# Evaluating Newton's Method to find Real and Complex roots of Various Functions

By
Manu Sharma
Shubham Sharma
Rounak Jangir
Prashant Verma
Aditya Prakash
Akhilesh Kumar

April 17, 2014

## Abstract

In our exploratory project, we evaluated Newtons method (Newton Raphson Method) for finding out real and complex roots of polynomial functions, and analysed the resulting Newton basins of Attraction. A Newton basin is the set of points (initial guesses) which converges to a particular solution after certain Newton iterations. We took different examples of polynomial functions, calculated their roots (both real and complex) using Newton's method and evaluated the resulting Newton basins. We also explored the different applications and limitations of Newton's method .

## Introduction

---

Our Project involves the exploration and use of Newton Raphsons method and its properties. Newton's Method is based on simple idea of linear approximation. This procedure can be interpreted geometrically,also. Newton's method is based on local information,namely f(x)and f'(x). We used Newton's method to evaluate the real and complex roots of polynomial functions and then analysed the resulting Newton basins of Attraction. We experimented with various polynomial functions. We also studied and explored the various applications of Newton's method. Using the high level language OCTAVE which is used for mathematical computation, we were able to compute the Newton basins and were able to appreciate the Newton's method and the role and power of computation in Mathematics.

# ⋆ Newton's Method ⋆

In order to obtain the roots of various polynomials, we can use Newton's Method. We use Newton's method to approximate the solution (both real and complex) of f(x)=0 where $f$ is a polynomial function.
Newtons method is a numerical and iterative algorithm for finding the zeros (or roots) of a function (of the form f(x)=0). It is a easy method to understand and is very efficient and powerful technique in approximation of roots.

Let f(x) be a good function and $p$ be a root of a function.We start with an estimate $x_0$ which is close to $p$. Now, by using $x_0$, we produce an improved approximation (or we think we improved) $x_1$. From $x_1$, we produce another estimate $x_2$. We go on further and further, until we get really close to $p$ or it becomes clear that we are getting nowhere.
The Newton-Raphson method works really good if the initial 'guess' is very close to root for a well defined function. But, it can go drastically wrong in other cases.

Now, let $x_0$ be very close to the root $p$. Then, suppose p=$x_0$+h. Here, $h$ shows how far the guess is from the root. Since, h is infinitesimally small, so we can apply linear approximation here,

0=f(p)=f($x_0$+h)≈f($x_0$)+ hf'($x_0$)

If f'($x_0$)!=0

h $\approx$ -$\frac{f(x_0)}{f'(x_0)}$

Therefore, p= $x_0$+h $\approx$ $x_0$ -$\frac{f(x_0)}{f'(x_0)}$

Our better approximation becomes
$x_1$= $x_0$ -$\frac{f(x_0)}{f'(x_0)}$

On generalising,we have

$x_{n+1}$= $x_n$ -$\frac{f(x_n)}{f'(x_n)}$

## Geometric Interpretation

In Newton's Method, we start with an initial guess that is close to a root. We then approximate the function using it's tangent line and then the x-intercept is computed. This x-intercept will be a better approximation of the root of the equation rather than the original guess. The method is repeated again with the x-intercept. This method is repeated again and again until the ultimate solution is obtained (or a solution which is very close to the real root). More iterations will result in a more closer solution to the real/complex root.
There is a catch, though. First, if an equation has multiple solutions (and most equations that would require Newton's Method will have multiple solutions), Newton's Method will only give one of the solutions for a given guess. To find the others, one must run Newton's Method with different guesses closer to the other solutions. This would require, for example, graphing the equation to see about where the solutions are, and then using those as guesses.

