
1	0.53886
2	0.58321
3	0.55804
4	0.57232
5	0.56418
6	0.56886
7	0.5661
8	0.56786
9	0.56652
10	0.56787
11	0.56606
12	0.56894
13	0.56406
14	0.57258

15	0.55768
16	0.58411
17	0.53811
18	0.62109
19	0.48028
20	0.74799
21	0.32537
22	1.3102
23	0

Code can be found in Appendix A.

Problem 2: Consider the problem of approximating the root p of $f(x) = x + ln(x)$ using Steffensen's Method.

Part A: Write a MatLab script to implement Steffensen's Method with $g(x) = e^{-x}$ and $p_0 = 0.4$. Use $TOL = |p_n - p_{n-1}| \leq 10^{-6}$ as the stopping criterion.

n	p_n	$\|p_n - p_{n-1}\|$
1	0.5703	0.1703
2	0.56714	0.003151
3	0.56714	1.0178e-06
4	0.56714	1.0633e-13

Code can be found in Appendix A.

Part B: Show that the solution obtained using Fixed-Point Iteration converges linearly to p by showing that

$$\frac{|p_{n+1}-p|}{|p_n-p|^2} \approx constant \tag{0}$$

for the 4 largest values of n.

For the iteration above, use $n = 1,2,3$.

For $n = 1$:

$$\frac{|p_{n+1}-p|}{|p_n-p|^2} = 0.10244 \tag{1}$$

For $n = 2$:

$$\frac{|p_{n+1}-p|}{|p_n-p|^2} = 0.10262 \tag{2}$$

For $n = 3$:

$$\frac{|p_{n+1}-p|}{|p_n-p|^2} = 0 \tag{3}$$

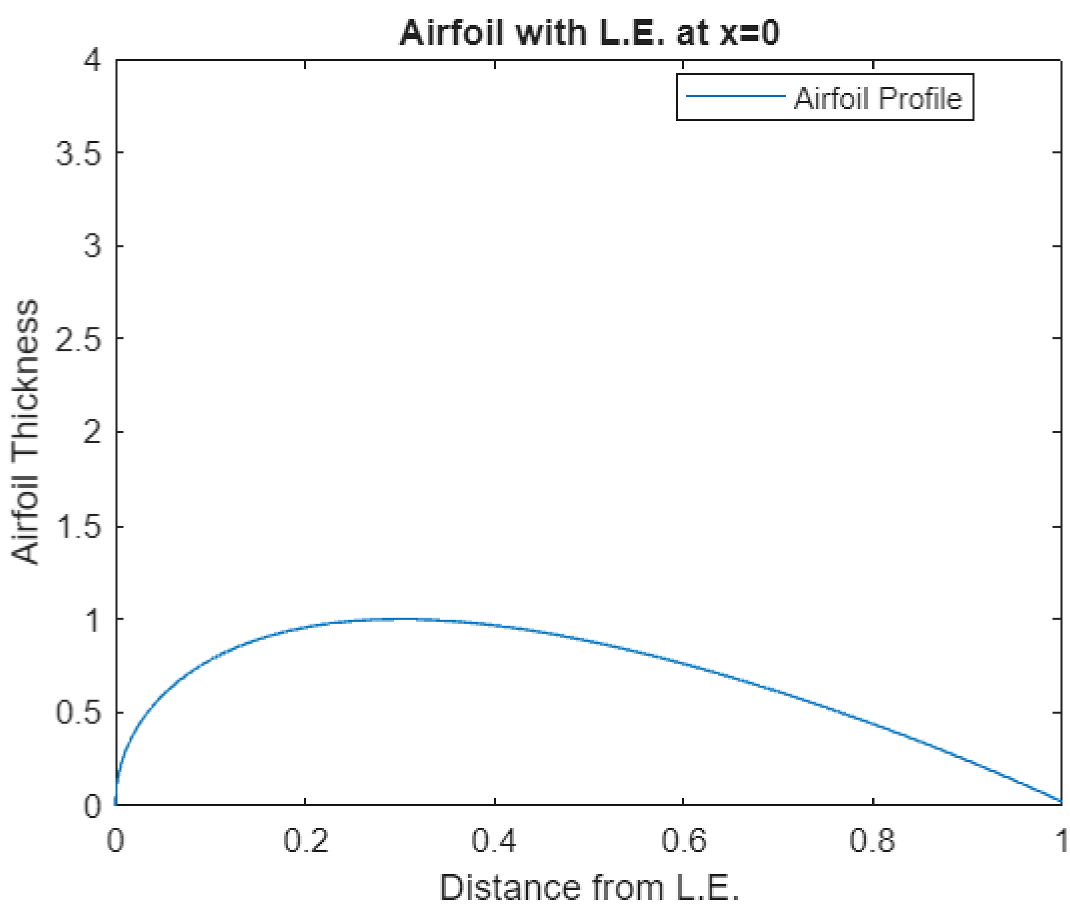
Note that this quotient equals zero for $n = 3$ because the value of p is taken to be the value of p_4, which was the first p_n that satisfied the error requirement. This quotient is relatively constant across all iterations of Steffensen's Method, and thus this iteration converges quadratically.

