Name: Chang Jin

1 Problem 1

Running Running the **ls -alF** command on the top level directory of the flash drive:
Output: included in the txt file: *Problem_1_outputs_a.txt*
Running the **ls -alF** command on the top level directory of the most recent backup on the flash drive:
Output: included in the txt file: *Problem_1_outputs_b.txt*

2 Problem 2

Using **diff** command on the python code use to solve Problem 3 and **img.py** we have the output: included in the .txt file: *Problem_2_outputs.txt*

3 Problem 3

The relevant code is included in the file *Problem_3.py*
Steps for graphing the triangle in an image file that is 512 pixels by 512 pixels:

- First, we set the entire image as blue
- Second, set the pixels at the boundary of the image to white with code:

```
#
# set the border pixels to white (requirement 1)
#
borderbottom1 = 0
bordertop1 = int(Y/10)
borderbottom2 = int(Y-Y/10)
bordertop2 = Y
borderbottom3 = 0
bordertop3 = int(X/10)
bordertop4 = X
borderbottom4 = int(X-X/10)

pvals[:,borderbottom1:bordertop1,:] = (0xff,0xff,0xff)  # white
pvals[:,borderbottom2:bordertop2,:] = (0xff,0xff,0xff)  # white
pvals[borderbottom3:bordertop3,:,:] = (0xff,0xff,0xff)  # white
pvals[borderbottom4:bordertop4,:,:] = (0xff,0xff,0xff)  # white
```

- Then set the pixels that is above the line $y = (3/4) \cdot x$ to white with the code (note, we only need to set the red and green plane to 255 because the blue plane is already 255):

```
for j in range(Y):
  for i in range(X):
    if j>i*(3/4):
      pvals[i,j,0] = 0xff  # red
      pvals[i,j,1] = 0xff    # green
```

  this gives us a filled 3-4-5 blue triangle.

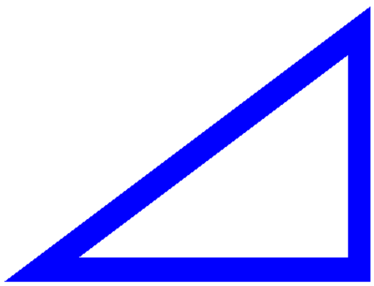
- Next, we need to cut a triangle within the filled 3-4-5 triangle by setting these pixels specified by the code to white:

```
for j in range(Y):
  for i in range(X):
    if j<=(i*(3/4)-Y//15) and (j>(bordertop1+Y//20)) and (i<(Y-(bordertop3+Y//20)))
                                    :
```

```
            pvals[i,j,0] = 0xff   # red
            pvals[i,j,1] = 0xff   # green
```

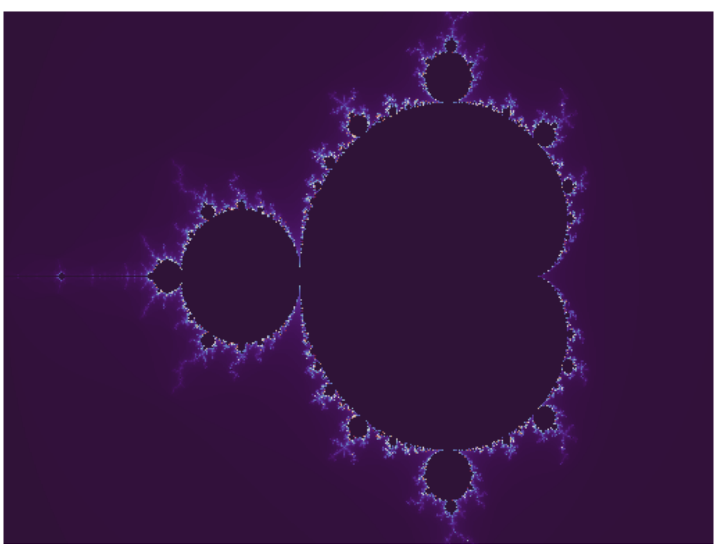
we can adjust the thickness of the triangle by changing the denominators for the parameters $Y//20$ and $Y//15$

the resulting 3-4-5 triangle:



4 Problem 4

From Wikipedia, we see that for c to be in the Mandelbrot set, $\mid z_n^2 + c \mid$ can never exceed 2. Hence we can set the limit to 2.
Using the code included in *Problem_4.py* we can plot the corresponding graph. Note: the image is saved as



Problem4.pdf in the folder.

