Homework 2

APM 5333.

Section 2.2.

problem 6. Added instruction: Verify each iteration formular has the fixed-point $p = 7^{1/5}$.

(b) $p_n = p_{n-1} - \dfrac{p_{n-1}^5 - 7}{p_{n-1}^2}$.

Let $g(p_{n-1}) = p_{n-1} - \dfrac{p_{n-1}^5 - 7}{p_{n-1}^2}$ and $p_{n-1} \to x$.

$$g(x) = x - \frac{x^5 - 7}{x^2}.$$

We know x is a fixed of g if $g(x) = x$. If $x = 7^{1/5}$. then $g(7^{1/5}) = 7^{1/5}$.

$$g(7^{1/5}) = 7^{1/5} - \frac{(7^{1/5})^5 - 7}{(7^{1/5})^2}$$

$$= 7^{1/5} - \frac{7 - 7}{(7^{1/5})^2}$$

$$= 7^{1/5}$$

This implies that $p = 7^{1/5}$ is a fixed-point of g.

$$g(x) = x - \frac{x^5 - 7}{x^2}$$

$$g'(x) = 1 - 3x^2 + \frac{14}{x^3}$$

$$g'(7^{1/5}) = \left| 1 - 3(7^{1/5})^2 + \frac{14}{(7^{1/5})^3} \right|$$

$$g'(7^{1/5}) = \left| 1 - 3 \cdot 7^{2/5} + \frac{14}{7^{3/5}} \right| \simeq 1.17790 > 1$$

Hence $P_n = g(P_{n-1})$ does not converge to $P = 7^{1/5}$.

(c) $P_n = P_{n-1} - \dfrac{P_{n-1}^5 - 7}{P_{n-1}^2 - 21}$

Let $P_n = g(P_{n-1}) = P_{n-1} - \dfrac{P_{n-1}^5 - 7}{5P_{n-1}^4}$ and let $P_{n-1} \to x$

$$g(x) = x - \frac{x^5 - 7}{5x^4}$$

Want to show $g(x) = x$ ie $g(7^{1/5}) = 7^{1/5}$

$$g(7^{1/5}) = 7^{1/5} - \frac{(7^{1/5})^5 - 7}{5(7^{1/5})^4}$$

$$= 7^{1/5} - \frac{7-7}{5(7^{4/5})}$$

$$= 7^{1/5}$$

Thus $P = 7^{1/5}$ is a fixed point of g.

Also $g'(x) = 1 - \frac{1}{5} + \frac{28}{5x^5}$ $\qquad -\frac{7}{5}x^{-4-1}$

$$|g'(7^{1/5})| = \left| \frac{4}{5} + \frac{28}{5(7^{1/5})^5} \right| \qquad -4 \times -7\ \frac{x^{-5}}{5}$$

$$\frac{28}{5x^5}$$

$$= \left| \frac{4}{5} + \frac{28}{35} \right|$$

$$= \left| \frac{8}{5} \right| = 1.6 > 1$$

Hence $P_n = g(P_{n-1})$ does not converge to $p = 7^{1/5}$.

(d) $P_n = P_{n-1} - \frac{P_{n-1}^5 - 7}{12}$

Let $P_n = g(P_{n-1}) = P_{n-1} - \frac{P_{n-1}^5 - 7}{12}$ and let $P_{n-1} \to x$

$$g(x) = x - \frac{x^5 - 7}{12}.$$

WTS $g(7^{1/5}) = 7^{1/5}$

$$\Rightarrow g(7^{1/5}) = 7^{1/5} - \frac{(7^{1/5})^5 - 7}{12}.$$

$$= 7^{1/5} - \frac{7-7}{12}$$

$$= 7^{1/5}$$

$p = 7^{1/5}$ is a fixed point.

$$g(x) = x - \frac{x^5 - 7}{12}.$$

$$g'(x) = 1 - \frac{5x^4}{12}.$$

$$g'(7^{1/5}) = \left| 1 - \frac{5(7^{1/5})^4}{12} \right|$$

$$= \left| 1 - \frac{5 \cdot 7^{4/5}}{12} \right|$$

$$= 0.9764 \quad < 1$$

Hence $g(P_{n-1}) \to P$.

11. Using fixed method to find an approximation to $\sqrt{3}$. that is accurate to 10^{-4}.

$x = \sqrt{3} \Rightarrow x^2 - 3 = 0$.

$f(x) = x^2 - 3$.

Problem 14 from Homework 1, we used bisection method to approximate $\sqrt{3}$ and $x \in [1, 2]$. Since we are going to compare the two results, we will choose $P_0 = 1.5$ which the midpoint of 1 and 2.

$f(x) = x^2 - 3$

$x^2 - 3 = 0$

$\frac{x^2}{x} = \frac{3}{x}$

$x = \frac{3}{x}$

$x - \frac{3}{x} = 0$

$x + x - x - \frac{3}{x} = 0$ (adding and Subtracting x)

$2x = x + \frac{3}{x}$

$x = \frac{1}{2}\left(x + \frac{3}{x}\right)$

$\Rightarrow g(x) = x$ and $g(x) = \frac{1}{2}\left(x + \frac{3}{x}\right)$.

$P_0 = 1.5$ and $g(x) = \frac{1}{2}\left(x + \frac{3}{x}\right)$.

$P_3 = 1.73205081$ to the accuracy of 10^{-4}.

```
>> ALG022
This is the Fixed-Point Method.
Input the function G(x) in terms of x
For example: cos(x)
 1/2 * (x + 3/x)
Input initial approximation
 1.5
Input tolerance
 10^(-4)
Input maximum number of iterations - no decimal point
 20
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
FIXED-POINT METHOD
  I    P
  1    1.75000000e+00
  2    1.73214286e+00
  3    1.73205081e+00

Approximate solution P =   1.73205081
Number of iterations =   3    Tolerance = 1.00000000e-04
```

Comparing with problem 14 in homework 1, We used the the same TOL = 10^{-4} and initial approximation as the midpoint of the interval used for the bisection method. It can be observed that the fixed point iteration method Converges faster that is we achieved the same accuracy with fewer number of iterations.

14. (a) and (c)

(a) $2+\sin x - x = 0$ Use $[2,3]$.

$f(x) = 2+\sin x - x = 0$

$\Rightarrow x = 2+\sin x$

Thus $g(x) = 2+\sin x$. Use $P_0 = 2.5$

Using the theorem from class: If $g(x)$ is continuous on $[a,b]$ with $g(a) \geq a$ and $g(b) \leq b$ or $g(a) \leq a$ and $g(b) \geq b$, then there is a fixed-point of $g(x)$ in $[a,b]$.

