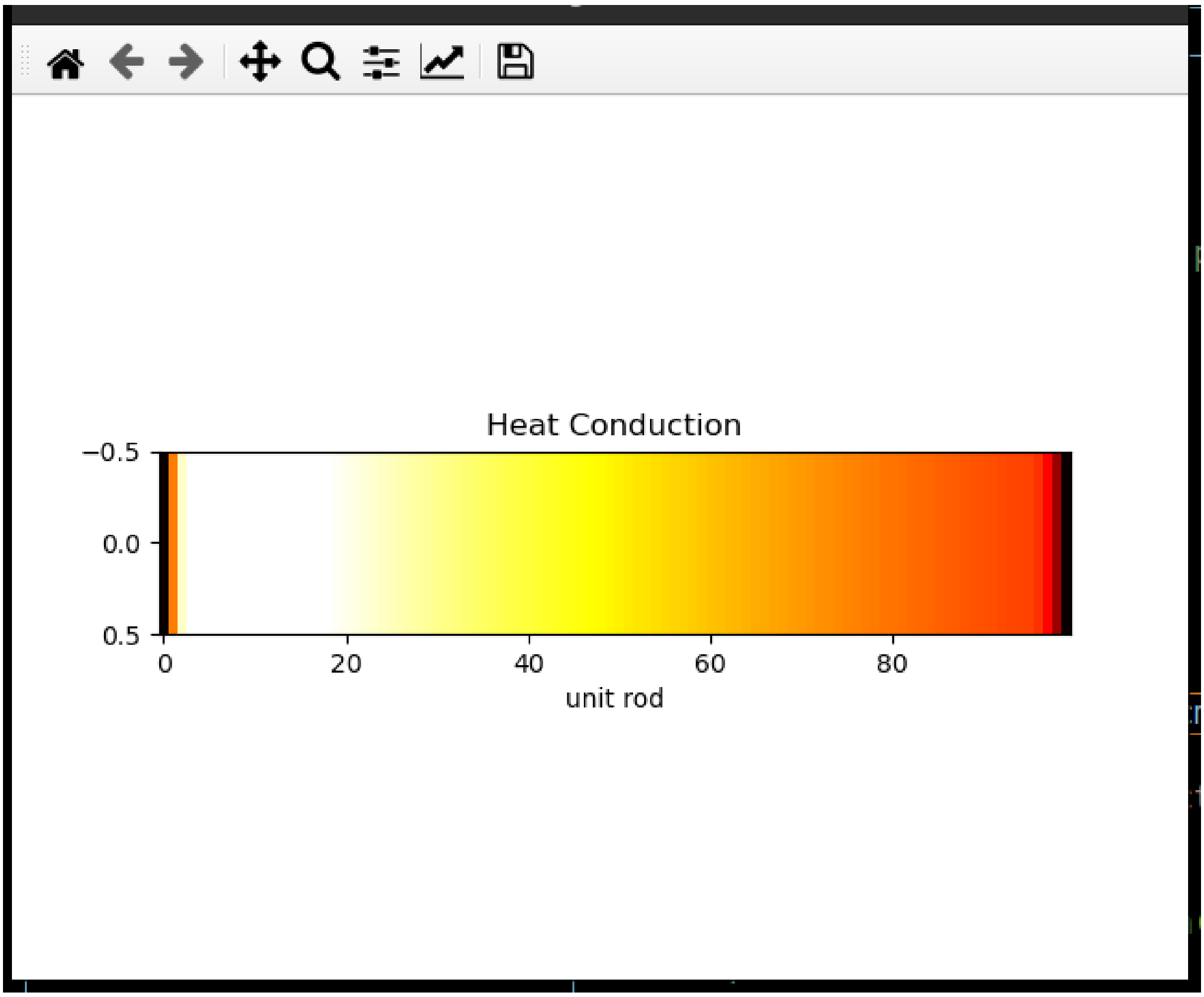
112001041
Assgn 7

LAB REPORT

1)
we generates a temperature distribution at each time step and visualizes the results using an animation with matplotlib. We use Forward Euler to solve the ODE obtained from equation in question.

