Math 342: Homework 2

Connor Emmons

Documentation: I used ChatGPT solely for looking up Latex commands. The main Homework 2 MatLab script and all required dependencies are located in the Homework 2 folder found here: https://github.com/Connor-Lemons/Emmons-Math-342. No other resources used.

Problem 1 (1):

Use the Bisection method to find p_3 for $f(x) = \sqrt{x} - cos(x) = 0$ on $[0,1]$.

To begin, set $a_1 = 0, b_1 = 1, p_1 = 0.5$. Calculating the value of f at each of these gives:

$$f(a_1) = f(0) = \sqrt{0} - cos(0) = -1$$

$$f(b_1) = f(1) = \sqrt{1} - cos(1) = 0.4597$$

$$f(p_1) = f(0.5) - \sqrt{0.5} - cos(0.5) = -0.1705$$

Because $f(p_1)$ has the same sign as $f(a_1)$, set $a_2 = p_1 = 0.5$ and $b_2 = b_1 = 1$. This gives $p_2 = 0.75$. Calculating the value of f at each of these gives:

$$f(a_2) = f(0.5) = -0.1705$$

$$f(b_2) = f(1) = 0.4597$$

$$f(p_2) = f(0.75) = 0.1343$$

Because $f(p_2)$ has the same sign as $f(b_2)$, set $a_3 = a_2 = 0.5$ and $b_2 = p_2 = 0.75$. This gives $p_3 = 0.625$.

Problem 2 (14):

Find an approximation to $\sqrt{3}$ correct to within 10^{-4} using the Bisection Algorithm.

Consider the function $f(x) = x^2 - 3$. Note that the positive root to this function is the value of $\sqrt{3}$. Thus, using the Bisection method to find the roots of $f(x)$ allows the approximation of $\sqrt{3}$. If the error between the approximation and the actual value is to be less than 10^{-4}, then Theorem 2.1 can be used to find the bound on the number of iterations necessary to achieve such accuracy. Use the interval $[1,2]$, which gives a $= 1$ and $b = 2$.

$$\frac{2-1}{2^n} \leq 10^{-4}$$

$$log_2\left(\frac{1}{10^{-4}}\right) \leq n$$

$$n \geq 13.2877 \rightarrow n = 14$$

Thus, the first value produced by the Bisection method which is guaranteed to satisfy the accuracy requirement is p_{14}. Performing the Bisection method for this problem reveals that the accuracy specification is met by $p_{13} = 1.7321$.

n	a_n	b_n	p_n	$f(p_n)$
1	1	2	1.5	-0.75
2	1.5	2	1.75	0.0625
3	1.5	1.75	1.625	-0.3594
4	1.625	1.75	1.6875	-0.1523

5	1.6875	1.75	1.7188	-0.0459
6	1.7188	1.75	1.7344	0.008057
7	1.7188	1.7344	1.7266	-0.01898
8	1.7266	1.7344	1.7305	-0.005478
9	1.7305	1.7344	1.7324	0.001286
10	1.7305	1.7324	1.7314	-0.002097
11	1.7314	1.7324	1.7319	-0.000406
12	1.7319	1.7324	1.7322	0.0004397
13	1.7319	1.7322	1.7321	0.00001682

The code and output can be found at the end of this document.

Problem 3 (10):

Use Theorem 2.3 to show that $g(x) = 2^{-x}$ has a unique fixed point on $\left[\frac{1}{3}, 1\right]$.

Because $g(x)$ is a continuously decreasing function, as long as the endpoints are within a given interval, the rest of the function is guaranteed to be within that interval. Evaluating $g(x)$ at each end on the interval gives:

$$g\left(\frac{1}{3}\right) = 0.7937$$

$$g(1) = 0.5$$

Given that $g \in \left[\frac{1}{3}, 1\right]$ for all $x \in \left[\frac{1}{3}, 1\right]$, g is guaranteed to have at least one fixed point in the interval $\left[\frac{1}{3}, 1\right]$.

Again, because g is a continuously decreasing function, and because g has a continuously decreasing derivative, the largest value of the derivative of g will occur at the leftmost endpoint.

$$g'\left(\frac{1}{3}\right) = -0.5502$$

Because there is a positive constant $k < 1$ for which $|g'(x)| \leq k$ for all $x \in \left[\frac{1}{3}, 1\right]$, Theorem 2.3 says that there is exactly one fixed point in the given interval.