$a=2$ $b=3$.

$g(2) = 2+\sin(2) \simeq 2.909 > 2$.

$g(3) = 2+\sin(3) \simeq 2.141 < 3$

So we have $g(2) \geq 2$ and $g(3) \leq 3$ hence there exist $c \in [2,3]$ such that $f(c)=0$ and c is a fixed point of $g(x)$.

Now the number of iteration needed to achieve the accuracy of 10^{-5}.

$|g'(x)| \leq k < 1$ for $x \in (a,b)$.

We known $|P_n - P| \leq k^n \max\{P_0 - a, b - P_0\}$.

$$|P_n - P| \leq \frac{k^n}{1-k}|P_1 - P_0|, \quad n = 1, 2, \cdots$$

$g'(x) = \cos(x)$. $|g'(2)| = |\cos(2)| \simeq 0.4161468365$.

$|g'(3)| = |\cos(3)| \simeq 0.9899924966$.

Max of these value is 0.9899924966, Thus $k = 0.9899924966$.

$P_1 = g(2.5) = 2 + \sin(2.5) = 2.59847144.$ where $P_0 = 2.5$

$\Rightarrow \frac{k^n}{1-k}|(P_1 - P_0)| \leq 10^{-5}$

$\Rightarrow \frac{(0.9899924966)^n}{(1-0.9899924966)} |2.59847144 - 2.5| \leq 10^{-5}$

$\Rightarrow (0.9899924966)^n \leq \frac{10^{-5} \cdot 0.0100075034}{0.09847144}$

Taking log of both sides, we have

$\Rightarrow \log(0.9899924966)^n \leq \log\left(10^{-5} \cdot \frac{0.0100075034}{0.09847144}\right)$

$\Rightarrow n \log(0.9899924966) \leq -5\log 10 + \log(0.1016284864)$

$\Rightarrow (-4.368 \times 10^{-3})n \leq -5\log 10 + \log(0.1016284864)$

$\Rightarrow n \geq \frac{-5.992984543}{-0.004368}$

$\Rightarrow n \geq 1372.020$

$\Rightarrow n \geq 1372.$

```
>> ALG022
This is the Fixed-Point Method.
Input the function G(x) in terms of x
For example: cos(x)
 2+sin(x)
Input initial approximation
 2.5
Input tolerance
 10^(-5)
Input maximum number of iterations - no decimal point
 100
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
FIXED-POINT METHOD
  I    P
  1    2.59847214e+00
  2    2.51680997e+00
  3    2.58492102e+00
  4    2.52836328e+00
  5    2.57551141e+00
  6    2.53632870e+00
  7    2.56897915e+00
  8    2.54183051e+00
  9    2.56444615e+00
 10    2.54563487e+00
 11    2.56130168e+00
 12    2.54826730e+00
 13    2.55912111e+00
 14    2.55008961e+00
 15    2.55760933e+00
 16    2.55135148e+00
 17    2.55656141e+00
 18    2.55222543e+00
 19    2.55583511e+00
```

20	2.55283080e+00
21	2.55533177e+00
22	2.55325015e+00
23	2.55498297e+00
24	2.55354068e+00
25	2.55474128e+00
26	2.55374195e+00
27	2.55457380e+00
28	2.55388140e+00
29	2.55445776e+00
30	2.55397801e+00
31	2.55437735e+00
32	2.55404495e+00
33	2.55432164e+00
34	2.55409133e+00
35	2.55428304e+00
36	2.55412346e+00
37	2.55425629e+00
38	2.55414573e+00
39	2.55423776e+00
40	2.55416115e+00
41	2.55422492e+00
42	2.55417184e+00
43	2.55421602e+00
44	2.55417925e+00
45	2.55420986e+00
46	2.55418438e+00
47	2.55420559e+00
48	2.55418793e+00
49	2.55420263e+00
50	2.55419040e+00
51	2.55420058e+00
52	2.55419210e+00

Approximate solution P = 2.55419210
Number of iterations = 52 Tolerance = 1.00000000e-05

(c) $3x^2 - e^x = 0$

$$3x^2 = e^x$$

$$x^2 = \frac{e^x}{3}$$

$$x = \sqrt{\frac{e^x}{3}}$$

$$g(x) = \sqrt{e^x/3}$$

Find the interval in which $g[a,b] \subseteq [a,b]$.

$$g'(x) = \frac{e^x}{6\sqrt{\frac{e^x}{3}}}$$

$$|g'(x)| < 1 \Rightarrow \left|\frac{e^x}{6\sqrt{\frac{e^x}{3}}}\right| < 1 \Rightarrow -6 < \frac{e^x}{\sqrt{\frac{e^x}{3}}} < 6$$

$$\Rightarrow -6 < \frac{\sqrt{3}\,e^x}{e^{\frac{1}{2}x}} < 6 \Rightarrow -\frac{6}{\sqrt{3}} < e^{\frac{1}{2}x} < \frac{6}{\sqrt{3}} \leftarrow \text{Rationalize.}$$

$$\Rightarrow -2\sqrt{3} < e^{\frac{1}{2}x} < 2\sqrt{3} \Rightarrow e^{\frac{1}{2}x} < 2\sqrt{3} \Rightarrow \frac{x}{2} < \ln(2\sqrt{3})$$

$$x < 2\ln(2\sqrt{3}) > 1.$$

We pick $[0,1]$. Now check if $g([0,1]) \subset [0,1]$

$$g(0) = \sqrt{\frac{e^0}{3}} = \frac{\sqrt{3}}{3} > 0$$

$$g(1) = \sqrt{\frac{e}{3}} \simeq 0.95189 < 1.$$

Thus $g([0,1]) \subset [0,1]$.

Now we look for the number of iterations to achieve the accuracy of 10^{-5}.