Suppose f : [a, b] → R is a differentiable function defined on [a, b] → R. Formula for the Newton's methods can be derived very easily. Suppose we have a initial guess $x_n$. Then we can derive a better approximation. The equation of the tangent to the curve y = f(x) at the point $x = x_n$ is

$$y = f'(x_n)(x - x_n) + f(x_n)$$

where f'(x) denotes the derivative of the function *f*. The x-intercept of this line (distance of the point from the origin where this line cuts the x-axis i.e. y=0) is then used as the next approximation of the root, $x_{n+1}$. Replacing x by $x_{n+1}$ and setting y=0, we get,

$$0 = f'(x_n)(x_{n+1} - x_n) + f(x_n)$$

solving for $x_{n+1}$ we get,

$$x_{n+1}=x_n - \frac{f(x_n)}{f'(x_n)}$$

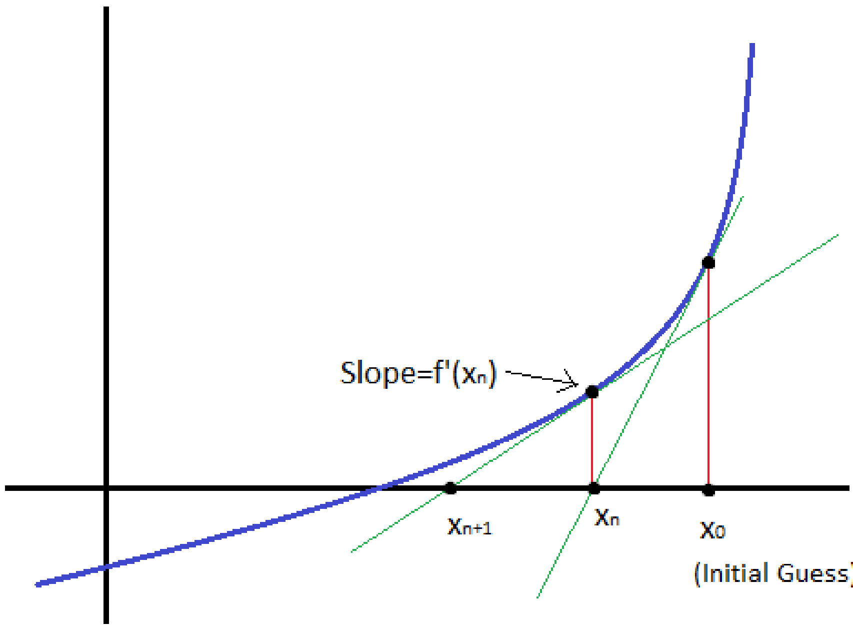


Figure 1: Iteration of Newton's Method

We start the process with some arbitrary value $x_0$. The closer to the root of the equation we are, the better. But in case of no initial idea, any value can be taken. This method usually converges for the guesses closer to the roots. But some exceptions are there where Newton's method doesn't converge. They will be discussed later. The set comprising of all the initial guesses for which there is a particular solution having the Newton's method converging on it is called a Newton Basin. The set comprising of all the initial guesses for which, within a specified number of iterations the Newton's method does not converge is called the Newton Wasteland.

The geometric interpretation of complex case is analogous to the real case. $x_{n+1}=x_n - \dfrac{f(x_n)}{f'(x_n)}$

This equation can be substituted for the complex function as

$z_{n+1}=z_n - \dfrac{f(z_n)}{f'(z_n)}$

The roots of a curve can be found by the following method

- First the initial guess $Z_k$ is taken and the value of the function is found of the curve.

- Then,the equation of the tangent plane $T_k$ is computed on that point found on the curve.

- The intersection of the plane $T_k$ and the x-y plane gives the equation of a line $L_k$.

- A perpendicular is drawn from the initial guess $Z_k$ to the line $L_k$ obtained in the previous step.

- The foot of perpendicular thus obtained gives the better approximation of the root of the complex function.

