भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Mathematical Methods for Engineers (ME5010)

Term Project

Fractals through Newton's iterative Method

Submitted to:
Dr. Niranjan S Ghaisas
Course Instructor ME5010
Assistant Professor
Dept. of Mech. and Aerospace Engg.
IIT, Hyderabad

Submitted by:
Pushkar Mishra (ME22RESCH11007)
Vipin Kumar Yadav (ME22RESCH01001)
Dheeraj Kailas Valecha (CE22RESCH01006)
Shikshit Nawani (ME22MTECH12004)
Thathagat Sarangi (ME22RESCH11010)

Contents

List of Figures

Chapter 1

Introduction

The fractals are geometrical patterns of infinite repeatability. These patterns vary with scale and similar structures can be obtained upon changing the scale size. This property of showing similar patterns with increasing small scales is called self-similarity. The fractal geometry derives its roots from the measure theory of mathematics. It differs from the topological geometries by the way they scale. The fractal geometry can be visualised by plotting the roots of the polynomial, using different color palettes. Newton Raphson method can be used to find roots of a polynomial. A more stable overall strategy is used to discover a better starting estimation of the solution if the residual does not decrease as expected. Stabilization of Newtons' method using variable change operations is a different strategy [1]. For some applications, such a strategy might be stable, but because of the possibility of local minima that could cause convergence to stall, good convergence is not assured.

Attraction basins for various roots of a nonlinear problem can have fractal borders in a Newton technique applied globally. This introduces orbits into the history of convergence that can severely inhibit convergence in addition to the potential singularities of the problem. Given that the fractal structure is still not fully understood, this is presented as evidence that Newton's technique behaves in unpredictable fashion. However, it has been demonstrated for the complex cubic that once the underlying fractal problem is understood, one can explain convergence behaviour in Newton's approach [2].

In the past, Schroeder was the first to address the issue of using Newton's method with complex polynomials in 1871. He inquired which from both roots an complex arbitrary plane beginning point would converge given a quadratic equation $(z^2 - 1)$. The imaginary axis, which he discovered right away, provided the answer to this query regarding the edge of the attraction basins. Cayley posed the identical query for a symmetric cubic $(z^3 - 1)$ in 1879 but was unable to provide a response. The query came to be known as Cayley's issue [3]. Gaston Julia utilised this issue as an illustration for explaining sets that eventually took on his name in a significant study. The union of all points that ultimately map to a single point is known as the Julia set. Later on, a precise definition is provided. A reflecting symmetry in relation to the real axis, invariance while rotating by $2n$-multiples, & presence of $z = 0$ & $z = \infty$ as mirror images are some further aspects of the set that Julia deduced while addressing Cayley's issue. Additionally, the set's fragmented nature was emphasised in the sense that Jordan curves were unable to accurately represent it. Cayley's issue, however, did not receive a full response. Nearly seven decades later, there was a resurgence of interest in linked characteristics of iterated polynomial mappings, however the Mandelbrot set received most of the focus, while Newton's method was primarily discussed historically and as an approachable example to explain Julia sets [5]. There have been some attempts to reduce the amount of iterations needed for convergence from any starting point and to demonstrate convergence in a mathematical sense for the numerical application of Newton's approach. A thorough solution to Cayley's issue, however, was still lacking. The research mentioned above focused on Julia set general qualities instead on particular Newton's method properties. A first thorough Cayley's problem answer was provided by altering this strategy and investigating the process through which Newton's approach creates fractal structures, identifying all key scale factors and symmetries, and integrating these results to give the structure's fractal dimension. The fractal structure can be approximated using Jordan curves alone without the use of complex computer graphics [4].

Chapter 2

Numerical Methodology

2.1 Complex Plane Analysis Using Numerical Iteration

Newton's methodology used for solving the equation $f(z) = 0$ brings out the estimation of the roots of the solution from iteration. Now lets take into account the iterated function.

$$z_{k+1} = \lfloor F(z_k), k = 1, 2, 3... \rfloor \tag{2.1}$$

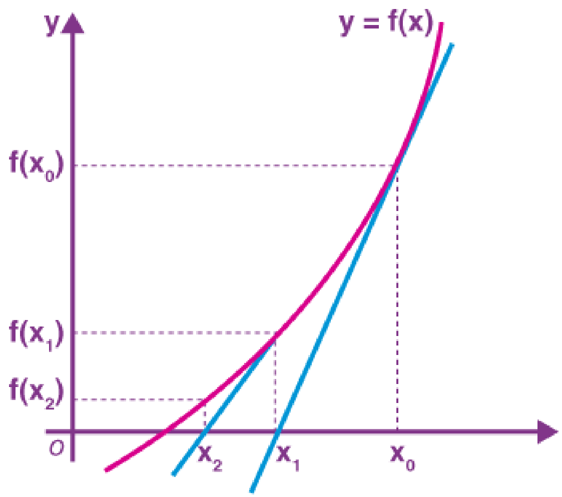


Figure 2.1: Newton Raphson method strategy to find roots

Depending on the coordinates of z_0, the sequence z_0,z_1,z_2,... may converge or diverge for each initial point z_0. Newton's method is a discrete dynamical system which is used to find out the roots at a specific location on the curve of a higher-order nonlinear function, This method is used to obtain root with better precision on succeeding iteration round .