$a = 0$ and $b = 1$

$$|P_n - P_0| \leq \frac{k^n}{1-k}|P_1 - P_0|.$$

$$g'(x) = \frac{e^x}{6\sqrt{\frac{e^x}{3}}} \Rightarrow |g'(0)| = \left|\frac{e^0}{6\sqrt{\frac{e^0}{3}}}\right| \simeq 0.288675$$

$$|g'(1)| = \left|\frac{e}{6\sqrt{\frac{e}{3}}}\right| \simeq 0.47594.$$

The maximum is 0.4759448347

$P_1 = g(P_0)$, we pick P_0 to be the midpoint of 0 and 1

Thus, $P_0 = 0.5$.

$$P_1 = g(0.5) = \sqrt{\frac{e^{0.5}}{3}} \simeq 0.74133242.$$

$$\Rightarrow \frac{k^n}{1-k}|P_1 - P_0| \leq 10^{-5}$$

$$\Rightarrow \frac{(0.4759448347)^n}{1 - 0.4759448347}|0.74133242 - 0.5| \leq 10^{-5}$$

$$(0.4759448347)^n \leq \frac{10^{-5} \cdot 0.5240551653}{0.24133242}$$

Taking ln of both sides.

$$n \log(0.4759448347) \leq -5 \log 10 + \log(2.171507522)$$

$$-0.3224433821\, n \leq -4.663238662$$

$$n \geq \frac{-4.663238662}{-0.3224433821}$$

$$n \geq 14.462$$

$n \geq 15$ iterations.

```
>> ALG022
This is the Fixed-Point Method.
Input the function G(x) in terms of x
For example: cos(x)
 sqrt(exp(x)/3)
Input initial approximation
 0.5
Input tolerance
 10^(-5)
Input maximum number of iterations - no decimal point
 20
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
FIXED-POINT METHOD
  I    P
  1    7.41332420e-01
  2    8.36407007e-01
  3    8.77127740e-01
  4    8.95169428e-01
  5    9.03281143e-01
  6    9.06952163e-01
  7    9.08618411e-01
  8    9.09375718e-01
  9    9.09720122e-01
 10    9.09876791e-01
 11    9.09948068e-01
 12    9.09980498e-01
 13    9.09995254e-01
 14    9.10001967e-01

Approximate solution P =   0.91000197
Number of iterations =  14    Tolerance = 1.00000000e-05
```

24. We want to show that $A > 0$, $x_n = \frac{1}{2}x_{n-1} + \frac{A}{2x_{n-1}}$ for $n \geq 1$.

Converges to $\sqrt{A}$ whenever $x_0 > 0$.

As we can see, $x_n = g(x_{n-1})$.

$$g(x_{n-1}) = \frac{1}{2}x_{n-1} + \frac{A}{2x_{n-1}}$$

So, $\lim_{n\to\infty} x_n = \lim_{n\to\infty} g(x_{n-1}) = \lim_{n\to\infty}\left(\frac{1}{2}x_{n-1} + \frac{A}{2x_{n-1}}\right)$

As $n \to \infty$, $x_n \to x$. $\Rightarrow \lim_{n\to\infty} g(x_{n-1}) = x$

$$\lim_{n\to\infty} g(x_{n-1}) = \lim_{n\to\infty}\left(\frac{x_{n-1}}{2}\right) + \lim_{n\to\infty}\left(\frac{A}{2x_{n-1}}\right)$$

$$x = \frac{1}{2}\lim_{n\to\infty}(x_{n-1}) + \frac{1}{2}\cdot\frac{A}{\lim_{n\to\infty} x_{n-1}}$$

As $n \to \infty$, $x_{n-1} \to x$

$$x = \frac{1}{2}\cdot x + \frac{A}{2x}$$

Solving for x,

$$x - \frac{1}{2}x = \frac{A}{2x}$$

$$\frac{x}{2} = \frac{A}{2x}$$

$$2x^2 = 2A$$

$$x^2 = A, \quad x = \sqrt{A}.$$

Thus, $X_n = \frac{1}{2} X_{n-1} + \frac{A}{2X_{n-1}}$, $n \geq 1$, Converges to $\sqrt{A}$ whenever $X_0 > 0$.

$\square$.

(b) Using Similar approach as in (a) above, we'll arrive at $x = \pm\sqrt{A}$. But this point, we'll pick $-\sqrt{A}$ because $X_0 < 0$

In Conclusion, $X_n = \frac{1}{2} X_{n-1} + \frac{A}{2X_{n-1}}$, $n \geq 1$, Converges to $-\sqrt{A}$ whenever $X_0 < 0$.

$\square$.

Section 2.3

Problem Q: Use all 3 methods in this Section to find Solutions to within 10^{-7} for the following problems.

* Newton's method
* Secant Method
* False-position method.

(a) $x^2 - 4x + 4 - \ln x = 0$ for $x \in [2, 4]$.

For the Newton's method, $x_0 = \frac{2+4}{2} = 3$.

$f(x) = x^2 - 4x + 4 - \ln x = 0$.

$f'(x) = 2x - 4 - \frac{1}{x}$.

```
>> ALG023M
This is Newtons Method
Input the function F(x) in terms of x
For example: cos(x)
 x^2 - 4*x + 4 - log(x)
Input the derivative of F(x) in terms of x
 2*x - 4 - 1/x
Input initial approximation
 3
Input tolerance
 10^(-7)
Input maximum number of iterations - no decimal point
 100
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
Newtons Method
  I   P                 F(P)
  1   3.05916737e+00    3.6927458e-03
  2   3.05710605e+00    4.4761506e-06
  3   3.05710355e+00    6.6080474e-12
  4   3.05710355e+00    0.0000000e+00

Approximate solution = 3.0571035500e+00
with F(P) = 0.0000000000e+00
Number of iterations = 4
Tolerance = 1.0000000000e-07
Iteration no. stored in NN and iteration results in
```

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

$P_0 = 2$ and $P_1 = 4$

```
>> SecantM
This is the Secant Method
Input the function F(x) in terms of x
For example: cos(x)
 x^2 - 4*x + 4 - log(x)
Input initial approximations P0 and P1 on separate lines.
 2
 4
Input tolerance
 10^(-7)
Input maximum number of iterations - no decimal point
 100
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
Secant Method
  I   P               F(P)
  2   2.41921865e+00  -7.07700341e-01
  3   2.75603971e+00  -4.42198716e-01
  4   3.31702250e+00   5.35480732e-01
  5   3.00976896e+00  -8.22299668e-02
  6   3.05067071e+00  -1.14525310e-02
  7   3.05728904e+00   3.31528419e-04
  8   3.05710284e+00  -1.26180984e-06
  9   3.05710355e+00  -1.37960754e-10
 10   3.05710355e+00   0.00000000e+00

Approximate solution P =   3.05710355
with F(P) =   0.00000000
Number of iterations = 10
Tolerance = 1.00000000e-07
```