n	$\frac{\lvert p_{n+1} - p\rvert}{\lvert p_n - p\rvert^2}$
1	0.10244
2	0.10262
3	0

Code can be found in Appendix A.

Part 2: Newton's Method

Problem 1: Plot the airfoil described by $y = 2.969\sqrt{x} - 1.26x - 3.516x^2 + 2.843x^3 - 1.015x^4$ from $x = 0$ to $x = 1$ where y is the thickness of the airfoil and x is the distance from the leading edge of the airfoil ($x = 0$).



Problem 2: Consider the problem of finding the thickest point of the airfoil using Newton's Method.

Part A: Derive a function $f(x)$ such that the root p of $f(x)$ corresponds to the location of the thickest point of the airfoil.

Note that the thickest point of the airfoil described by y will simply be the maximum value of y. For some function of x, $g(x) = y = 2.969\sqrt{x} - 1.26x - 3.516x^2 + 2.843x^3 - 1.015x^4$, this will occur when the function's derivative is equal to zero. Thus, the function is maximized (i.e., the airfoil achieves its maximum thickness) at some from the leading edge $0 < p < 1$ such that $f(p) = g'(p) = 0$. Taking this derivative gives

$$f(x) = 1.484x^{-\frac{1}{2}} - 1.26 - 7.032x + 8.529x^2 - 4.06x^3 \quad (1)$$

where the root of $f(x)$ corresponds to the location of the thickest point of the airfoil.

Part B: Write a MatLab script to implement Newton's Method with $f(x)$ and $p_0 = 0.1$. Use $TOL = |p_n - p_{n-1}| \leq 10^{-7}$ as the stopping criterion.

n	p_n	$\lvert p_n - p_{n-1}\rvert$
1	0.1972486	0.09724864
2	0.2762193	0.0789707
3	0.2986261	0.02240674
4	0.2998248	0.001198702
5	0.2998279	3.103562e-06
6	0.2998279	2.069539e-11

Code can be found in Appendix A.

Problem 3: Now consider the problem of finding the thickest point of the airfoil using Steffensen's Method.

Part A: Derive Steffensen's Method from Newton's Method by replacing $f'(x)$ with the forward difference and applying the substitution $h = f(x)$. Explain why $h = f(x)$ is a valid choice.

Begin with Newton's Method:

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})} \quad (1)$$

The forward difference approximation for the derivative of the function is:

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \quad (2)$$

This comes from the definition of the derivative, which is:

$$f^{'}(x) = \lim_{h\to 0}\frac{f(x+h)-f(x)}{h} \tag{3}$$

Note that this means that the approximation in (2) gets more accurate the closer to zero h gets. Because of this, letting $h = f(x)$ is a valid choice because Newton's method iterates to find the x such that $f(x) = 0$, and subsequent x values will drive $f(x)$ to zero. Applying this gives the function $g(x)$:

$$f'(x) \approx g(x) = \frac{f(x+f(x))-f(x)}{f(x)} \tag{4}$$

Substituting into equation (1) gives

$$p_n = p_{n-1} - \frac{f(p_{n-1})}{g(p_{n-1})}; g(p_{n-1}) = \frac{f(p_{n-1}+f(p_{n-1}))-f(p_{n-1})}{f(p_{n-1})} \tag{5}$$

which is Steffensen's Method.

Part B: Write a MatLab script to implement Steffensen's Method with $f(x)$ and $p_0 = 0.1$. Use $TOL = |p_n - p_{n-1}| \leq 10^{-7}$ as the stopping criterion.

n	p_n	$\lvert p_n - p_{n-1}\rvert$
1	0.25318	0.15318
2	0.33455	0.081362
3	0.31674	0.017805
4	0.30374	0.012996
5	0.30004	0.0037032
6	0.29983	0.00021297
7	0.29983	6.4019e-07
8	0.29983	5.7534e-12

Code can be found in Appendix A.