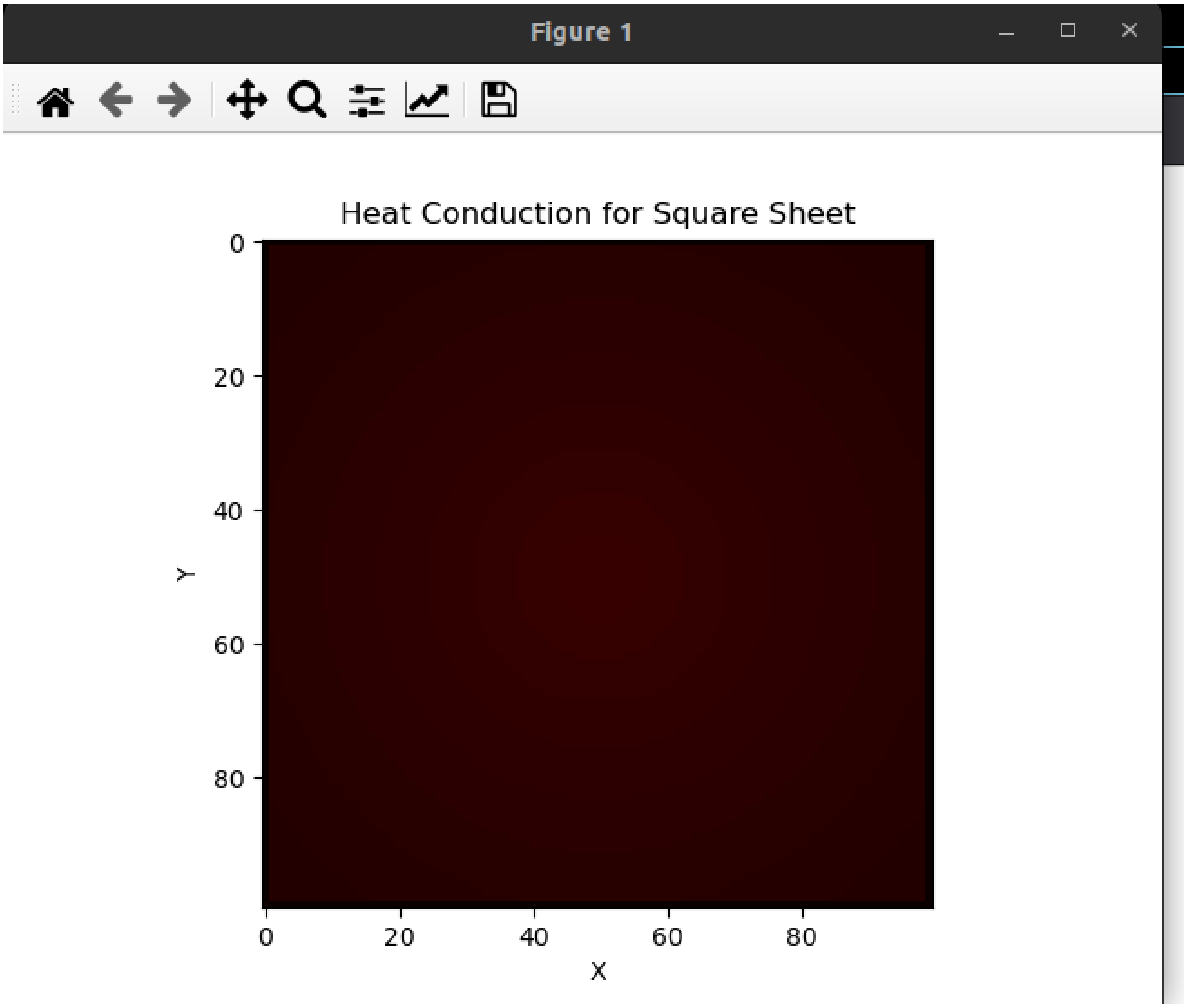
$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2}$$



2)

Its similar to the first question adnd we use forward euler to solve the equation.