Method of False Position

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

$P_0 = 2$, $P_1 = 4$.

```
>> FalsePosition
This is the Method of False Position
Input the function F(x) in terms of x
For example: cos(x)
 x^2 - 4*x + 4 - log(x)
Input endpoints P0 < P1 on separate lines.
 2
 4
Input tolerance
 10^(-7)
Input maximum number of iterations - no decimal point
 100
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
METHOD OF FALSE POSITION OR REGULA FALSII

 I   P              F(P)
 2  2.41921865e+00 -7.07700341e-01
 3  2.75603971e+00 -4.42198716e-01
 4  2.93604457e+00 -2.00883848e-01
 5  3.01198157e+00 -7.84914992e-02
 6  3.04078742e+00 -2.88780406e-02
 7  3.05126966e+00 -1.03898916e-02
 8  3.05502608e+00 -3.70810330e-03
 9  3.05636483e+00 -1.31959622e-03
10  3.05684101e+00 -4.69120819e-04
11  3.05701026e+00 -1.66713174e-04
12  3.05707040e+00 -5.92377867e-05
```

13	3.05709177e+00	-2.10478491e-05
14	3.05709937e+00	-7.47841410e-06
15	3.05710206e+00	-2.65710534e-06
16	3.05710302e+00	-9.44076392e-07
17	3.05710336e+00	-3.35432533e-07
18	3.05710348e+00	-1.19179928e-07
19	3.05710353e+00	-4.23448885e-08

Approximate solution P = 3.05710353
with F(P) = -0.00000004
Number of iterations = 19 Tolerance = 1.00000000e-07

The Newton's method need the least number of iterations to achieve the accuracy of 10^{-7}.

Problem 14 $\quad f(x) = \tan(\pi x) - 6.$

```
>> ALG021
This is the Bisection Method.
Input the function F(x) in terms of x
For example: cos(x)
  tan(pi*x) - 6
Input endpoints A < B on separate lines
 0
 0.48
Input tolerance
 10^(-10)
Input maximum number of iterations - no decimal point
 10
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
Bisection Method
  I    P                F(P)
  1   2.40000000e-01   -5.0609375e+00
  2   3.60000000e-01   -3.8748918e+00
  3   4.20000000e-01   -2.1052571e+00
  4   4.50000000e-01    3.1375151e-01
  5   4.35000000e-01   -1.1711826e+00
  6   4.42500000e-01   -5.2452115e-01
  7   4.46250000e-01   -1.3434976e-01
  8   4.48125000e-01    8.1674389e-02
  9   4.47187500e-01   -2.8237441e-02
 10   4.47656250e-01    2.6230775e-02

Iteration number  10 gave approximation   0.44765625
F(P) =   0.02623078 not within tolerance :  1.00000000e-10
```

$$P_{10} = 0.44765625$$

And $\quad \frac{1}{\pi}\tan^{-1}(6) \simeq 0.447431543$

```
>> FalsePosition
This is the Method of False Position
Input the function F(x) in terms of x
For example: cos(x)
 tan(pi*x) - 6
Input endpoints P0 < P1 on separate lines.
 0
 0.48
Input tolerance
 10^(-10)
Input maximum number of iterations - no decimal point
 10
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
METHOD OF FALSE POSITION OR REGULA FALSII

  I    P                 F(P)
  2  1.81194242e-01 -5.36010528e+00
  3  2.86187166e-01 -4.74221095e+00
  4  3.48981227e-01 -4.05282126e+00
  5  3.87052621e-01 -3.30106907e+00
  6  4.10304720e-01 -2.54563776e+00
  7  4.24566483e-01 -1.85955039e+00
  8  4.33336313e-01 -1.29515333e+00
  9  4.38737409e-01 -8.68485046e-01
 10  4.42066949e-01 -5.66358054e-01

Iteration number  10 gave approximation   0.44206695
F(P) =  -0.56635805 not within tolerance:  1.00000000e-10
Iteration no. stored in NN and iteration results in PN
```

$P_{10} = 0.44206695$

$\frac{1}{\pi}\tan^{-1}(6) \simeq 0.447431543$

```
>> SecantM
This is the Secant Method
Input the function F(x) in terms of x
For example: cos(x)
 tan(pi*x) - 6
Input initial approximations P0 and P1 on separate lines.
 0
 0.48
Input tolerance
 10^(-10)
Input maximum number of iterations - no decimal point
 10
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
Secant Method
  I   P                F(P)
  2    1.81194242e-01   -5.36010528e+00
  3    2.86187166e-01   -4.74221095e+00
  4    1.09198611e+00   -5.70269458e+00
  5   -3.69229667e+00   -4.55114253e+00
  6   -2.26006499e+01   -2.94356269e+00
  7   -5.72228325e+01   -6.84237191e+00
  8    3.53875815e+00   -1.41720946e+01
  9   -1.13944405e+02   -5.82354540e+00
 10   -1.95894995e+02   -5.65760531e+00

Iteration number 10 gave approximation -195.89499482
with F(P) =  -5.65760531 not within tolerance   1.00000000e-10
Iteration no. stored in NN and iteration results in PN
```

$P_{10} = -195.89499482$ which is nowhere close to $\frac{1}{\pi}\tan^{-1}(6) \simeq 0.44743154 3$.

Hence the secant method failed.

The bisection method is the most successful

(30) Secant method can be written in a simpler form as $p_n = \dfrac{f(p_{n-1})\, p_{n-2} - f(p_{n-2})\, p_{n-1}}{f(p_{n-1}) - f(p_{n-2})}$, why is this formula less accurate?

We can observe that p_{n-1} and p_{n-2} are values approaching the real solution p. Hence these values might be close to each other. Thus $f(p_{n-1}) \cdot p_{n-2}$ and $f(p_{n-2}) \cdot p_{n-1}$ are going to be values that are very close to each other. So, taking the difference of these two values as shown in the formula above will lead to a lot of cancellation error. But this might not be the case in the actual Secant method formula $p_n = p_{n-1} - \dfrac{f(p_{n-1})(p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})}$ because we have two initial approximations which are used to get the next approximate solution. In the formula, we subtract subsequent terms that corrected from each other that is p_{n-1} and p_{n-2}. which gives us more accurate in the results that we obtain when we use the Secant Method formula as we have in the algorithm.

(32) Derive the error formular for Newton's method

$$|P - P_{n+1}| \le \frac{M}{2|f'(P_n)|}|P - P_n|^2$$ assuming the hypothesis of theorem 2.6 hold, that $|f'(P_n)| \neq 0$ and $M = \max|f''(x)|$.