Note that for Steffensen's Method, the equation $f(x) + x$ was in the iteration used to account for the fact that Steffensen's Method finds $p = g(p)$.

Part C: Show that the solution obtained in Part B converges quadratically to p by calculating $g'(p_n) \approx 0$ for your largest value of n.

For $n = 6$:

$$g'(p_6) = 0.08628 \approx 0 \tag{1}$$

The rest of the derivative values are as follows:

n	$g'(p_n)$
1	-1.563
2	1.043
3	0.4613

4	0.1087
5	0.005993
6	1.797e-5
7	0.03098
8	0.08628

Code can be found in Appendix A.

Project 1

Part 1

Problem 1

Part A

Part B

```
clear; clc;

syms x
g(x) = exp(-x);
g_prime(x) = diff(g, x);

a = 1/exp(1);
b = 1;

vpa(a, 4)
```

ans = 0.3679

```
vpa(b, 4)
```

ans = 1.0

```
vpa(g(a), 4)
```

ans = 0.6922

```
vpa(g(b), 4)
```

ans = 0.3679

```
vpa(g_prime(a), 4)
```

ans = -0.6922

Part C

```
p_0 = 0.4;
TOL = 10^-6;

[p_vec, p] = fixedPoint(g, p_0, TOL);
```

```
n            p_n          |p_n-p_{n-1}|
1            0.67032      0.27032
2            0.51154      0.15878
3            0.59957      0.088024
4            0.54905      0.05052
5            0.5775       0.028451
6            0.5613       0.016199
```

7	0.57047	0.0091664
8	0.56526	0.0052052
9	0.56821	0.00295
10	0.56654	0.0016737
11	0.56749	0.00094903
12	0.56695	0.00053831
13	0.56725	0.00030528
14	0.56708	0.00017314
15	0.56718	9.8194e-05
16	0.56712	5.5691e-05
17	0.56715	3.1584e-05
18	0.56714	1.7913e-05
19	0.56715	1.0159e-05
20	0.56714	5.7617e-06
21	0.56714	3.2677e-06
22	0.56714	1.8533e-06
23	0.56714	1.0511e-06
24	0.56714	5.9611e-07

Part D

```
const = [];

for n = 1:23

    const(n) = abs(p_vec(n+1) - p)/abs(p_vec(n) - p);

end

vpa(const', 5)
```

```
ans =
```

$$\begin{pmatrix} 0.53886 \\ 0.58321 \\ 0.55804 \\ 0.57232 \\ 0.56418 \\ 0.56886 \\ 0.5661 \\ 0.56786 \\ 0.56652 \\ 0.56787 \\ 0.56606 \\ 0.56894 \\ 0.56406 \\ 0.57258 \\ 0.55768 \\ 0.58411 \\ 0.53811 \\ 0.62109 \\ 0.48028 \\ 0.74799 \\ 0.32537 \\ 1.3102 \\ 0 \end{pmatrix}$$

Problem 2

Part A

```
clear; clc;

syms x
g(x) = exp(-x);

p_0 = 0.4;
TOL = 10^-6;

[p_vec, p] = SteffensenMethod(g, p_0, TOL);
```

```
n               p_n             |p_n-p_{n-1}|
1               0.5703          0.1703
2               0.56714         0.003151
3               0.56714         1.0178e-06
4               0.56714         1.0633e-13
```

Part B

```
const = [];
```

```
for n = 1:3

    const(n) = abs(p_vec(n+1) - p)/abs(p_vec(n) - p)^2;

end

vpa(const', 5)
```