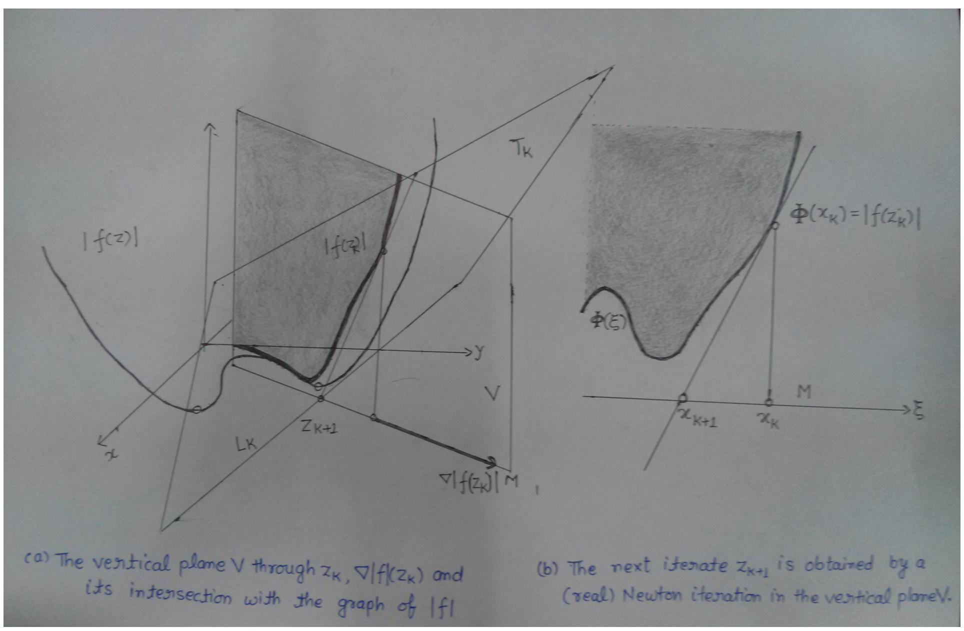


Figure 2: A 'side view' of Newton Complex iteration

The gradient of the function at $Z_k$ is given by the opposite direction of the $Z_{k+1}$-$Z_k$.

## Examples

### 1. *f(x)*=$x^2 - 16$=*0*

Suppose we want to solve for the given function f(x)=$x^2 - 16 = 0$. The graph of the function is given below:

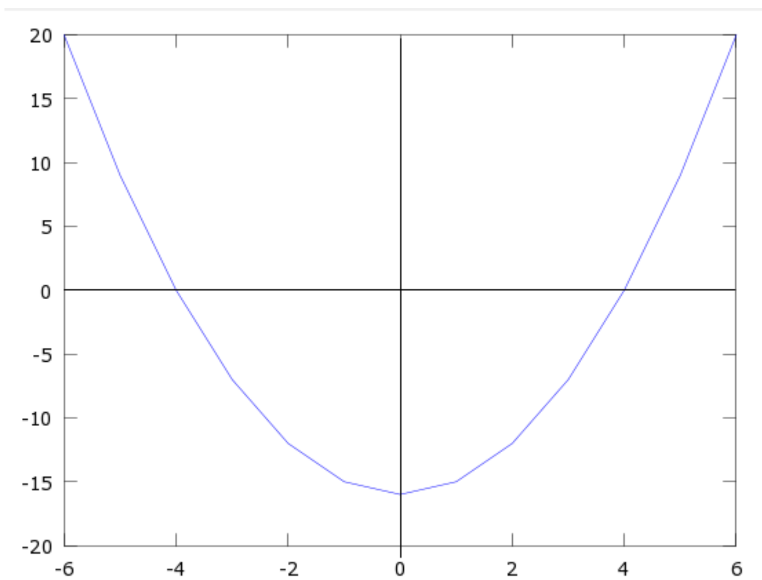


Figure 3: Polynomial $x_2 - 16 = 0$

From the above figure, we can easily make out that the roots of function $f$ are 4 and -4. Since
f(x)=$x^2 - 16 = 0$
=(x-4)(x+4)=0
=±4

Now, using Newtons method to solve, we iteratively use formula derived for the Newton's Method to approximate x such that f(x)=0. Let us take $x^0 = 5$. We then apply Newton's method to obtain the closer approximation.