We know that the Newton's method formula is given as $P_n = P_{n-1} - \frac{f(P_{n-1})}{f'(P_{n-1})}$, for $n \ge 1$. and we can write $P_n = g(P_{n-1})$ which implies that

$$g(P_{n-1}) = P_{n-1} - \frac{f(P_{n-1})}{f'(P_{n-1})}.$$

Let ε be the error which is equal to $P_n - P$. So we can write $\varepsilon_n = P_n - P$ and $\varepsilon_{n-1} = P_{n-1} - P$ then we have $\boxed{P_{n-1} = \varepsilon_{n-1} + P.}$

Since $P_n = \varepsilon_{n-1} + P$ then we can write $g(P_{n-1}) = g(\varepsilon_{n-1} + P)$

So $P_n = g(P_{n-1}) = g(\varepsilon_{n-1} + P)$.

Using Taylor's theorem which is $f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0) + \frac{f''(x_0)}{2!}(x - x_0)^2 + \cdots$

The Taylor series expansion of $P_n = g(P_{n-1})$ around $x_0 = P$

$$\Rightarrow P_n = g(P_{n-1}) = g(P) + g'(P)(P_{n-1} - P) + \frac{g''(P)}{2!}(P_{n-1} - P)^2 + \cdots$$

But $g(P) = P$ and $P_{n-1} - P = \varepsilon_{n-1}$

$$\Rightarrow P_n = P + g'(P)\cdot\varepsilon_{n-1} + \frac{g''(P)}{2!}(\varepsilon_{n-1})^2 + \cdots$$

$$\Rightarrow \quad P_n - P = g'(P) \cdot \varepsilon_{n-1} + \frac{g''(P)}{2!} (\varepsilon_{n-1})^2 + \cdots$$

$$|P_n - P| = \varepsilon_n$$

$$\Rightarrow \quad \varepsilon_n = g'(P) \cdot \varepsilon_{n-1} + \frac{g''(P)}{2!} (\varepsilon_{n-1})^2 + \cdots \quad \text{——} (1)$$

We know that $g(P_{n-1}) = P_{n-1} - \dfrac{f(P_{n-1})}{f'(P_{n-1})} \Rightarrow g(P) = P - \dfrac{f(P)}{f'(P)}$

So $g'(P) = \dfrac{d}{dp}\left[p - \dfrac{f(P)}{f'(P)}\right]$

$$\boxed{\frac{d}{dp} = \frac{v\,du - u\,dv}{v}}$$

$$= 1 - \left[\frac{f'(P) \cdot f'(P) - f(P) f''(P)}{[f'(P)]^2}\right]$$

$$= 1 - \frac{[f'(P)]^2}{[f'(P)]^2} + \frac{f(P) f''(P)}{[f'(P)]^2}$$

$$= \frac{f(P) f''(P)}{[f'(P)]^2}.$$

But p is the exact root of the function f So, $f(P) = 0$ which implies $g'(P) = \dfrac{f(P) f''(P)}{[f'(P)]^2} = 0$.

Now, to get $g''(P)$, we take the derivative again w.r.t p.

Since we do not want to have at least the second derivative in the equation, we can rewrite $g'(P) = f(P) \cdot \dfrac{f''(P)}{[f'(P)]^2}$

Then we can use the product rule to get derivative so that $u = f(p)$ and $v = \frac{f''(p)}{[f'(p)]^2}$

$$\boxed{\frac{d}{dp} = v\,du + u\,dv}$$

Thus, $g''(p) = f'(p) \cdot \left[\frac{f''(p)}{[f'(p)]^2}\right] + \underbrace{f(p) \cdot \left[\frac{f''(p)}{[f'(p)]^2}\right]'}$

goes to zero because $f(p) = 0$.

$$g''(p) = f'(p) \cdot \left[\frac{f''(p)}{[f'(p)]^2}\right]$$

So we we substitute the derivatives into the Taylor series expansion labelled equation 1 above we have,

$$\varepsilon_n = 0 \cdot \varepsilon_{n-1} + \cancel{f'(p)} \cdot \frac{f''(p)}{[f'(p)]^{\cancel{2}}} \cdot (\varepsilon_{n-1})^2$$

$$\varepsilon_n = \frac{f''(p)}{f'(p)} \cdot (\varepsilon_{n-1})^2, \quad f'(p) \neq 0.$$

Then we substitute back $\varepsilon_n = p_n - p$ and $\varepsilon_{n-1} = p_{n-1} - p$

$$\Rightarrow |p_n - p| = \frac{f''(p)}{f'(p)} \cdot |p_{n-1} - p|^2.$$

If we shift the index of n by 1, $\varepsilon_{n+1} = p_{n+1} - p$.

$$\Rightarrow |p_{n+1} - p| = \frac{f''(p)}{f'(p)} |p_n - p|^2.$$ Since we are taking the absolute value, $p - p_{n+1} = p_{n+1} - p$ and by definition $M = \max|f''(x)|$, which gives us the inequality $\leq$ in the final solution.

Hence

$$|P - P_{n+1}| \le \frac{M}{f'(P)} |P_n - P|^2 .$$

$\square$.

Section 2.4

(4) Use the midpoint as the initial guess in each part.

a. $P_{15} = 0.739076586$.

Method used: Compute a more accurate solution using a tighter $TOL = 10^{-10}$ with the updated formular. Then compute another result using the updated formular with $tol = 10^{-5}$ and then compare the results by using the more accurate result as the solution. Find the two errors and compare them.

$$P_n = P_{n-1} - \frac{f(P_{n-1}) \cdot f'(P_{n-1})}{f'(P_{n-1})^2 - f(P_{n-1}) \cdot f''(P_{n-1})} \leftarrow \text{Updated formular.}$$

(b) $P = -2.5$

$P_9 = -1.33434594$

(d) $P_{44} = 3.37354190$ $P_0 = 4$

```
>> a = 1 - 4*x*cos(x) + 2*x^2 + cos(2*x);
>> b = x^2 + 6*x^5 + 9*x^4 - 2*x^3 - 6*x^2 + 1;
>> d = exp(3*x) - 27*x^6 + 27*x^4*exp(x) - 9*x^2*exp(2*x);
```

a

Using the updated Newtons formula to compute a more accurate using tolerance of 10^{-10}. Then proceed to compute another solution using the new formula using the tolerance of 10^{-5}

```
>> ALG022
This is the Fixed-Point Method.
Input the function G(x) in terms of x
For example: cos(x)
 x-((a*diff(a))/(diff(a)^2-a*diff(diff(a))))
Input initial approximation
 0.5
Input tolerance
 10^(-10)
```