By Corollary 2.5, the error bound for fixed point iteration is given by:

$$error \leq k^n(max\{p_0 - a, b - p_0\})$$

Note that the bound on the error will be minimized when p_0 is chosen exactly between a and b. Applying this to the problem with a desired accuracy of 10^{-4} gives:

$$k^n\left(\frac{2}{3}\right) \leq 10^{-4}$$

From the previous analysis of the derivative of g on the given interval, a suitable value of k is $k = 0.551$. Solving the above inequality for n gives:

$$n \geq 14.77$$

This means that the p_{15} is the first approximation that is guaranteed to meet the accuracy requirements. Performing fixed point iteration reveals that $p_9 = 0.64117$ meets the accuracy requirement of 10^{-4}.

n	p_n	$\lvert p_n - p_{n-1} \rvert$
1	0.62996	0.036706
2	0.64619	0.016234
3	0.63896	0.0072304
4	0.64217	0.0032103
5	0.64075	0.0014274
6	0.64138	0.00063427
7	0.6411	0.00028192
8	0.64122	0.00012529
9	0.64117	0.000055684

The code and output can be found at the end of this document.

Problem 4 (1):

Let $f(x) = x^2 - 6$ and $p_0 = 1$. Use Newton's method to find p_2.

$$p_1 = p_0 - \frac{f(p_0)}{f'(p_0)} = 1 - \frac{-5}{2} = \frac{7}{2}$$

$$p_2 = \frac{7}{2} - \frac{f\left(\frac{7}{2}\right)}{f'\left(\frac{7}{2}\right)} = \frac{73}{28}$$

Problem 5 (12a):

Use both Newton's Method (p_0 is the interval midpoint) and Secant Method (p_0 and p_1 are interval endpoints) to find solutions with accuracy 10^{-7} for:

$$x^2 - 4x + 4 - ln(x) = 0; [1,2]; [2,4]$$

Newton's Method on [1,2]: $p_4 = 1.412391$

n	p_n	$\lvert p_n - p_{n-1} \rvert$
1	1.4067	0.09327906
2	1.41237	0.005649022
3	1.412391	2.121447e-05
4	1.412391	2.988791e-10

Secant Method on [1,2]: $p_8 = 1.412391$

n	p_n	$\lvert p_n - p_{n-1} \rvert$

2	1.590616	0.4093839
3	1.284548	0.3060683
4	1.427966	0.1434183
5	1.413635	0.01433147
6	1.412378	0.001256454
7	1.412391	1.299672e-05
8	1.412391	1.073098e-08

Newton's Method on $[2,4]$: $p_4 = 3.057104$

n	p_n	$\lvert p_n - p_{n-1} \rvert$
1	3.059167	0.05916737
2	3.057106	0.002061319
3	3.057104	2.504693e-06
4	3.057104	3.698235e-12

Secant Method on $[2,4]$: $p_{10} = 3.057104$

n	p_n	$\lvert p_n - p_{n-1} \rvert$
2	2.419219	1.580781
3	2.75604	0.3368211
4	3.317023	0.5609828
5	3.009769	0.3072535
6	3.050671	0.04090175
7	3.057289	0.006618332
8	3.057103	0.0001861977
9	3.057104	7.059887e-07
10	3.057104	7.719825e-11

The code and output can be found at the end of this document.

Problem 6 (4b):

Use Mueller's Method to find the zeros of $f(x) = x^4 - 2x^3 - 12x^2 + 16x - 40$ with an accuracy of 10^{-5}.

Looking at a graph of the function, there are likely real zeros around $x = -3$ and $x = -4$. Choosing $p_0 = -1, p_1 = -2, p_2 = -3$ and $p_0 = 5, p_1 = 6, p_2 = 7$ should produce these zeros, and indeed running Mueller's Method with these initial conditions produces $x = -3.5482$ and $x = 4.3811$ respectively.

The inflection point just to the right of $x = 0$ indicates a likely imaginary root at this location, so choose $p_0 = 1, p_1 = 2, p_2 = 3$ for Mueller's Method. This produces $x = 0.58356 + 1.4942i$, which means that $x = 0.58356 - 1.4942i$ is also a root.