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} + \frac{\partial^2 u(x,t)}{\partial y^2} + f(x,y,t)$$



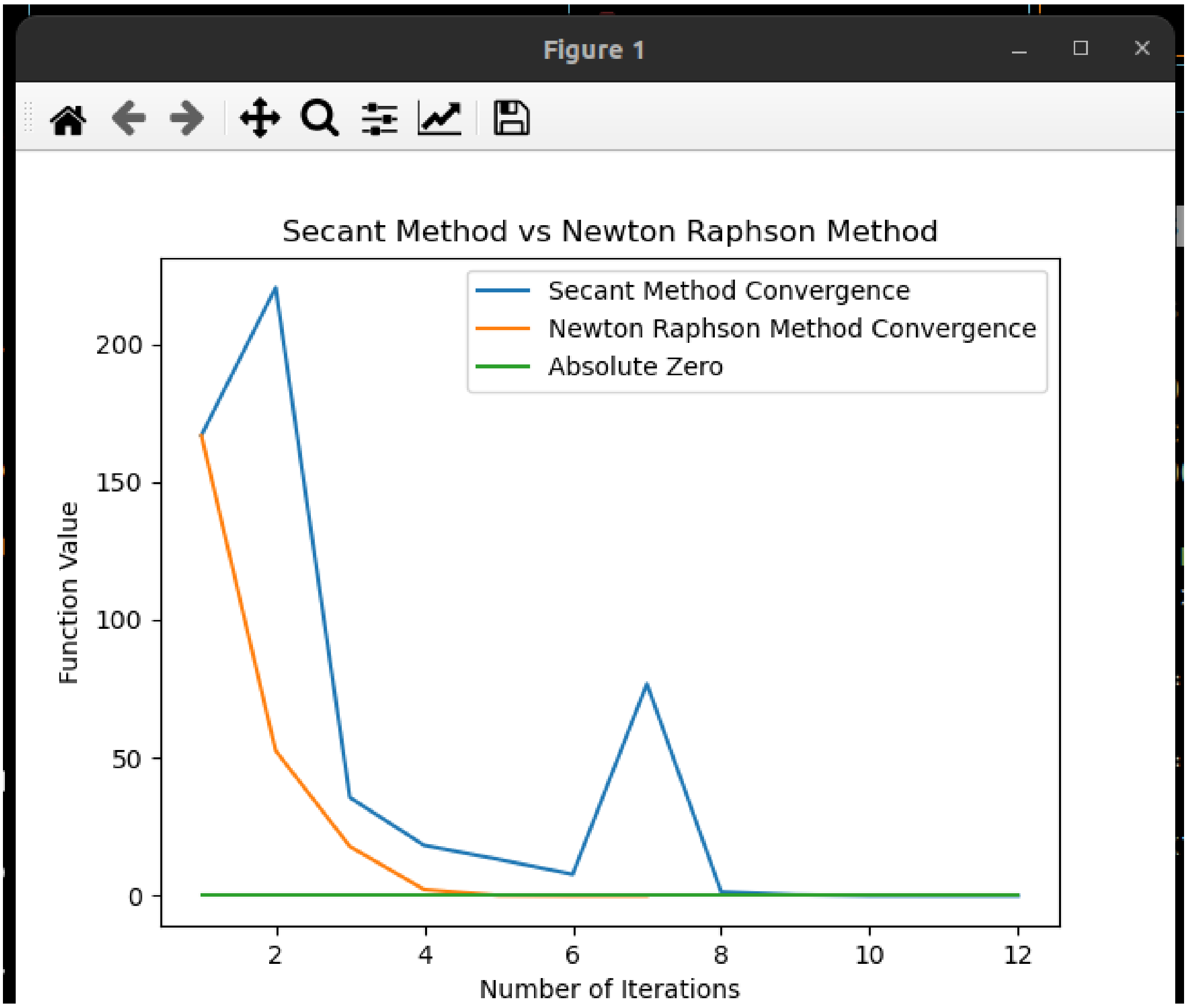
3)