5 Problem 5

The relevant code is included in *Problem5.py*
Using the equation we obtained from ScienceDirect:

$$PSF(r') = [2\frac{J_1(r')}{r'}]^2$$

where the normalized radius r' is given by

$$r' = \frac{\pi D}{\lambda f} r$$

where

$$\begin{aligned} D &= \text{ aperature diameter} \\ f &= \text{focal length} \\ \lambda &= \text{wavelength of light} \\ J_1 &= \text{Bessel function of the first kind to the first order} \end{aligned}$$

Assuming the focal length of the telescope is $1m$ and we are looking at light in the IR regime with wavelength $\lambda = 700nm$ and the diameter of the lens used is $10m$, then we can plot the normalized peak as where

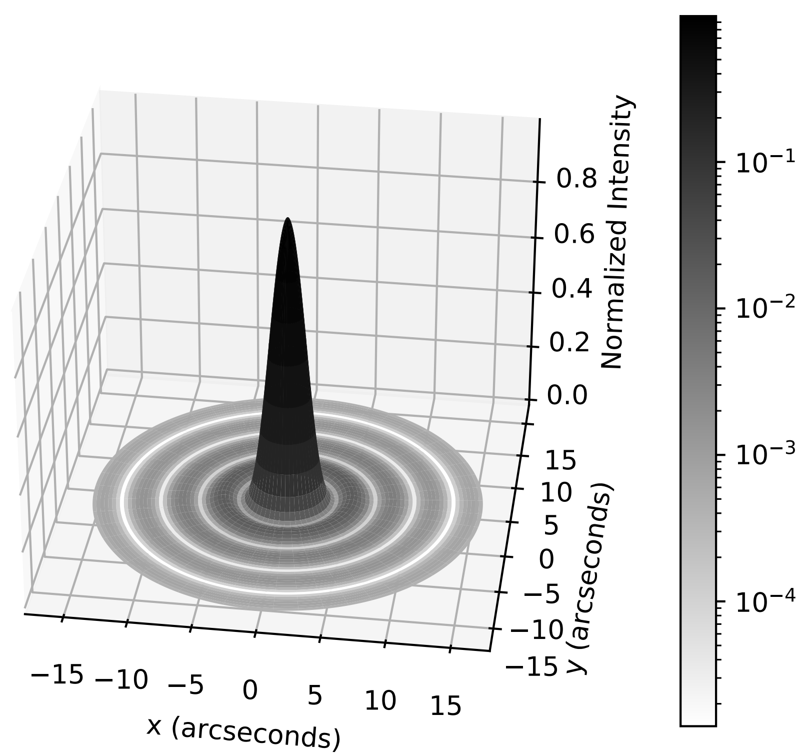


Figure 1: Linear z-scale

$$x = \cos(\theta) r$$
$$y = \sin(\theta) r$$

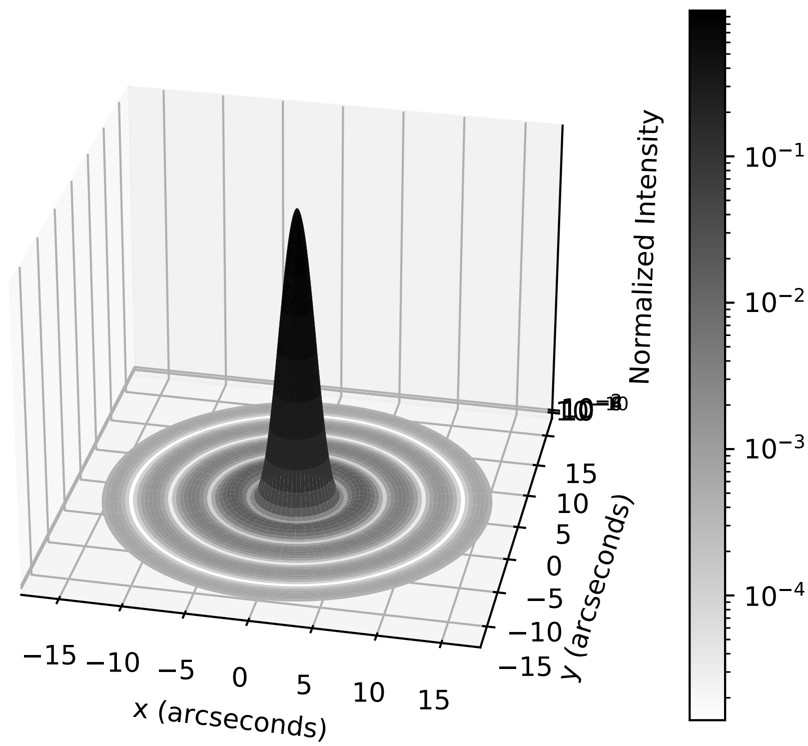


Figure 2: Log z-scale