$x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$ f(x)=$x^2 - 16$ f'(x)=2x

$$\Rightarrow \;\; x_1 = 5 - \frac{f(5)}{f'(5)}$$

$$\Rightarrow \;\; x_1 = 5 - \frac{9}{10} \qquad\qquad \Rightarrow x_1 = 4.1$$

Similarly now for $x_2$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

$$\Rightarrow \;\; x_2 = 4.1 - \frac{f(4.1)}{f'(4.1)}$$

$$\Rightarrow \;\; x_2 = 4.1 - \frac{0.81}{8.2} \qquad\qquad \Rightarrow x_1 = 4.001$$

which is a much better approximation than $x_1$. Similarly, by repeating Newton's Method iteratively, we will eventually find the root of the function or a much better approximation.

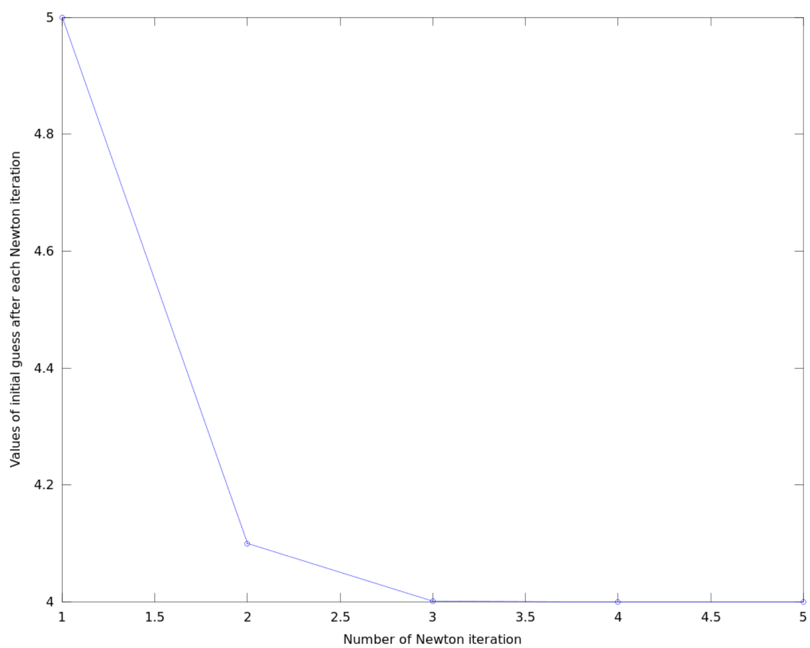


Figure 4: Iteration of Newton's method for $x_0$ =5

Above is the figure showing how Newton's method iterates and converges towards one of the root of the polynomial.

## 2. *f(x)*=$x^2 - 4x + 4$=*0*

Suppose we want to solve for the given function f(x)=$x^2 - 4x + 4$=0. The graph of the function is given below:

From the above figure, we can easily make out that the root of function $f$ is 2. Root 2 have a multiplicity of 2. Since

f(x)=$x^2 - 4x + 4$=0

=$x^2 - 2x - 2x + 4 = 0$

=x(x-2)-2(x-2)=0

1 =(x-2)(x-2)