Let $f : C \Longrightarrow C$ be a function with a complex value and a continuous derivative In the complex plane, Newton's iteration method is given as:

$$N(z) = \left\lfloor z_{n+1} = z_n - \frac{f(z_n)}{f'(z_n)} \right\rfloor , \tag{2.2}$$

where n = 0,1,2,3,... and $f'(z_n) \neq 0$

$N : C \cup (\infty) \implies C \cup (\infty)$ and for the initial value which is known as seed z_o travels to the maximum amount of iterations to an approximation of the root To put it another way, if z_0 is the seed, then on the root of the solution will converge on $f(z)$ where n is defined as a maximum iteration numbers]. The complex derivative of the rational function of $N(z)$, its complex derivative is defined as

$$N'(z) = \left\lfloor \frac{f(z)f''(z)}{(f'(z))^2} \right\rfloor \tag{2.3}$$

2.2 Newton Fractal's symmetric Order μ

As was already said, it is generally known that the borders of the basins of attraction for various polynomial roots are fractals. We will start by providing the definitions that are required for a technical discussion. Properties for the corresponding fractals for a an assortment of polynomials will be quantified and presented.

2.2.1 Definitions

unique class of polynomials are surfed in this subsection that are defined in the following way.

The order of symmetric polynomial μ is defined by

$$z^\mu = 1, z \in \boldsymbol{C}, \mu \in \boldsymbol{N}, \mu > 2. \tag{2.4}$$

When the roots of symmetric polynomials are identified using Newton's,We'll examine the areas that draw those roots. The standard definition of Newton's technique will be the focus of this report.

From the Newton method the iteration is defined on $f(z)$ as.

$$z^{(n+1)} = z^{(n)} - \frac{f(z^{(n)})}{f_z(z^{(n)})} = g(z^{(n)}) \tag{2.5}$$

from the equation $z^{(n)}$ is denoting the n^{th} iterate, and f_z, states $\frac{df}{dz}$.

Roots z_n from the definition, of $f(z)$ with $f(z_n) = 0$ are denoted as a stable fixed points in the Newton iteration method.

The Newton approach will be viewed in a slightly different light than implied by (2.5). Instead of determining the next iteration given a beginning point, we will inquire as to whose points $z^{(n)} = z$ are given by mapping $z^{(n+1)} = z_0$ by (2.5).

The order μ complex Newton polynomial is defined as

$$(\mu - 1)z^{\mu} - \mu z_0 z^{\mu-1} + 1 = 0, z, z_0 \in \boldsymbol{C} \tag{2.6}$$

Despite the fact that fractals have a "common-sense concept," it might be challenging to define them in general terms.Mandelbrot defined fractals as a set for which the topological dimension exceeds the Hausdorff-Besicovitch dimension. Even if it is exact, this definition is difficult to use in reality if the Hausdorff-Besicovitch dimension is elusive or even unknowable, similar to how many popular fractals work.

The order of these fractal μ is defined by all the mapped point which are combined into a origin which is singular in nature by the help of Newton mapping (2.6).

This definition makes it clear that since z is a part of the Newton fractal, z_0 must also be a part of it. It is important to keep in mind that the fractal is not made up of the actual basins of attraction, but rather the a union of edges that are points.

We shall make reference to the Julia set theory's conceptual underpinnings since this investigation is focused on analysis of how many iteration a complex variable can have. We define the requisite sets in the manner described by Falconer.

The closure of the complez-variable function which is given by Julia set $J(g)$ is a sets of a complez-variable function g is the closure of the set of resisting recurring points of g.The Julia set $J(g)$. $F(g)$ is a fatou set $F(g)$ of a functions called complex -variable which is also defined by g i.e. stable sets.

The combination of every Julia in the mapping would thus be the equivalent definition of the Newton fractal using this method.To identify the fractal structures order of the Julia points is a important.

On the fractals the specific julia point's order is given as the iterations function one needs to reach to the origin point.

2.3 General properties

We may now offer specific study for inverse mapping in general (2.6) & newly discovered Newton fractal after laying groundwork. Shortly, the essential conclusions are drawn; for a more thorough discussion including the example of $\mu = 3$.

