Puneet Grewal
CSC349a - Assignment 3
V00951156

a) $f(x) = \frac{\ln x}{x}$

$$f'(x) = \frac{\frac{d}{dx}[\ln(x)] \cdot x - \ln(x) \cdot \frac{d}{dx}[x]}{x^2}$$

$$= \frac{1/x - \ln(x)}{x^2} \rightarrow \frac{1 - \ln(x)}{x^2}$$

$\tilde{x} = 1.0025$

$$\text{Condition number} = \frac{\tilde{x} f'(\tilde{x})}{f(\tilde{x})}$$

$$= 339.4997919$$

Large condition number proves this function is ill-conditioned.

b) $g(1.003) = 2.986549332 \times 10^{-3}$

$g(1.0025) = 2.490653565 \times 10^{-3}$

$$e_t = \left| \frac{g(x) - g(\tilde{x})}{g(x)} \right|$$

$$= 0.166043052\overline{4}$$

$$e_t \text{ for } x \text{ and } \tilde{x} = \left| \frac{1.0025 - 1.003}{1.0025} \right|$$

$$= 4.98753 \times 10^{-4}$$

The condition number helps us predict this as the output relative error divided by the relative input error is equal to the condition number.

$$\frac{e_t \text{ output}}{e_t \text{ input}} = \frac{0.166043052\overline{4}}{4.98753 \times 10^{-4}}$$

$$= 332.916 \approx \text{Condition Number (339)}$$

The large condition number predicts that a small change in the input results in a large change in the output which is shown by the relative error(s).

Puneet Grewal
CSC349a - Assignment 2
V00951156

a.) $(x-1.5)^4 = 10^{-8}$ $-\sqrt{}$ both sides twice

$\pm(x-1.5)^2 = \pm 10^{-4}$

$\pm(\pm(x-1.5)) = \pm(\pm 10^{-2})$

Roots → 1: $x = 1.5 - 10^{-2}$

2: $x = 1.5 + 10^{-2}$

3: $x = 1.5 + \sqrt{-10^{-2}} = 1.5 + i \cdot 10^{-2}$

4: $x = 1.5 - \sqrt{-10^{-2}} = 1.5 - i \cdot 10^{-2}$

b.) $et = \left| \frac{5.06249999 - 5.0625}{5.0625} \right|$

$= 1.9753 \times 10^{-9}$

largest root from a = $1.5 + 10^{-2}$

$et = \left| \frac{1.51 - 1.5}{1.5} \right|$

$= 0.0066666$

c.) largest relative output error = 0.006666

relative input error = 1.9753×10^{-9}

Due to the largest difference in input and output change, the problem is ill-conditioned. Changing input by an extremely small number such as 10^{-8}, changes output by 0.66% which is relatively large.

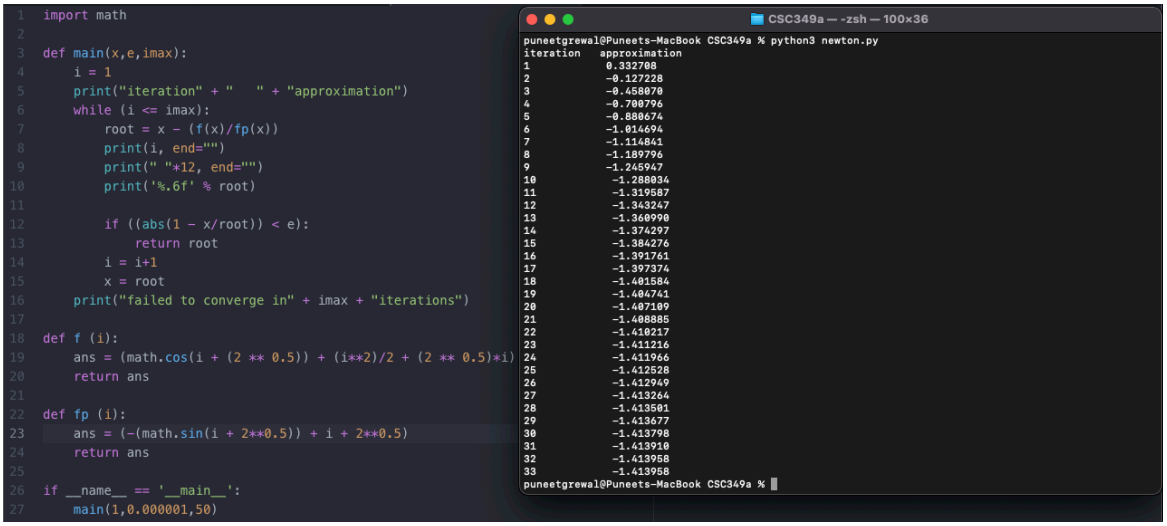
Question 3.