=2

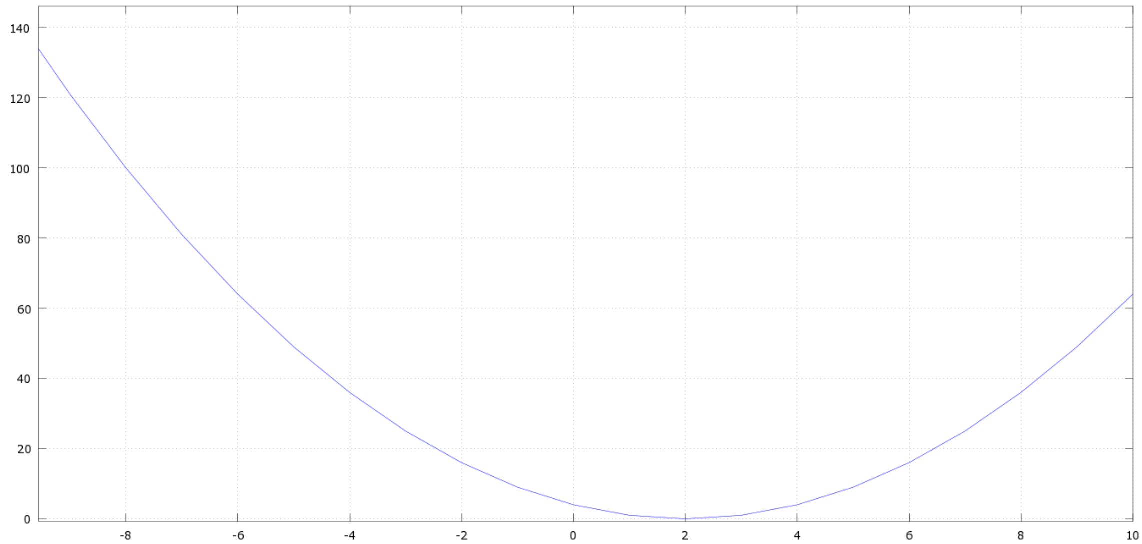


Figure 5: Polynomial $x^2 - 4x + 4$=0

Now, using Newtons method to solve, we iteratively use formula derived for the Newton's Method to approximate x such that f(x)=0. Let us take $x^0 = 3$. We then apply Newton's method to obtain the closer approximation.

$x_1 = x_0 - \dfrac{f(x_0)}{f'(x_0)}$ f(x)=$x^2 - 4x + 4$ f'(x)=2x-4

$\Rightarrow \ x_1 = 3 - \dfrac{f(3)}{f'(3)}$

$\Rightarrow \ x_1 = 3 - \dfrac{1}{2}$ $\Rightarrow x_1 = 2.5$

Similarly now for $x_2$

$x_2 = x_1 - \dfrac{f(x_1)}{f'(x_1)}$

$\Rightarrow \ x_2 = 2.5 - \dfrac{f(2.5)}{f'(2.5)}$

$\Rightarrow \ x_2 = 2.5 - \dfrac{0.25}{1}$ $\Rightarrow \ x_2 = 2.25$

$x_3 = 2.125$

$x_4 = 2.06353$

$x_3 = 2.03177$

which is a much better approximation than $x_4$. Similarly, by repeating Newton's Method iteratively, we will eventually find the root of the function after 23 iterations at 2.