2.3.1 Classical result

From (2.4), it is obvious that the μ^{th}-order (symmetric) polynomial's roots are

$$z_n = e^{\frac{2\pi}{\mu}(n-1)i}, n = 1, 2, ..., \mu. \tag{2.7}$$

When derivative in equation (2.5) singularizes, the main Julia point is always

$$\chi_0 = 0. \tag{2.8}$$

2.3.2 Parent structure

From (2.6), it is clear that the inverse Newton iteration applied to the origin yields first-order Julia points once are

$$\chi_{1,n} = \frac{1}{\sqrt[\mu]{\mu - 1}}.e^{\frac{\pi}{\mu}(2n-1)i}, n = 1, 2, ..., \mu. \tag{2.9}$$

As a result, the μ^{th} order Newton fractal's distance from the origin is

$$r = \frac{1}{\sqrt[\mu]{\mu - 1}}. \tag{2.10}$$

reaching 1 as μ reaches large value. It is clear from comparing (2.7) and (2.9) that first-order Julia points are always positioned among 2 adjacent roots.

These points establish the parent structure, which, as we'll see later, contains all the data required to describe the fractal. Since the points in the parent structure have radii greater than r, attractive circle notion, which proved a very helpful approximation for $\mu = 3$, cannot be extended for general μ. All ideas related to the alluring circle, however, are universal.

2.3.3 Global symmetry

Inspection reveals that under the coordinate transformation, (2.6) is invariant.

$$z \mapsto ze^{\frac{2\pi m}{\mu}i}, z_0 \mapsto z_0 e^{\frac{2\pi m}{\mu}i}, m \in \boldsymbol{N}. \tag{2.11}$$

Continually rotating by $\frac{2\pi}{\mu}$ only with origin acting as centre are the equivalent to this. Additionally, it is evident since (2.6) is unchanging as

$$z \mapsto (z)^*, z_0 \mapsto ((z_0))^*, \tag{2.12}$$

with the conjugate of z denoted by z^*. Branches of the fractal must be aligned along straight lines because of rotating and reflecting symmetry interaction.

As a result, the Newton fractal of order μ contains μ branches and a μ-fold global rotational symmetry. Additionally, it has symmetry of the reflection with regards to real axis. The μ-shaped branches are arranged in a line.

2.3.4 Global scale factor

The governing equation (2.6), with increasing z, z_0 $((|z|), (|z_0|) \gg 1)$, could be broadened to

$$z_0 = \frac{\mu - 1}{\mu} z + \mathrm{O}(z^{1-\mu}), \tag{2.13}$$

Under scaling, the fractal is unchanging since it contains both z and z_0.

$$z \mapsto \lambda_{\mu,\infty} z, \lambda_{\mu,\infty} = \frac{\mu}{\mu - 1}. \tag{2.14}$$

It must be highlighted that the limit $|z| \mapsto \infty$ is the only one where this perfect invariance is possible. It may, however, be utilised much sooner as a rough estimate. Therefore, we claim that the Newton fractal's global scale factor is of order μ is $\lambda_{\mu,\infty}$.

2.3.5 Local symmetry

The local symmetry can be obtained followed by global symmetry.

$$\omega_n = \frac{\pi}{\mu(\mu - 1)} + \frac{2\pi}{\mu(\mu - 1)} n, n = 0, 1, 2, ..., \mu(\mu - 1) - 1. \tag{2.15}$$

The angles $\omega_n = 0$ and $\omega_n = \pi$ never exist, as can be observed. According to this formula's interpretation, the fractal's $\mu(\mu - 1)$ identical local branches move in the direction of the origin at angle

$$\theta_\mu = \frac{2\pi}{\mu(\mu - 1)} \tag{2.16}$$

between neighbouring branches.

An integer multiple of local symmetry defines the global symmetry, as can be seen. This trait is shared by all Newton fractals. We observe that the local symmetry persists across the entire fractal because every Julia point represents the origin locally. In summary, Newton fractal of order μ demonstrates $\mu(\mu-1)$-fold local symmetry of rotation on every point and locally consistent to θ_μ rotations.

2.3.6 Local scale factor

With the help of the preexisting properties, we could indeed state that the parent structure of such fractal will consist of a blob surrounding two fractal chains and with a variety of chains flowing through it, with the overall number of chains for every blob to be $(\mu - 1)$ as a result of the symmetry findings. There are scale factors called cross-chain scale factors $\Lambda\mu, n$ that regulate the local scaling whenever a point changes from one branch to the next during a Newton step in addition to every chain holding a scale factor of $\lambda\mu$.