← Using tol $= 10^{-10}$

```
Input maximum number of iterations - no decimal point
 50
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
FIXED-POINT METHOD
  I    P
  1    7.21663504e-01
  2    7.39016185e-01
  3    7.39085132e-01
  4    7.39085132e-01

Approximate solution P =   0.73908513
Number of iterations =   4    Tolerance = 1.00000000e-10

>> ALG022
This is the Fixed-Point Method.
Input the function G(x) in terms of x
For example: cos(x)
 x-((a*diff(a))/(diff(a)^2-a*diff(diff(a))))
Input initial approximation
 0.5
Input tolerance
 10^(-5)
```

← Using tol $= 10^{-5}$

```
Input maximum number of iterations - no decimal point
 50
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 1
FIXED-POINT METHOD

Approximate solution P =   0.73908513
Number of iterations =   4    Tolerance = 1.00000000e-05
>> P_tol_5 = P

P_tol_5 =
```

```
    7.390851321662941e-01

>> P_tol_10 = 7.390851321662941e-01;

>> abs(P_tol_10 - P_tol_5)
```

The error using tol $= 10^{-5}$

```
ans =

    0

>> abs(0.739076586 - P_tol_10)
```

The error of the result from problem (2)

```
ans =

    8.546166294087776e-01
```

The error using tol $= 10^{-5}$ is 0. Thus it is more accurate than what we had in problem(2).

6

```
>> ALG022
This is the Fixed-Point Method.
Input the function G(x) in terms of x
For example: cos(x)
 x-((b*diff(b))/(diff(b)^2-b*diff(diff(b))))
Input initial approximation
 -2.5
Input tolerance
 10^(-10)
Input maximum number of iterations - no decimal point
 100
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 1
FIXED-POINT METHOD

Approximate solution P =  -1.33434594
Number of iterations =  54    Tolerance = 1.00000000e-10
>> P_tol_10 = P

P_tol_10 =

   -1.334345940447259e+00
```

← Result using tol $= 10^{-10}$

```
>> ALG022
This is the Fixed-Point Method.
Input the function G(x) in terms of x
For example: cos(x)
 x-((b*diff(b))/(diff(b)^2-b*diff(diff(b))))
Input initial approximation
 -2.5
Input tolerance
 10^(-5)
Input maximum number of iterations - no decimal point
 100
Select output destination
1. Screen
2. Text file
Enter 1 or 2
```

```
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 1
FIXED-POINT METHOD

Approximate solution P =  -1.33434594
Number of iterations =  53    Tolerance = 1.00000000e-05
>> P_tol_5 = P

P_tol_5 =

   -1.334345940447255e+00
```

← Result for tol = 10^{-5}

```
>> error_new = abs(P_tol_10 - P_tol_5);
```

— Error using the New Newton's formula

```
>> error_old = abs(P_tol_10 - -1.33434594);
>> error_new > error_old

ans =

  logical

0
```

= False : This implies using the new formula with tol = 10^{-5} has less error than the old formula.

d

```
>> ALG022
This is the Fixed-Point Method.
Input the function G(x) in terms of x
For example: cos(x)
 x-((d*diff(d))/(diff(d)^2-d*diff(diff(d))))
Input initial approximation
 4
Input tolerance
 10^(-10)
Input maximum number of iterations - no decimal point
 100
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 1
FIXED-POINT METHOD

Approximate solution P =   3.73307681
Number of iterations =  19    Tolerance = 1.00000000e-10
>> P_tol_10 = P

P_tol_10 =

    3.733076807361961e+00

>> ALG022
This is the Fixed-Point Method.
Input the function G(x) in terms of x
For example: cos(x)
 x-((d*diff(d))/(diff(d)^2-d*diff(diff(d))))
Input initial approximation
 4
```

```
Input tolerance
 10^(-5)
Input maximum number of iterations - no decimal point
 50
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 1
FIXED-POINT METHOD

Approximate solution P =   3.73305886
Number of iterations =   4    Tolerance = 1.00000000e-05
>> P_tol_5 = P

P_tol_5 =

    3.733058859993986e+00

>> error_new = abs(P_tol_10 - P_tol_5);
>> error_old = abs(3.37354190 - P_tol_10);
>> error_new > error_old

ans =

  logical

   0
```

False: which implies the old newton's formula has more error than the new one.

(8) (a) We want show that the sequence $P_n = 10^{-2^n}$ converges quadratically to 0.

$$\lim_{n\to\infty} P_n = \lim_{n\to\infty}\left(10^{-2^n}\right) = \lim_{n\to\infty}\left(\frac{1}{10^{2^n}}\right) = 0.$$

Hence $\lim_{n\to\infty} P_n = P \Rightarrow P = 0.$

So $\{P_n\}_{n=0}^{\infty}$ converges to P and $P_n \neq P$, for all n.

If λ and α exist with

$$\lim_{n\to\infty} \frac{|P_{n+1} - P|}{|P_n - P|^{\alpha}} = \lambda, \text{ then } \{P_n\}_{n=0}^{\infty} \to P \text{ of order } \alpha.$$

$$P_n = 10^{-2^n} \Rightarrow P_{n+1} = 10^{-2^{n+1}}$$
$$= 10^{-2^n \cdot 2}$$
$$= \left(10^{-2^n}\right)^2$$
$$= (P_n)^2.$$

$$\Rightarrow \lim_{n\to\infty} \frac{|P_{n+1} - P|}{|P_n - P|^{\alpha}} = \lambda$$

$$\Rightarrow \lim_{n\to\infty} \frac{|P_{n+1} - 0|}{|P_n - 0|^2} = \lim_{n\to\infty} \frac{10^{-2^{n+1}}}{10^{-2^{n+1}}} = 1.$$

$\therefore \lim_{n\to\infty} \frac{|P_{n+1} - P|}{|P_n - P|^2} = 1$, $\alpha = 2$ and $\lambda = 1$.

$\{P_n\}_{n=0}^{\infty}$ converges to p of 2, with asymptotic error constant of 1.

(b) Want to show $P_n = 10^{-n^k}$ does not converge to 0 quadratically, regardless of the size of the exponent $k > 1$.