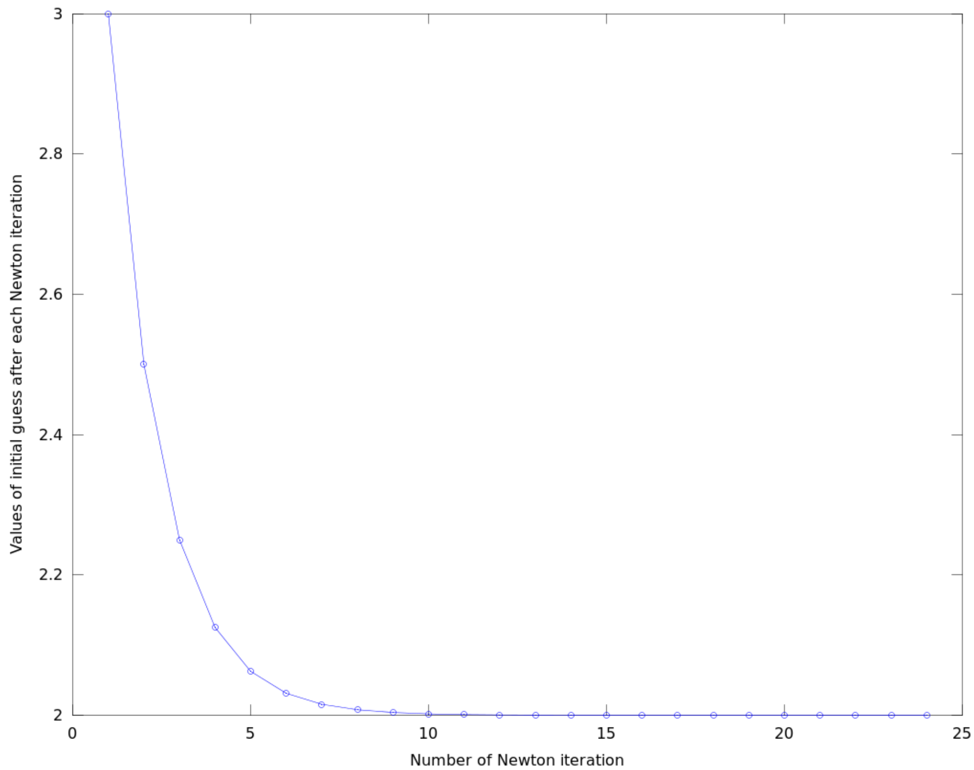


Figure 6: Polynomial $x^2 - 4x + 4$=0

## Limitations of Newton's method

### Bad Starting Point

This kind of limitation happens when the choice of starting point is wrong. Sometimes, a good function satisfies all the conditions necessary for convergence but the starting point is chosen wrong. This happens because most probably that point doesn't lie in the interval where the function converges.

1. **Iterative point is stationary**
   If a stationary point of a function is encountered (i.e. the value of f'(x)=0), then the method will give the next iteration as infinity. It means, the method will give a wrong result. For eg.

   Consider the function:
   f(x)= $4-x^2$
   The solutions of the function f(x)=0 are at x = ±2. Let us take $x_0 = 0$ as our starting point. for f(x)= $4-x^2$, f'(x) = 2x
   Therefore,
   $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$
   Now for $x_0 = 0$, $f'(x_0) = 0$,
   $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0 - \frac{4}{0} \approx \infty$

   The same issue will occur if instead of this point, any other iterative point is chosen which is stationary. Hence, in this example, 0 is the stationary point for this function f(x).

2. **Starting point enters a cycle**
   In some cases, the starting points sometimes enter a infinite cycle, where
   $x_k{=}x_{k+p}$ where p is natural number.
   i.e. The values of newton's iterations repeats themselves after a certain interval of iterations. For eg.

   Consider a function:
   f(x)= $x^3 - 2x + 2$
   Therefore, f'(x)= 3x − 2

Let $x_0 = 0$. Therefore,

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = x_1 = 0 - \frac{f(0)}{f'(0)}$$
$$x_1 = 0 - \frac{2}{-2} = x_1 = 1$$

Similarly,

$$x_2 = 1 - \frac{f(1)}{f'(1)} = x_2 = 1 - \frac{1}{1} = x_2 = 0$$

Consider a function f(x)=$e^x$ $-$ 2x=0.

Therefore,f'(x) = $e^x$ $-$ 2.

Let $x_0$ = 1,we get

$$x_{n+1} = x_n \text{ - } \frac{e^{x_n} - 2x_n}{e^{x_n} - 2}$$
$$x_1 = 1 - \frac{f(1)}{f'(1)} = 1 - 1 = 0$$

We observe that the first iteration produces 1 and second iteration produces 0. This concludes that the sequence now alternates between these two values without actually converging to the root.

### Derivative Issues

The function has to be continuously differentiable in the neighbourhood of the root for the Newton's Method to work. If it is not, Newton's method will always diverge or fail except in the lucky case in which solution is obtained on the first guess.

1. If the function f(x) is not differentiable in the neighbourhood of root, then the newton method will give false results.

2. If the derivative of the function does not exist at the roots, then the next iterations will not converge to the roots, instead the distance between the root and $x_{k+1}$ increases.