For scale factor for every cross-chain involved, general expressions cannot be made. We could provide the scale factor's expression which are connected to the fixed points of cycle μ .

$$\Lambda_{\mu,2,n} = \frac{1}{2(\mu-1)\sin\frac{\pi n}{\mu}}, n = 1, 2, ..., \mu - 1 \tag{2.17}$$

We may provide a justification for amount of various scale factors for every cross-chain that should occur, despite not being able to say them clearly. It should be observed that the $(\mu - 1)$ branches that consists of blobs of Newton-fractal ordered μ are symmetrically oriented around the blob's centre line. One of it's branches and centre line are parallel if μ is even. The blobs on each branch are further divided into $(\mu - 1)$ sub-branches, which is important for the following stage of the inverse mapping. As a result, there are $(\mu - 1)^2$ overall sub-branches. Here, we consider the centre line's symmetry, or the feature that for even μ, one branch (the centre) is symmetrical by itself. This restricts the range of possible sub-branches to

$$b_\mu = \left\lfloor \frac{1}{2}\left[(\mu-1)^2 + 1\right] \right\rfloor, \tag{2.18}$$

which each should have a scale factor for every cross-chain attached. The operator for truncating integers is indicated by $\lfloor . \rfloor$.

These results for the origin at all fractal locations can be generalised once more because of the local conformity of the inverse Newton mapping. The n^{th} ordered Julia points move geometrically toward the $(n-1)^{th}$ ordered Julia point with a local scale factor λ_μ on a branch. Different cross-chain scale factors $\Lambda_{\mu,2,n}$ and scale factors for every cross-chain are $\left\lfloor \frac{1}{2}(\mu - 1) \right\rfloor$ for cycles of μ and b_μ.

2.4 Algorithm for computing

Python was used as a programming language to compute and analyse the faractals, and the libraries used were Numpy and matplotlib.The Algorithm for the computed was as follows.

- **Step - 1** define separate functions stating the given polynomials and its derivative in this case $z^n - 1$ where the input arguments to the function are the point at which it has to be evaluated and the bvalue of n for which the fractal has to be generated.

- **Step - 2** The function `newton_root_finding` with arguments n, and the domain for which the function has to be evaluated are provided

- **Step - 3** N by N meshgrid is constructed, which giverns the no. Of divisions in X and Y axis. Suppose the grid size is 2500 by 2500 and the range is between $[-2, 2]$ the X and Y axis will have 2500 division between $[-2, 2]$ respectively.

- **Step - 4** For every gridpoint a complex function "$x + iy$" is formed which serves as the initial guess for the Newton Raphson method.

- **Step - 5** `max_iter` is set to be 50 (later which is updated to be 100 so more points converge to a root.

- **Step - 6** Itterative Newton method is applied by

  $$z^{(n+1)} = z^{(n)} - \frac{f(z_n)}{f'(z_n)} \tag{2.19}$$

  where $z = x + iy$

- **Step - 7** actual roots of the polynomial are calculated by passing a list of coefficients to the function `np.roots(list_of_coefficients_of_the_polynomial)`

- **Step - 8** After performing Newton iteration every starting point is assigned a label according to the root at which it converges and newton fractals are plotted as per those labels.

- **Step - 9** Function `fractal_plot` plots the newton fractal.

Chapter 3

Results and discussion

This study is focused on visualizing fractal pattern for polynomials of different degrees.

3.1 Third degree polynomial

This function has two real roots $+1$ and -1 there is only one demarcation line between the roots as shown in the figure below, so we may comment that we may obtain so called "beautiful" fractals only in the case when we have a polynomial of degree greater than or equal to 3.

Figure 3.1: Newton fractal generated for $z^2 - 1$

As it can be seen that if there are two complex roots then the complex plane is equally divided into two regions. It can be expected that something similar happens with the function of degree 3 ($z^3 - 1$) that the complex plane will be divided in three regions where iterates of initial points in each region converge to one of the three roots.

Now let's see what the behavior is for the function $f(z) = z^3 - 1$. This function has three roots $z = 1, z = \frac{-1}{2} \pm i\left(\frac{\sqrt{3}}{2}\right)$. The basins of attraction can be seen in figure.