Mueller's Method for $p_0 = -1, p_1 = -2, p_2 = -3 \rightarrow p = -3.5482$

n	p	$f(p)$

3	-3.8366	51.588
4	-3.5245	-3.5838
5	-3.5478	-0.058736
6	-3.5482	-4.3176e-05
7	-3.5482	4.1489e-11

Mueller's Method for $p_0 = 5, p_1 = 6, p_2 = 7 \rightarrow p = 4.3811$

n	p	$f(p)$
3	4.3964 -0.56232i	-22.392
4	3.8375 -0.59332i	-70.125
5	4.4484 -0.099854i	8.445
6	4.3801 -0.0058419i	-0.13351
7	4.3811 -6.0609e-05i	-0.0030177
8	4.3811 -5.4476e-09i	4.1537e-08
9	4.3811 -2.1934e-16i	-7.1054e-14

Mueller's Method for $p_0 = 1, p_1 = 2, p_2 = 3 \rightarrow p = 0.58356 \pm 1.4942i$

n	p	$f(p)$
3	-0.7047 +0 i	-56.288
4	0.65918 +3.214 i	209.54
5	0.27268 +0.53586i	-32.695
6	0.38851 +1.3029 i	-10.019
7	0.542 +1.4709 i	-1.2194
8	0.58457 +1.4936 i	-0.035052
9	0.58356 +1.4942 i	5.8656e-05
10	0.58356 +1.4942 i	-3.4301e-10

The code and output can be found at the end of this document.

Problem 7 (6):

Show that the sequence $p_n = \frac{1}{n}$ converges linearly to $p = 0$ and determine the n which satisfies $|p_n - p| \leq 5 \times 10^{-2}$.

Consider the limit $\lim_{n\to\infty} \frac{|p_{n+1}-p|}{|p_n-p|}$, which simplifies to $\lim_{n\to\infty} \frac{|p_{n+1}|}{|p_n|}$. This limit evaluates to 1, which means that the original sequence converges to $p = 0$ of order 1 and with asymptotic error constant 1. In order to find n such that $|p_n - p| \leq 5 \times 10^{-2}$, simply plug in and solve for n.

$$\frac{1}{n} - 0 \leq 5 \times 10^{-2}$$

$$n \geq \frac{1}{5 \times 10^{-2}} = 20$$

Homework 2

Problem 1 (1)

```
clear; clc;

syms x

f(x) = sqrt(x) - cos(x);

a_1 = 0;
b_1 = 1;
p_1 = (a_1 + b_1)/2
```

```
p_1 = 0.5000
```

```
vpa(f(a_1), 4)
```

ans = -1.0

```
vpa(f(b_1), 4)
```

ans = 0.4597

```
vpa(f(p_1), 4)
```

ans = -0.1705

```
a_2 = p_1;
b_2 = b_1;
p_2 = (a_2 + b_2)/2
```

```
p_2 = 0.7500
```

```
vpa(f(a_2), 4)
```

ans = -0.1705

```
vpa(f(b_2), 4)
```

ans = 0.4597

```
vpa(f(p_2), 4)
```

ans = 0.1343

```
a_3 = a_2;
b_3 = p_2;
```

```
p_3 = (a_3 + b_3)/2
```

```
p_3 = 0.6250
```

Problem 2 (14)

```
clear; clc;

syms n x

TOL = 10^-4
```

```
TOL = 1.0000e-04
```

```
f(x) = x^2 - 3;

n = log2(1/TOL)
```

```
n = 13.2877
```

```
bisectionMethod(f, 1, 2, TOL)
```

```
n           a           b           p           f(p)
1           1           2           1.5         -0.75
2           1.5         2           1.75        0.0625
3           1.5         1.75        1.625       -0.35938
4           1.625       1.75        1.6875      -0.15234
5           1.6875      1.75        1.7188      -0.045898
6           1.7188      1.75        1.7344      0.0080566
7           1.7188      1.7344      1.7266      -0.018982
8           1.7266      1.7344      1.7305      -0.0054779
9           1.7305      1.7344      1.7324      0.0012856
10          1.7305      1.7324      1.7314      -0.0020971
11          1.7314      1.7324      1.7319      -0.00040603
12          1.7319      1.7324      1.7322      0.0004397
13          1.7319      1.7322      1.7321      1.6823e-05
ans = 1.7321
```