x^n=a

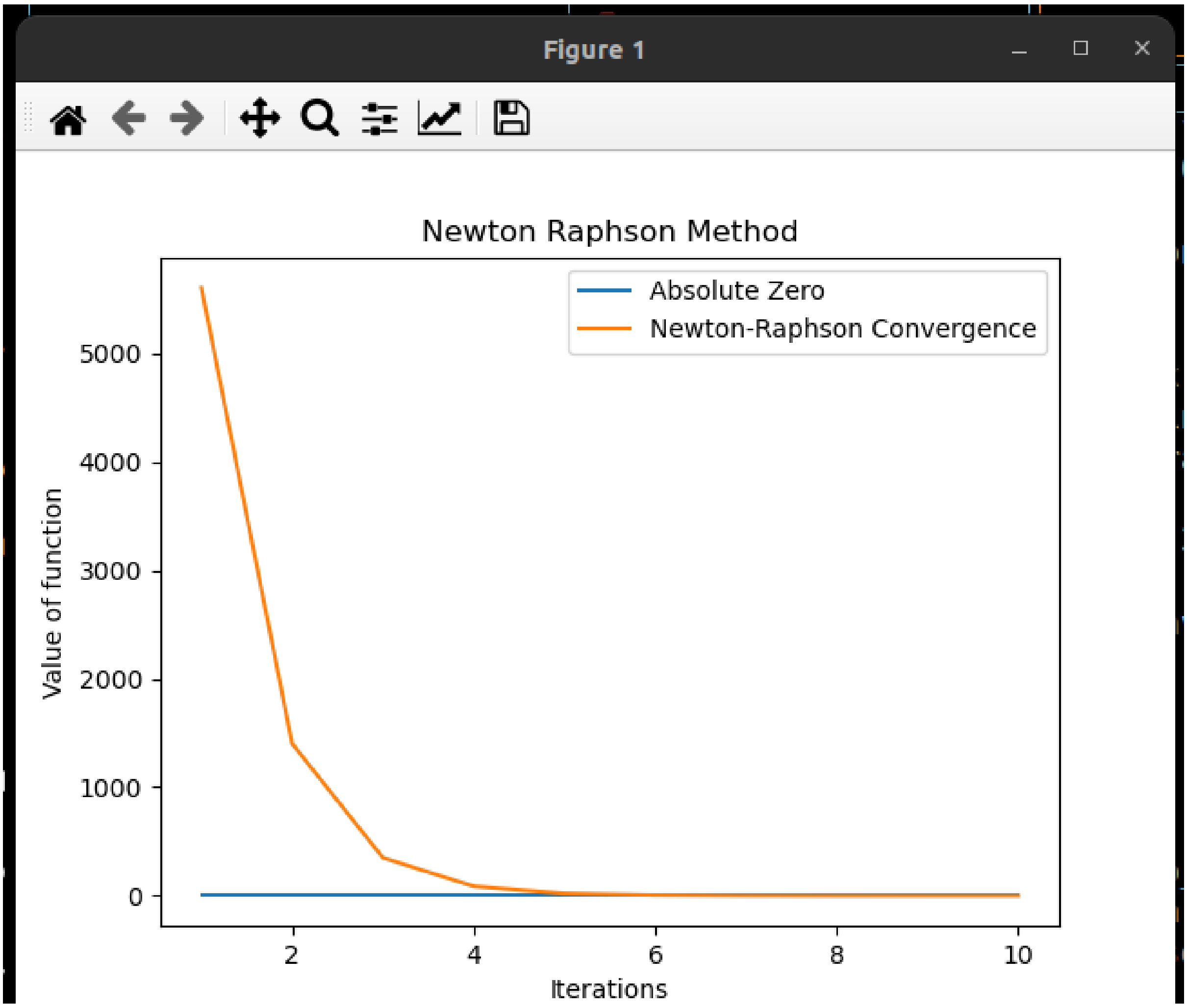
we used the Bisection Method to find the nth root of a given number. The Bisection Method repeatedly divides the interval between a and 1 (where a is the given number) in half until the difference between the high and low points is less than the given error (epsilon).in the worst-case time complexity is proportional to log(1/epsilon).

4)
We need to compare the convergence of the Secant Method and the Newton-Raphson Method for finding the root of a given function .

The function we used is `18*x*x - 5*cos 5*x  + 1`



5)
We used the Newton Raphson iterative relation for multivariate equations for this. We computed the jacobian matrix and its inverse and multiply it with function value.



6)
We used the alberth method to find the roots and can see that roots found by Aberth Method are extremely close to the actual roots of the equation.

```
libGL error: failed to load driver: swrast
(base) sid@sid-HP-Spectre-x360-Convertible-13-aw0xxx:~/Documents/cma/week7$ python3 q6.py
The roots of the polynomial are: [-3.0, 0.9999246522428408, 1.000075341874753, 4.000000000000001, 5.000000000000001]
```

7)
We used the method added in previous question to find root of any continuous function. In our case we test our method by taking continuous method as cos(x) in range [-3,1]. For this we need to convert the continuous function into polynomial by best_fit method.

```
8.545846246133304e-07
-1.2596213676954501e-07
The roots of the polynomial: [-1.5707963294905318]
```