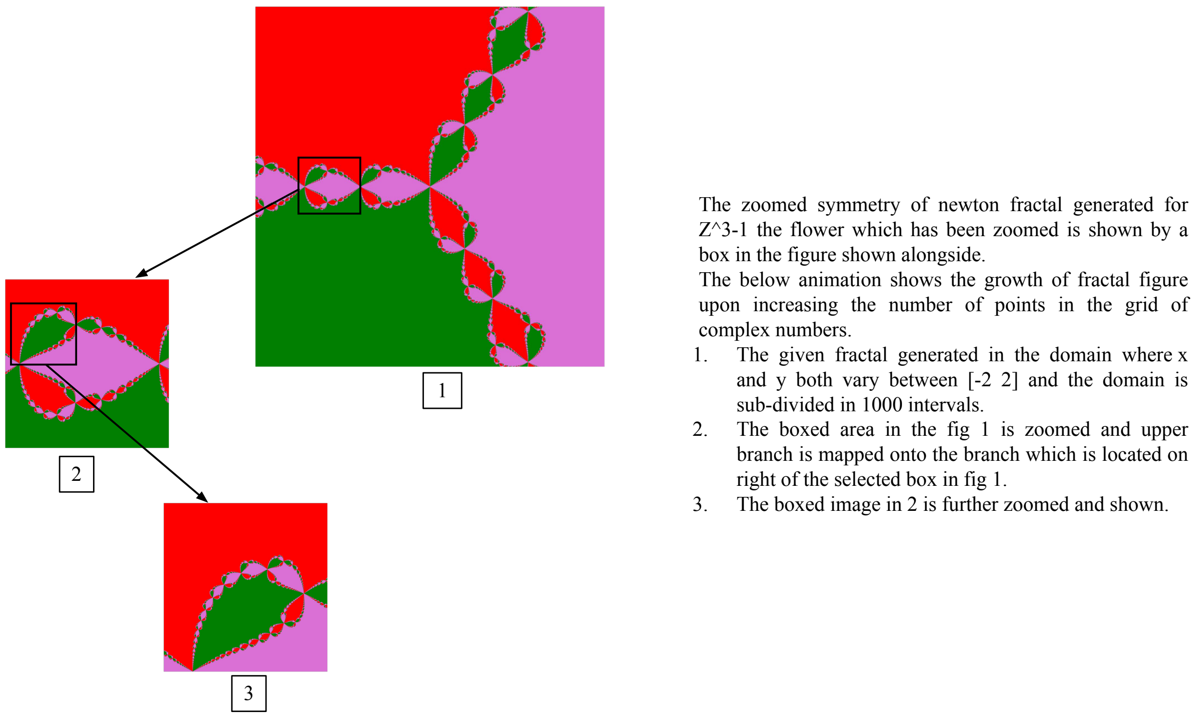


Figure 3.2: Newton fractal for $z^3 - 1$

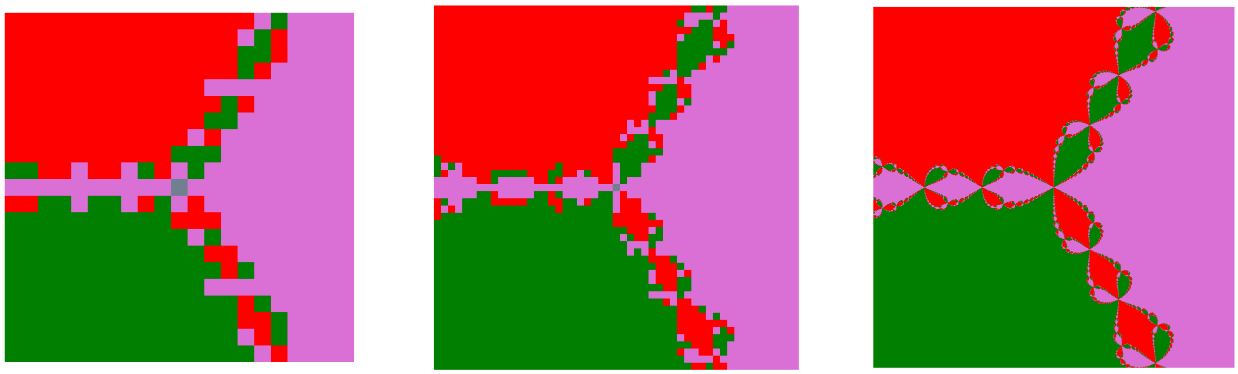
The zooming symmetry and explanation of flower and to the area at which they map has been shown in the above figure. the procedure adopted for construction of the above figure is described below

1. At first the grid was divided into N by N and these each cells in N by N matrix are considered to be as initial guesses for Newton's method.

2. Each grid is colored as per the destination(root reached).

3. There are 4 possibilities in case of a cubic equation either the initial guess would converge to a given root or it would diver from a given root, each of this possibilities is given a different colour.

