**Name:** **Registration No:**

**Department of Physics, NITK**

**Even Sem, Jan 2025 - Apr 2025**

**MSc ($2^{nd}$ Year), PH 755, Computational Physics**

**Assignment 1, Dated 28.01.2025. Submit it by 06-02-2025.**

1. Plot the following curves in $x - y$ plane:
   (a) $x^2 + y^2 = 2.0$
   (b) $(x^2 + y^2)^2 = 2x^2y$

2. Plot the following curves as polar plots:
   (a) $r = 3\theta$
   (b) $r = e^{4\theta}$
   (c) $r = e^{\cos(\theta)} - 2\cos(4\theta) + \sin^4(\theta/12)$

3. Itereate the difference equation $z_{i+1} = z_i^2 + c$, where $z$ and $c$ are complex numbers, for 2000 times, starting from $z =$`complex(0,0)` and the real and imaginary part of $c$ varying from -2 to 2 in steps of 0.01.
   The iteration should stop if $z$ diverges (for practical purpose, we take that $z$ is diverging when `abs`$(z) > 2.0$); store the maximum number of iterations one can go before $z$ reaches `abs`$(z) = 2.0$ against corresponding $c$.

   These values can be appended to a list and after all iterations converted to a `Numpy` array. Save this array to a file using the command:
   `numpy.save('my filename.npy')`

   Now write another function to plot this array, using `imshow`.

```
import matplotlib.pyplot as plt
x=numpy.load('my filename.npy')
n=numpy.sqrt(len(x))
n=int(n)
y=x.reshape((n,n))
plt.imshow(y)
```

   The boundaries of this plot can be interesting. See if you can produce those boundaries in higher resolution and zoom there.

4. Now keep $c = 0.285 + 0.01i$, and iterate for different initial conditions $z_0$. Again plot pseudocolor plots of the number of iterations against $\mathrm{Re}(z_0)$ and $\mathrm{Im}(z_0)$. By varying $c$ one can get interesting family of sets. Try this for:
   (a) $c = -0.7269 + 0.1889i$

(b) $c = -0.8 + 0.156i$

(c) $c = -0.4 + 0.6i$