3. If f is twice differentiable then the error in the tangent line approximation is $1/2h^2$f"(c),for some c between $x_0$ and $x_0$ + h.So,if | f"(x) | is very large ,then the error in tangent line approximation is very large.For eg:

   Consider the function:

   f(x)= x$^{\frac{1}{3}}$

   Therefore,

   $$\mathrm{x}_{n+1} = \mathrm{x}_n - \frac{x_n^{\frac{1}{3}}}{\frac{1}{3}x_n^{\frac{1}{3}-1}} = \mathrm{x}_n - 3\mathrm{x}_n = \text{-}2\mathrm{x}_n$$

   Thus, we observe that the after each iteration, distance gets doubled from the solution. As also visible, it overshoots the value and lands on the side opposite to the side of the y−axis where the iterative point is. In fact, it is clearly visible that the iterations diverges towards infinity.

# Applications of Newton's Method

## Minimization and maximization problems

Newton's Method can be used to find the maxima or minima for a function. The derivative is zero at the minimum or maximum, therefore Newton's Method can be applied to the derivative of the function to find maxima or minima. Note that Newton's method only gives the value. Further calculations have to be made to find whether it is a maxima or minima. The iteration becomes:

$$\mathrm{x}_{n+1} = \mathrm{x}_n - \frac{f'(x_n)}{f''(x_n)}$$

## Multiplicative inverse of numbers

Newton's Method can be used to find the reciprocal of a number using only subtraction and multiplication. It is an important application of this method.

Consider the number D whose reciprocal is to be found out. Let $x$ be it's reciprocal. Therefore,
D= $\frac{1}{x}$ = Dx = 1
Therefore, to find the reciprocal of the number D amounts to finding the solution of the equation:
f(x)=Dx-1 $\Longrightarrow$ f(x) = D $- \frac{1}{x}$

Newton's iteration will be: $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$

$$x_{n+1} = x_n - \frac{D - \frac{1}{x_n}}{\frac{1}{x_n^2}}$$

$x_{n+1} = x_n(2-Dx_n)$

## Convergence Rate

For general iterative methods, define error at iteration $n$ by
$e_n = x_n$ R, where $x_n$ is approximate solution and R is true solution.
For methods that maintain interval known to contain solution, rather than specific approximate value for solution, take error to be length of interval containing solution.

Sequence converges with rate $r$ if
$lim_{n\to\infty} \frac{e_{n+1}}{|e_n|^r} = C$
for some finite non zero constant C.
If r=1, the rate of divergence is linear. Similarly if r=2, the rate of convergence is quadratic.

If R = g(R) and $|g'(R)| < 1$, then there is an interval containing R such that iteration

$$x_{n+1} = g(x_n)$$

converges to R if started within that interval.

Representing Newton's Method as above,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = g(x_n)$$

Therefore for Newton's Method, Condition for convergence is

$$|g'(x)| = |\frac{f(x)f''(x)}{[f'(x)]^2}| < 1$$

If $|g'(x)|$ >1, then the newton's method do not converges to any root.

**Convergence rate of Newton's method**

R $-$ $x_{n+1}$ = g(R) $-$ g($x_n$)
because R = g(R) and $x_{n+1}$ = g($x_n$)

Using Taylor Expansion, we can write,
g($x_n$) = g(R) + g'(R)$\frac{(R-x_n)}{1!}$ + g''(R)$\frac{(R-x_n)^2}{2!}$ + g''(R)$\frac{(R-x_n)^3}{3!}$ + ...
Ignoring higher order terms, we get
g($x_n$) = g(R) + g'(R)(R$-x_n$) + g''(R)$\frac{(R-x_n)^2}{2}$