4. On further refining the mesh we can obtain a ”beautiful” fractal.

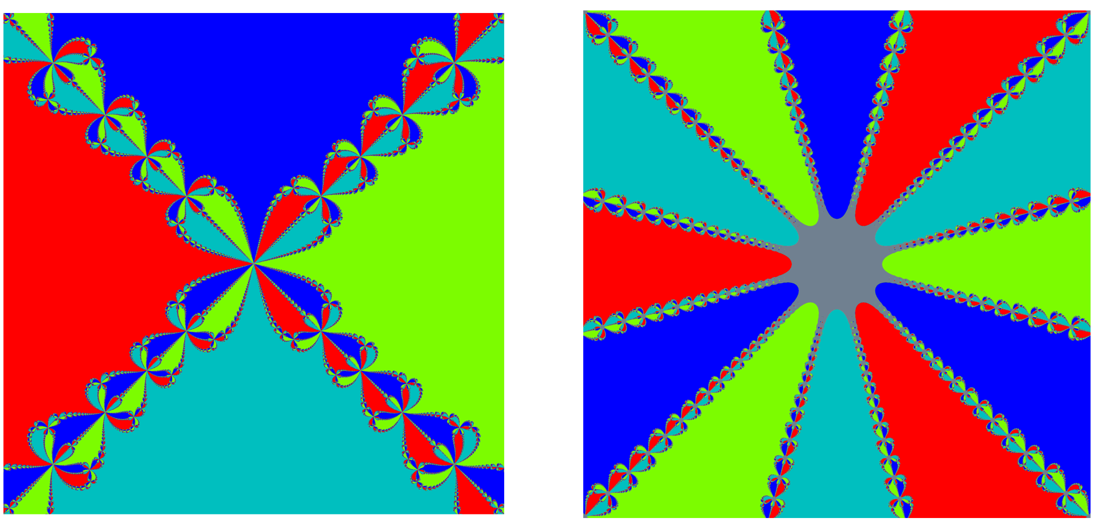


(a) 10 by 10 grid (b) 20 by 20 grid (c) 1000 by 1000 grid

Figure 3.3: Newton fractal generated in range [-2,2] for different grid sizes

3.2 Fourth degree polynomial

The function $f(z) = z^4 - 1$, which has two real and two complex roots all lie on a unit circle with origin at the center of the complex plane. The newton fractal thus obtained is given below.

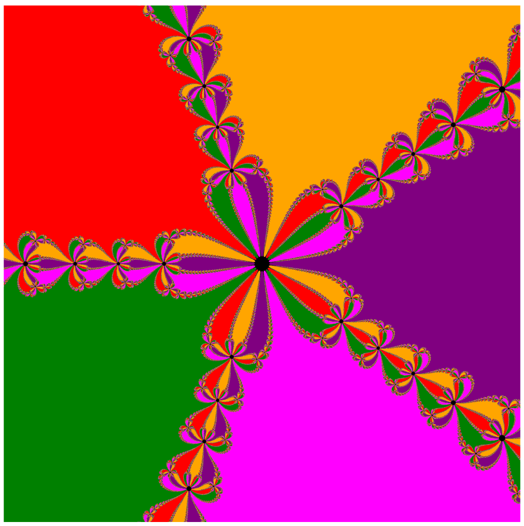


(a) in domain [-2,2] (b) in domain [-0.02,0.02]

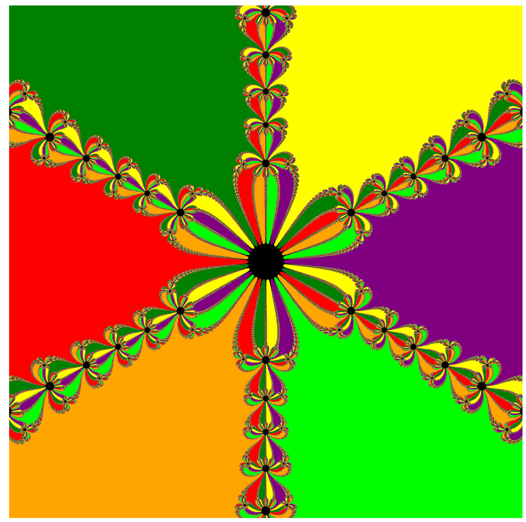
Figure 3.4: Fractal for $z^4 - 1$ with different zooming scales for a 2500 by 2500 grid

3.3 Fractals for some more values of n in $z^n - 1$

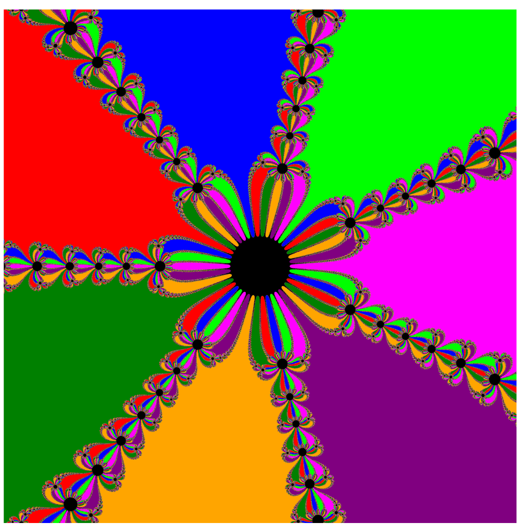
Moving further some more fractals were generated according to the polynomial $z^n - 1$ for different values of n, keeping maximum number of iterations as constant. It can be observed that as the value of n increases, the size of escape point increases, which means we would be requiring more number of iterations to increase at a particular root.