$P_n = 10^{-n^k}$ Since we want show P_n does not converge quadratically, we will take $\alpha = 2$.

$$P_n^2 = 10^{-2n^k} \quad \text{and} \quad P_{n+1} = 10^{-(n+1)^k}.$$

$$\lim_{n\to\infty} P_n = \lim_{n\to\infty} 10^{-n^k} = \lim_{n\to\infty} \frac{1}{10^{n^k}} = 0.$$

By definition, $\dfrac{|P_{n+1} - P|}{|P_n - P|} = \dfrac{10^{-(n+1)^k} - 0}{10^{-2n^k} - 0} = 10^{-(n+1)^k + 2n^k}$.

$$= 10^{2n^k - (n+1)^k}.$$

$$(n+1)^k = n^k + kn^{k-1} + \frac{k(k-1)}{2!} n^{k-2} + \cdots + 1$$

This give a degree k polynomial and we denote the coefficients $a_0 = 1$, $a_1 = k$ etc.

$$(n+1)^k = n^k + a_1 n^{k-1} + a_2 n^{k-2} + \cdots + 1$$

$$\lim_{n\to\infty} \frac{|P_{n+1} - P|}{|P_n - P|} = \lim_{n\to\infty} 10^{2n^k - (n^k + a_1 n^{k-1} + a_2 n^{k-2} + \cdots + 1)}$$

$$= \lim_{n\to\infty} 10^{n^k - (a_1 n^{k-1} + a_2 n^{k-2} + \cdots + 1)}$$

We denote the degree $k-1$ polynomial in n as $P_{k-1}(n)$

$$\lim_{n\to\infty} 10^{n^k - P_{k-1}(n)}$$

$k \in \mathbb{Z}^+$, so regardless of the value of k we pick, the sequence will not converge. that is

$$\lim_{n\to\infty} 10^{n^k - P_{k-1}(n)} = \infty.$$

Hence P_n cannot converge to 0 quadratically.

(13) $$p_n = g(p_{n-1}) = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})} - \frac{f''(p_{n-1})}{2f'(p_{n-1})}\left[\frac{f(p_{n-1})}{f'(p_{n-1})}\right]^2, \quad n=1,2,3,\cdots$$

and $g'(p) = g''(p) = 0$.

From example 1, $f(x) = e^x - x - 1$. $p_0 = 1$

```
>> g = exp(x) - x - 1;
>> ALG022
This is the Fixed-Point Method.
Input the function G(x) in terms of x
For example: cos(x)
 x - g/diff(g) - diff(diff(g))/(2*diff(g)) * (g/diff(g))^2
Input initial approximation
 1
Input tolerance
 10^(-10)
Input maximum number of iterations - no decimal point
 16
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
FIXED-POINT METHOD
  I   P
  1   4.43756654e-01
  2   1.79303629e-01
  3   6.92878312e-02
  4   2.62852883e-02
  5   9.90029135e-03
  6   3.71874198e-03
  7   1.39539291e-03
  8   5.23394057e-04
  9   1.96289894e-04
 10   7.36111182e-05
 11   2.76045058e-05
 12   1.03517364e-05
 13   3.88191427e-06
 14   1.45571855e-06
 15   5.45855345e-07
 16   2.04660396e-07

Iteration number  16 gave approximation   0.00000020
not within tolerance 1.00000000e-10
```

Using the updated formula to do the iterations and using the order code from class to check the order convergence, the updated formula has order of convergence $\alpha = 3$.

Section 2.5 — Problem 9

problem 9.

```
>> Stefensens
This is Steffensens Method.
Input the function G(x) in terms of x
For example: cos(x)
 x/2 + 3/(2*x)
Input initial approximation
 2
Input tolerance
 10^(-4)
Input maximum number of iterations - no decimal point
 20
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
Select amount of output
1. Answer only
2. All intermediate approximations
Enter 1 or 2
 2
STEFFENSENS METHOD
  I    P
  1  1.73076923e+00
  2  1.73205081e+00
  3  1.73205081e+00

Approximate solution =   1.73205081
Number of iterations =   3 Tolerance =  1.00000000e-04
```

9 Using Steffensen's method, $P_0 = 2$.

$Tol = 10^{-4}$.

$x = \sqrt{3}$, $x^2 = 3$, $x = \frac{3}{x} = x + x - x = \frac{3}{x}$

$= 2x = \frac{3}{x} + x$

$x = \frac{1}{2}\left[\frac{3}{x} + x\right]$

$$\boxed{\therefore \quad g(x) = \frac{x}{2} + \frac{3}{2x}}$$

$x = 1.73205081$.

Comparing with fixed point iteration in problem 11 of section 2.2, using $TOL = 10^{-4}$. The Steffensen's method gave the same result after 3 iterations which implies that the Steffensen's method converges at the speed of the fixed-point iteration method. Also, the Steffensen's method Converges faster than the bisection method where we computered the same result for $\sqrt{3}$ which took more iterations with the same tolerance of 10^{-4}. From Section 2.1 problem 14.

(14) $\{P_n\}_{n=0}^{\infty}$ is said to be Superlinearly Convergent to p if

$$\lim_{n \to \infty} \frac{|P_{n+1} - P|}{|P_n - P|} = 0$$

(a) Want to show that $P_n \to P$ of order α for $\alpha > 1$, then $\{P_n\}$ is superlinearly Convergent to p.

Went we have $\alpha > 1$ and $\lambda \neq 0$,

$$\lim_{n \to \infty} \frac{|P_{n+1} - P|}{|P_n - P|^{\alpha}} = \lambda \quad \text{and} \quad \lim_{n \to \infty} \frac{|P_{n+1} - P|}{|P_n - P|} = 0 \quad \text{so}$$

we will start with the Superlinearly Convergent formula then and manipulate to the get the order formula.

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|} \cdot \frac{|p_n - p|^{\alpha}}{|p_n - p|^{\alpha}} = \lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^{\alpha}} \cdot \frac{|p_n - p|^{\alpha-1}}{1}$$

$$= \underbrace{\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|^{\alpha}}}_{\lambda} \cdot \lim_{n\to\infty} |p_n - p|^{\alpha-1}$$

$$= \lambda \lim_{n\to\infty} |p_n - p|^{\alpha-1}$$

As $n\to\infty$, $p_n \to p$,

$$= \lambda\, |p - p|^{\alpha-1}$$

$$= \lambda\, (0)^{\alpha-1}$$

$$= 0.$$

Thus $\{p_n\}_{n=0}^{\infty}$ is superlinearly convergent to p.