a)

```
import math

def main(x, e, imax):
   i = 1
   print("iteration" + "   " + "approximation")
   while (i <= imax):
      root = x - (f(x)/fp(x))
      print(i, end="")
      print(" "*12, end="")
      print('%.6f' % root)
      if ((abs(1 - x/root)) < e):
         return root
      i = i+1
      x = root
   print("failed to converge in" + imax + "iterations")

def f (i):
   return ans

def fp (i):
   return ans

if __name__ == '__main__':
          main()
```

```
import math

def main(x,e,imax):
   i = 1
   print("iteration" + "   " + "approximation")
   while (i <= imax):
      root = x - (f(x)/fp(x))
      print(i, end="")
      print(" "*12, end="")
      print('%.6f' % root)

      if ((abs(1 - x/root)) < e):
         return root
      i = i+1
      x = root
   print("failed to converge in" + imax + "iterations")

def f (i):
   ans = (math.cos(i + (2 ** 0.5)) + (i**2)/2 + (2 ** 0.5)*i)
   return ans

def fp (i):
   ans = (-(math.sin(i + 2**0.5)) + i + 2**0.5)
   return ans

if __name__ == '__main__':
         main(1,0.000001,50)
```

iteration	approximation
1	0.332708
2	-0.127228
3	-0.458070
4	-0.700796
5	-0.880674
6	-1.014694
7	-1.114841
8	-1.189796
9	-1.245947
10	-1.288034
11	-1.319587
12	-1.343247
13	-1.360990
14	-1.374297
15	-1.384276
16	-1.391761
17	-1.397374
18	-1.401584
19	-1.404741
20	-1.407109
21	-1.408885
22	-1.410217
23	-1.411216
24	-1.411966
25	-1.412528
26	-1.412949
27	-1.413264
28	-1.413501
29	-1.413677
30	-1.413798
31	-1.413910
32	-1.413958
33	-1.413958



```python
import math

def main(x,e,imax):
    i = 1
    print("iteration" + "   " + "approximation")
    while (i <= imax):
        root = x - (f(x)/fp(x))
        print(i, end="")
        print(" "*12, end="")
        print('%.6f' % root)

        if ((abs(1 - x/root)) < e):
            return root
        i = i+1
        x = root
    print("failed to converge in" + imax + "iterations")

def f (i):
    ans = (math.cos(i + (2 ** 0.5)) + (i**2)/2 + (2 ** 0.5)*i)
    return ans

def fp (i):
    ans = (-(math.sin(i + 2**0.5)) + i + 2**0.5)
    return ans

if __name__ == '__main__':
    main(1,0.000001,50)
```

```
CSC349a — -zsh — 100×36
puneetgrewal@Puneets-MacBook CSC349a % python3 newton.py
iteration   approximation
1            0.332708
2            -0.127228
3            -0.458070
4            -0.700796
5            -0.880674
6            -1.014694
7            -1.114841
8            -1.189796
9            -1.245947
10            -1.288034
11            -1.319587
12            -1.343247
13            -1.360990
14            -1.374297
15            -1.384276
16            -1.391761
17            -1.397374
18            -1.401584
19            -1.404741
20            -1.407109
21            -1.408885
22            -1.410217
23            -1.411216
24            -1.411966
25            -1.412528
26            -1.412949
27            -1.413264
28            -1.413501
29            -1.413677
30            -1.413798
31            -1.413910
32            -1.413958
33            -1.413958
puneetgrewal@Puneets-MacBook CSC349a %
```

c)

```
def q3c(root):


    root = round(root, 3)
    ans1 = f(root)
    print("f(xa)=" , ans1)
    ans2 = fp(root)
    print("fp(xa)=" ,ans2)
    true_value = -(2**0.5)

    relative_error = abs((true_value - root)/root)
    print("error=" , relative_error)
```

```
f(xa)= -2.220446049250313e-16
fp(xa)= 1.6233681066069039e-12
error= 0.00015103421010977501
```

```
def q3c(root):


    root = round(root, 3)
    ans1 = f(root)
    print("f(xa)=" , ans1)
    ans2 = fp(root)
    print("fp(xa)=" ,ans2)
    true_value = -(2**0.5)

    relative_error = abs((true_value - root)/root)
    print("error=" , relative_error)
```

```
puneetgrewal@Puneets-MacBook CSC349a % python3 newton3b.py
f(xa)= -2.220446049250313e-16
fp(xa)= 1.6233681066069039e-12
error= 0.00015103421010977501
puneetgrewal@Puneets-MacBook CSC349a %
```

d)
Matplotlib wouldn't print my graph for some reason after much trouble shooting . I have included the file although. After trying to save the plot as a blurry png which I could barely see, I think the approximation is inaccurate as the graph was a big curve.