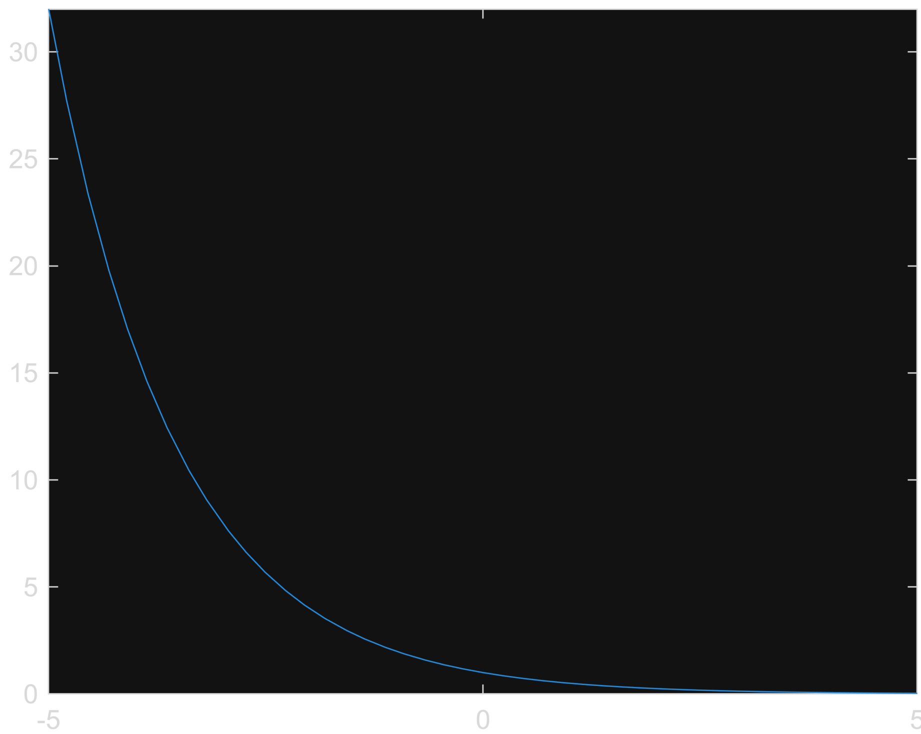
Problem 3 (10)

```
clear; clc;

syms x n

g(x) = 2^(-x)
```

g(x) =

$$\frac{1}{2^x}$$

```
fplot(g)
```




```
vpa(g(1/3), 4)
```

ans = 0.7937

```
vpa(g(1), 4)
```

ans = 0.5

```
vpa(subs(diff(g, x), x, 1/3), 4)
```

ans(x) = -0.5502

```
n_sol = vpasolve(0.551^n*(2/3)==10^-4, n)
```

n_sol = $14.772773267990065312154981867054$

```
fixedPoint(g, 2/3, 10^-4);
```

```
n               p_n             |p_n-p_{n-1}|
1               0.62996         0.036706
2               0.64619         0.016234
3               0.63896         0.0072304
4               0.64217         0.0032103
5               0.64075         0.0014274
```

```
6              0.64138        0.00063427
7              0.6411         0.00028192
8              0.64122        0.00012529
9              0.64117        5.5684e-05
```

Problem 4 (1)

```
clear; clc;

syms x

p_0 = 1;

f(x) = x^2 - 6
```

f(x) = $x^2 - 6$

```
f_prime(x) = diff(f, x)
```

f_prime(x) = $2\,x$

```
p_1 = p_0 - f(p_0)/f_prime(p_0)
```

p_1 =

$$\frac{7}{2}$$

```
p_2 = p_1 - f(p_1)/f_prime(p_1)
```

p_2 =

$$\frac{73}{28}$$

Problem 5 (12a)

```
clear; clc;

syms x

f(x) = x^2 - 4*x + 4 - log(x)
```

f(x) = $x^2 - \log(x) - 4\,x + 4$

```
g = @(x) x^2 - 4*x + 4 - log(x)
```

```
g = function_handle with value:
    @(x)x^2-4*x+4-log(x)
```

```
NewtonMethod(f, 1.5, 10^-7);
```

```
n               p_n             |p_n-p_{n-1}|
1               1.406721        0.09327906
2               1.41237         0.005649022
3               1.412391        2.121447e-05
4               1.412391        2.988791e-10
```

```
secantMethod(g, 1, 2, 10^-7);
```

```
n               p_n             |p_n-p_{n-1}|
2               1.590616        0.4093839
3               1.284548        0.3060683
4               1.427966        0.1434183
5               1.413635        0.01433147
6               1.412378        0.001256454
7               1.412391        1.299672e-05
8               1.412391        1.073098e-08
```

```
NewtonMethod(f, 3, 10^-7);
```

```
n               p_n             |p_n-p_{n-1}|
1               3.059167        0.05916737
2               3.057106        0.002061319
3               3.057104        2.504693e-06
4               3.057104        3.698235e-12
```

```
secantMethod(g, 2, 4, 10^-7);
```

```
n               p_n             |p_n-p_{n-1}|
2               2.419219        1.580781
3               2.75604         0.3368211
4               3.317023        0.5609828
5               3.009769        0.3072535
6               3.050671        0.04090175
7               3.057289        0.006618332
8               3.057103        0.0001861977
9               3.057104        7.059887e-07
10              3.057104        7.719825e-11
```