② Want to show $p_n = \frac{1}{n^n}$ is superlinearly convergent to 0.

$$\lim_{n\to\infty} \frac{|p_{n+1} - p|}{|p_n - p|} = 0$$

$$\Rightarrow \lim_{n\to\infty} \frac{\left|\frac{1}{(n+1)^{n+1}} - 0\right|}{\left|\frac{1}{n^n} - 0\right|} = \lim_{n\to\infty} \frac{\left|\frac{1}{(n+1)^{n+1}}\right|}{\left|\frac{1}{n^n}\right|}$$

$$= \lim_{n\to\infty} \frac{n^n}{(n+1)^n (n+1)}$$

$$= \lim_{n\to\infty} \frac{n^n}{(n+1)^n} \cdot \frac{1}{(n+1)}$$

$$= \lim_{n\to\infty} \left(\frac{n}{n+1}\right)^n \left(\frac{1}{n+1}\right)$$

$$= \lim_{n\to\infty} \left(\frac{\cancel{n}}{\cancel{n}\left(1+\frac{1}{n}\right)}\right)^n \lim_{n\to\infty}\left(\frac{1}{n+1}\right)$$

$$= \lim_{n\to\infty} \left(\frac{1}{\left(1+\frac{1}{n}\right)^n}\right) \lim_{n\to\infty}\left(\frac{1}{n+1}\right)$$

$\lim_{n\to\infty}\left(1+\frac{1}{n}\right)^n = e \leftarrow$ Special limit. $\quad \lim_{n\to\infty}\left(\frac{1}{n+1}\right) = 0$

$$= \frac{1}{e}(0)$$

$$= 0.$$

Hence $P_n = \frac{1}{n^n}$ converges to 0 superlinearly. $\square$.

Want to show that $P_n = \frac{1}{n^n}$ does not converge to 0 of order α for any $\alpha > 1$.

$$\lim_{n\to\infty} \frac{|P_{n+1} - P|}{|P_n - P|^\alpha} = \lambda$$

$$\Rightarrow \lim_{n\to\infty} \frac{\left|\frac{1}{(n+1)^{n+1}} - 0\right|}{\left|\frac{1}{n^n} - 0\right|^\alpha} = \lim_{n\to\infty} \frac{\left|\frac{1}{(n+1)^n (n+1)}\right|}{\left|\frac{1}{n^n}\right|^\alpha}$$

$$= \lim_{n\to\infty} \frac{n^{n\alpha}}{(n+1)^n (n+1)}$$

$$\boxed{\begin{aligned}\text{We can write } n\alpha &= n + n\alpha - n \\ &= n + (\alpha-1)n\end{aligned}}$$

$$= \lim_{n\to\infty} \frac{n^n \cdot n^{(\alpha-1)n}}{(n+1)^n (n+1)}$$

$$= \lim_{n\to\infty}\left(\frac{n^n}{(n+1)^n}\right) \cdot \lim_{n\to\infty}\left(\frac{n^{(\alpha-1)n}}{n+1}\right)$$

$$= \lim_{n\to\infty}\left(\frac{\cancel{n}}{\cancel{n}(1+\frac{1}{n})}\right)^n \cdot \lim_{n\to\infty}\left(\frac{n^{(\alpha-1)n}}{n+1}\right)$$

$$= \lim_{n\to\infty} \frac{1}{(1+\frac{1}{n})^n} \lim_{n\to\infty}\left(\frac{n^n}{n+1}\right) \cdot \lim_{n\to\infty}\left(n^{\alpha-1}\right)$$

$$= \frac{1}{e} \cdot \infty$$

$$= \infty \neq \lambda$$

Not an asymptotic error contant.

Thus $P_n = \frac{1}{n^n}$ does not converge to 0 for an $\alpha > 1$.

$\square$

Section 2.6

4.(e).

$$f(x) = 16x^4 + 88x^3 + 159x^2 + 76x - 240$$

$TOL = 10^{-5}$

Choose $P_0 = 0.5$, $P_2 = 1.5$, $P_3 = 2.5$ because Muller's method converge for any initial approximations.

$x_1 = 0.846743.$

* Computation can be on the next page.

>> Mullers
This is Mullers Method.
Input the Polynomial P(x)
For example: to input x^3-2*x+4 enter
 [1 0 -2 4]
 [16 88 159 76 -240]
Input tolerance
 10^(-5)
Input maximum number of iterations - no decimal point
 20
Input the first of three starting values
 0.5
Input the second of three starting values
 1.5
Input the third starting value
 2.5
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
MULLERS METHOD
The output is i, approximation x(i), f(x(i))

The real and imaginary parts of x(i) are
followed by real and imaginary parts of f(x(i)).

3	0.954428	0.000000	67.160830	0.000000
4	0.767903	0.000000	-42.469882	0.000000
5	0.850910	0.000000	2.397193	0.000000
6	0.846751	0.000000	0.005039	0.000000
7	0.846743	0.000000	0.000000	0.000000

4(e)

First real root

Method Succeeds
Approximation is within 1.0000000000e-05
in 7 iterations

>> Mullers
This is Mullers Method.
Input the Polynomial P(x)
For example: to input x^3-2*x+4 enter
 [1 0 -2 4]
 [16 88 159 76 -240]
Input tolerance
 10^(-5)
Input maximum number of iterations - no decimal point
 20
Input the first of three starting values
 -4
Input the second of three starting values
 -3.5

Input the third starting value
 -3
Select output destination
1. Screen
2. Text file
Enter 1 or 2
 1
MULLERS METHOD
The output is i, approximation x(i), f(x(i))

The real and imaginary parts of x(i) are
followed by real and imaginary parts of f(x(i)).

3 -3.367387 0.000000 4.126279 0.000000
4 -3.358176 0.000000 0.057832 0.000000
5 -3.358045 0.000000 0.000053 0.000000
6 -3.358044 0.000000 -0.000000 0.000000

Method Succeeds
Approximation is within 1.0000000000e-05
in 6 iterations

Second real root.

Using Matlab to find the complex roots using double precision.

```
>> syms x
p = 16*x^4 + 88*x^3 + 159*x^2 + 76*x - 240;
R = solve(p,x)

R =

root(z^4 + (11*z^3)/2 + (159*z^2)/16 + (19*z)/4 - 15, z, 1)
root(z^4 + (11*z^3)/2 + (159*z^2)/16 + (19*z)/4 - 15, z, 2)
root(z^4 + (11*z^3)/2 + (159*z^2)/16 + (19*z)/4 - 15, z, 3)
root(z^4 + (11*z^3)/2 + (159*z^2)/16 + (19*z)/4 - 15, z, 4)

>> Rnumeric = vpa(R)

Rnumeric =

                        0.84674257172220062498043996675235
- 1.4943490451576121734267376700522 - 1.7442181428080475010999688717236i
- 1.4943490451576121734267376700522 + 1.7442181428080475010999688717236i
                        -3.3580444814069762781269646266648
```

} Complex roots.