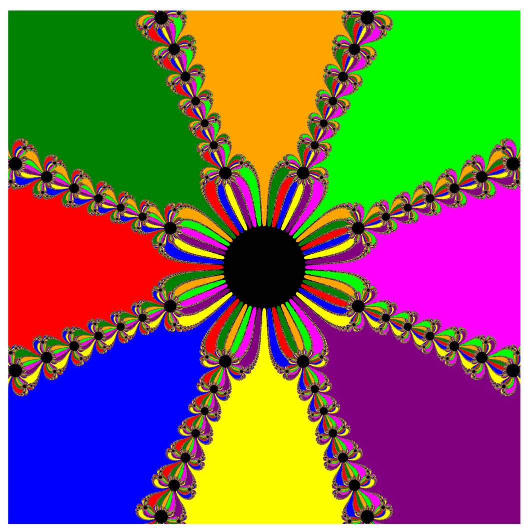
(a) n = 5



(b) n = 6



(c) n = 7

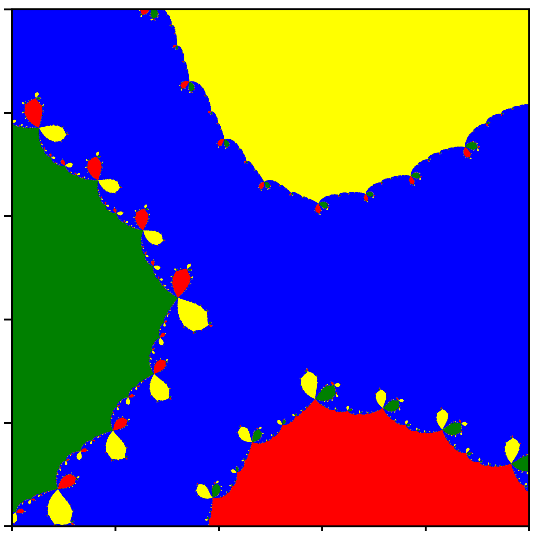


(d) n = 8

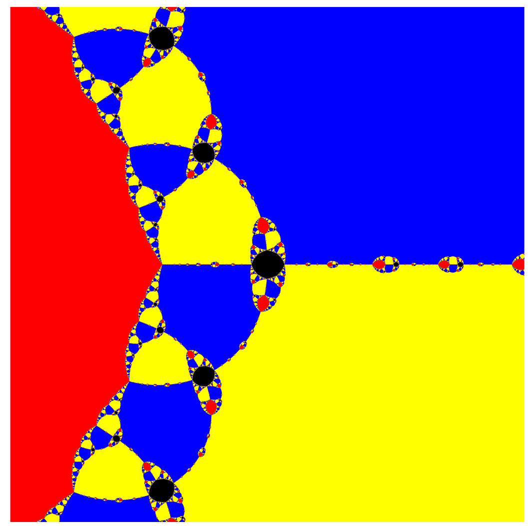
Figure 3.5: Fractals for z^n in range [-2,2] with a 2500 by 2500 grid with n varying from 5 to 8

3.4 Fractals for different polynomials

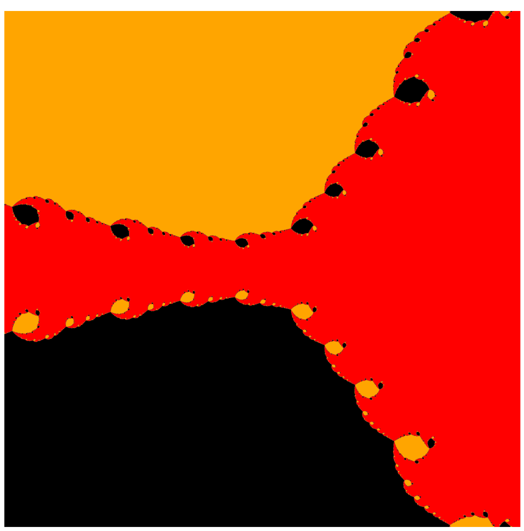
Using the same code in python the following fractals were generated for different polynomials, some of which are shown below:



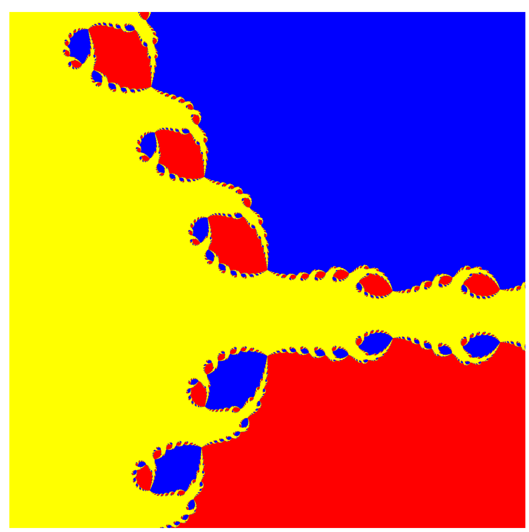
(a) $z^4 + iz^3 + 2z - 1$



(b) $z^3 - 2z + 2$



(c) $z^3 - z^2 + z - 1$



(d) $z^3 + iz^2 + 2$

Figure 3.6: Fractals for different polynomials in range [-2,2] with a 2500 by 2500 grid