ans =

$$\begin{pmatrix} 0.10244 \\ 0.10262 \\ 0 \end{pmatrix}$$

Part 2

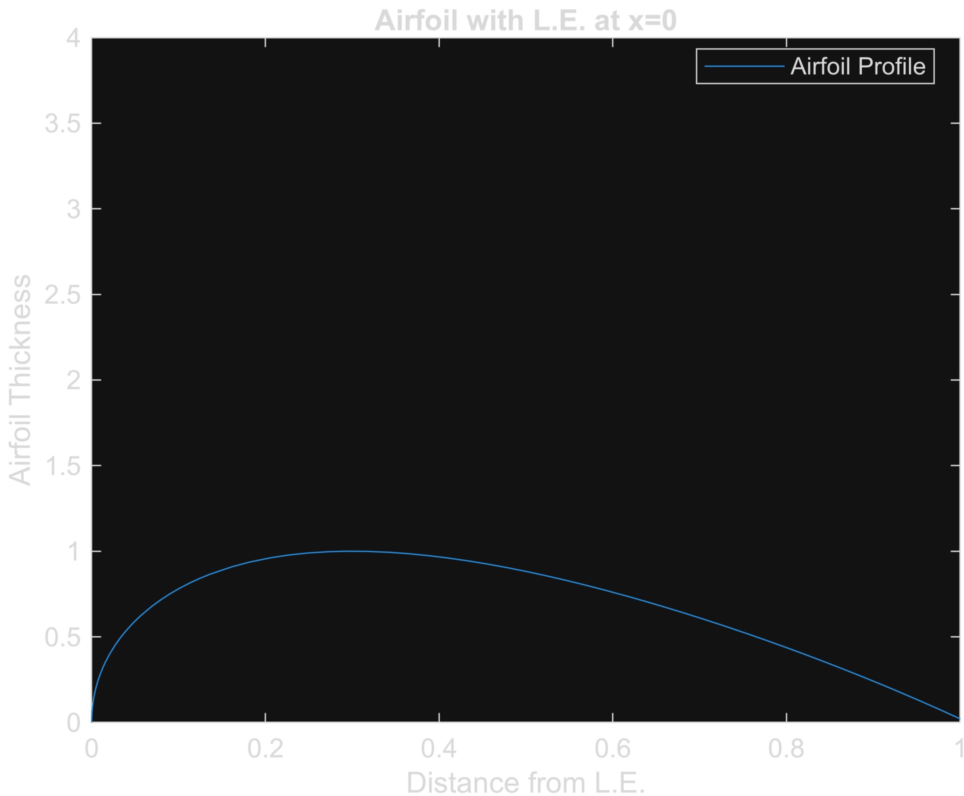
Problem 1

```
clear; clc;

syms x

y(x) = 2.969*sqrt(x) - 1.26*x - 3.516*x^2 + 2.843*x^3 - 1.015*x^4;

figure
fplot(y, [0, 1])
axis([0 1 0 4])
title("Airfoil with L.E. at x=0")
xlabel("Distance from L.E.")
ylabel("Airfoil Thickness")
legend("Airfoil Profile", "Location", "best")
```



Problem 2

Part A

```
clear; clc;

syms x

y(x) = 2.969*sqrt(x) - 1.26*x - 3.516*x^2 + 2.843*x^3 - 1.015*x^4;
y_prime(x) = diff(y, x);

vpa(y_prime)
```

```
ans(x) =
```

$$8.529\,x^2 - 7.032\,x + \frac{1.484}{\sqrt{x}} - 4.06\,x^3 - 1.26$$

Part B

```
p_0 = 0.1;
TOL = 10^-7;

p = NewtonMethod(y_prime, p_0, TOL);
```

```
n            p_n           |p_n-p_{n-1}|
1            0.1972486     0.09724864
2            0.2762193     0.0789707
```

```
3             0.2986261      0.02240674
4             0.2998248      0.001198702
5             0.2998279      3.103562e-06
6             0.2998279      2.069539e-11
```

Problem 3

Part A

Part B

```
clear; clc;

syms x

y(x) = 2.969*sqrt(x) - 1.26*x - 3.516*x^2 + 2.843*x^3 - 1.015*x^4;
y_prime(x) = diff(y, x) + x;

p_0 = 0.1;
TOL = 10^-7;

[p_vec, p] = SteffensenMethod(y_prime, p_0, TOL);
```

```
n             p_n            |p_n-p_{n-1}|
1             0.25318        0.15318
2             0.33455        0.081362
3             0.31674        0.017805
4             0.30374        0.012996
5             0.30004        0.0037032
6             0.29983        0.00021297
7             0.29983        6.4019e-07
8             0.29983        5.7534e-12
```

```
vpa(p_vec)
```

ans = $\begin{pmatrix} 0.2532 & 0.3345 & 0.3167 & 0.3037 & 0.3 & 0.2998 & 0.2998 & 0.2998 \end{pmatrix}$

Part C

```
y(x) = 2.969*sqrt(x) - 1.26*x - 3.516*x^2 + 2.843*x^3 - 1.015*x^4;
y_prime(x) = diff(y, x);

f(x) = (y_prime(x + y_prime(x)) - y_prime(x))/y_prime(x);
g(x) = x - y_prime(x)/f(x);
g_prime(x) = diff(g, x);

vpa(g_prime(p_vec'))
```

ans =

$$\begin{pmatrix} -1.563 \\ 1.043 \\ 0.4613 \\ 0.1087 \\ 0.005993 \\ 1.797\text{e-}5 \\ 0.03098 \\ 0.08628 \end{pmatrix}$$