Problem 6 (4b)

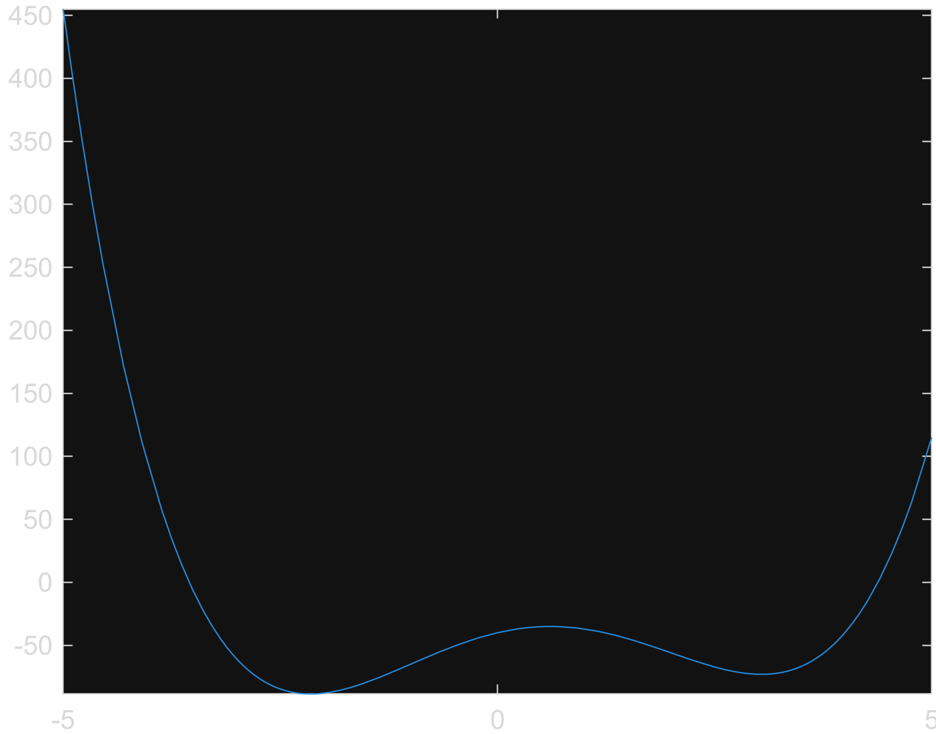
```
clear; clc;

syms x

coeffs = [1 -2 -12 16 -40];

f(x) = poly2sym(coeffs)
```

f(x) = $x^4 - 2\,x^3 - 12\,x^2 + 16\,x - 40$

```
vpa(f(0.5835597))
```

ans = $-35.031032165320712500205913895326$

```
fplot(poly2sym(coeffs))
```




```
MuellerMethod(coeffs, 1, 2, 3, 10^-5);
```

```
n            p                 f(p)
3            -0.7047 +0      i   -56.288
4            0.65918 +3.214  i   209.54
5            0.27268 +0.53586i   -32.695
6            0.38851 +1.3029 i   -10.019
7            0.542   +1.4709 i   -1.2194
8            0.58457 +1.4936 i   -0.035052
9            0.58356 +1.4942 i   5.8656e-05
10           0.58356 +1.4942 i   -3.4301e-10
```

```
MuellerMethod(coeffs, -1, -2, -3, 10^-5);
```

```
n            p                 f(p)
3            -3.8366 +0      i   51.588
4            -3.5245 +0      i   -3.5838
5            -3.5478 +0      i   -0.058736
6            -3.5482 +0      i   -4.3176e-05
7            -3.5482 +0      i   4.1489e-11
```

```
MuellerMethod(coeffs, 5, 6, 7, 10^-5);
```

```
n            p                 f(p)
3            4.3964  -0.56232i   -22.392
4            3.8375  -0.59332i   -70.125
5            4.4484  -0.099854i   8.445
6            4.3801  -0.0058419i   -0.13351
7            4.3811  -6.0609e-05i   -0.0030177
8            4.3811  -5.4476e-09i   4.1537e-08
9            4.3811  -2.1934e-16i   -7.1054e-14
```

Problem 7 (6)

```
clear; clc;

n = 1/(5e-2)
```

```
n = 20
```