Chapter 4

Conclusion

The following inferences can be drawn from the project work that was carried out for ME5010:

- Newton Raphson when applied for complex number (Z = x+iy) generates fractals with rotation and zoom symmetry.
- When talking for polynomial $z^n - 1$ the roots lie on the boundary of a unit circle.
- For each root the angle that is spanned by each arm is $\frac{2\pi}{n}$.
- Number of iterations play a crucial role in convergence of roots, whereas fractals can also be generated based on number of iterations required to converge.
- There seems to be a significant relation between value of **n** and number of iterations needed for converging to a particular root.
- The fractals generated can be zoomed in infinitely by varying the scale which was decided during the starting phase.
- Upon increasing the number of iteration the "black" spots (which do not converge to a root) reduce in size, so we may conclude that we can iterate infinitely to completely remove those small spots.
- These black spots are known as escape points and a set of all those escape points is known as **julia set**.

Author contribution

The authors confirm their contribution to the project as follows:

- **Study conception** and **code design**: Pushkar, Vipin and Shikshit
- **Data collection**: Dheeraj, Vipin, Tathagat, Pushkar
- **Analysis** and **interpretation of results**: Pushkar, Shikshit
- **Report preparation**: Tathagat, Dheeraj, Shikshit
- **Presentation**: Vipin, Pushkar
- All authors reviewed the results and approved the final version of the report.

Bibliography

[1] M. Abromowitz and I. Stegun, *Handbook of mathematical functions dover*, 1965.

[2] K. Falconer, "Fractal geometry john wiley & sons," *Inc., Chichester*, 1990.

[3] P. Henrici, *Applied and computational complex analysis, Volume 3: Discrete Fourier analysis, Cauchy integrals, construction of conformal maps, univalent functions.* John Wiley & Sons, 1993, vol. 41.

[4] D. Knoll and P. McHugh, "An inexact newton algorithm for solving the tokamak edge plasma fluid equations on a multiply-connected domain," *Journal of Computational Physics*, vol. 116, no. 2, pp. 281–291, 1995.

[5] M. Drexler, I. Sobey, and C. Bracher, "Fractal characteristics of newton's method on polynomials," 1996.

project_fractal

November 16, 2022

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
```

[ ]:
```
def f(Z,n):
    return((Z**n)-1) #defining function Z^n-1
```

[ ]:
```
def f_dash(Z,n):
    return(n*Z**(n-1)) #defining derivative of function Z^n-1
```

[ ]:
```
def newton_root_finding(n,start_x,end_x,start_y,end_y,n_pts): #implementing Newton Raphson
    x=np.linspace(start_x,end_x,n_pts)
    y=np.linspace(start_y,end_y,n_pts)
    X,Y=np.meshgrid(x,y)

    Z=X+1j*Y
    max_iter=50
    for i in range (1,max_iter):
        try:
            Z=Z-np.divide(f(Z,n),f_dash(Z,n))

        except:
            next
    return(Z)
```

[ ]:
```
def fractal_plot(n,start_x,end_x,start_y,end_y,n_pts): #plotting fractal as per root reached
    a = [0]*(n+1)
    a[0] = 1
    a[-1]=-1
    r=np.roots(a)
    converged_Z=newton_root_finding(n,start_x,end_x,start_y,end_y,n_pts)
    eps=0.000000001;
    z=[]
```

```
    for i in range (len(r)):
        condi=abs(converged_Z-r[i])<eps                                                    ␣
 ↪#classifying on basis of root reached
        z.append(condi.astype(int))
    tmp=z[0]
    for i in range(1,len(z)):
        tmp=np.logical_or(tmp,z[i])
    tmp=np.logical_not(tmp)
    pp1=z[0]
    for i in range (1,len(z)):
        pp1=pp1+z[i]*(i+1)
    pp1=pp1+(len(z)+1)*tmp
    cmap =␣
 ↪ListedColormap(['red','green','blue','yellow','orange','magenta','lime','purple','black'])
    plt.imshow(pp1, cmap=cmap, origin='lower')
    plt.axis('off')
    plt.savefig('final n='+str(n)+'.png',format='png',dpi=1500)
    return [r,pp1]
```

[ ]:
```
start_x = start_y = -2
end_x = end_y = 2;
n_pts = 2501
# n_s = [3,4,5,6,7,8]       #uncomment for getting fractals for these values
n_s=[3]
for n in n_s:
    a=fractal_plot(n,start_x,end_x,start_y,end_y,n_pts);
```