$|g'(x)| = |\frac{f(x)f''(x)}{[f'(x)]^2}|$

Now for R,
$|g'(R)| = |\frac{f(R)f''(R)}{[f'(R)]^2}|$

and f(R)=0 because R is the true solution of function *f*.
Therefore,
$|g'(R)|$=0

g($x_n$) = g(R) + g''(R)$\frac{(R-x_n)^2}{2}$

$x_{n+1}$ = R + g''(R)$\frac{(R-x_n)^2}{2}$

$x_{n+1}$ $-$ R = g''(R)$\frac{(R-x_n)^2}{2}$

$e_{n+1}$ = g''(R)$\frac{e_n^2}{2}$

$\frac{|e_{n+1}|}{|e_n|^2} = \frac{g''(R)}{2}$

$\frac{g''(R)}{2}$ = C, where C is a finite non zero constant. Therefore,

$$\frac{|e_{n+1}|}{|e_n|^2} = \mathrm{C}$$

and we know that,

$$lim_{n\to\infty} \frac{|e_{n+1}|}{|e_n|^r} = \mathrm{C}$$

which implies $r = 2$. Therefore, Newton's Method is **quadratically convergent**.

## Using Octave as Mathematical Computational tool

In order the illustrate the Newton's Basins of Attraction for a given polynomial, we need a software which can implement these calculations and plot basins accordingly. The software we have used is Octave.

### What is Octave?

Octave, mostly known as GNU Octave is a high-level programming language, which is primarily intended for numerical computations. It provides a command line interface to solve linear and non-linear problems numerically. This language is mostly compatible with MATLAB, which is another high-level programming language which serves the same purpose but is more vast and can be used for more purposes also. GNU Octave takes fraction of time to solve and compute numerical computations than in other languages like C, C++, Java etc. Octave language is an interpreted programming language and supports many common C standard library functions. Octave is made available under the GNU General Public License, therefore it is freely available. Octave has a built-in compatibility for complex numbers and have powerful built-in math functions. Therefore, Octave seemed to be the best suitable language for illustrating Newton Basins of Attraction.

## Octave Computational Results

We used Octave to find the roots(real or complex) of the polynomial, show the sequential iteration of a number towards any of the root of the polynomial and finally, to illustrate Newton Basins of Attraction for some polynomials of different degrees.

### Newton Basins of Attraction

As discussed earlier, Newton Basin is a set of initial guesses for which Newton's method converges to a particular root of the polynomial. Below illustrated are Newton Basins for different polynomials of different degrees. The polynomials used are:

| **Polynomial** | **Roots** |
|---|---|
| $z^2 - 8z + 15$ | 3, 5 |
| $z^3 - 1$ | $1, \dfrac{-1-\sqrt{3}i}{2}, \dfrac{-1+\sqrt{3}i}{2}$ |
| $x^3 - 15x^2 + 71\text{x} - 105$ | 3, 5, 7 |
| $z^3 - 18z^2 + 104\text{z} - 192$ | 4, 6, 8 |
| $z^4 + 1$ | $-0.70 + 0.70i$, $-0.70 - 0.70i$, $0.70 + 0.70i$, $0.70 - 0.70i$ |
| $z^4 + 2z^3 + 3z^2 + 4\text{z} + 5$ | $0.28 + 1.416i$, $0.28 - 1.416i$, $-1.28 + 0.85i$, $-1.28 - 0.85i$ |
| $z^4 - 0.98z^3 + 0.4802z^2 + 0.2705\text{z} + 0.0762$ | $-0.2 - 0.2i$, $-0.2 + 0.2i$, $0.69 + 0.69i$, $0.69 - 0.69i$ |
| $z^4 - 6z^3 + 8z^2 - 8\text{z} + 90$ | $3.93 + 1.42i$, $3.93 - 1.42i$, $-0.93 + 2.07i$, $-0.93 - 2.07i$ |
| $z^4 - 34z^3 + 206z^2 - 744\text{z} + 945$ | $2.24 + 3.39i$, $2.24 - 3.39i$, 27.43 and 2.085 |
| $x^5 + 9x^4 + 7x^3 + 9x^2 + 3\text{x} + 245$ | $1.392 + 1.5797i$, $1.392 - 1.5797i$, $-1.7245 + 1.9119i$, $-1.7245 - 1.9119i$, -8.3353 |

### Degree 2 polynomials

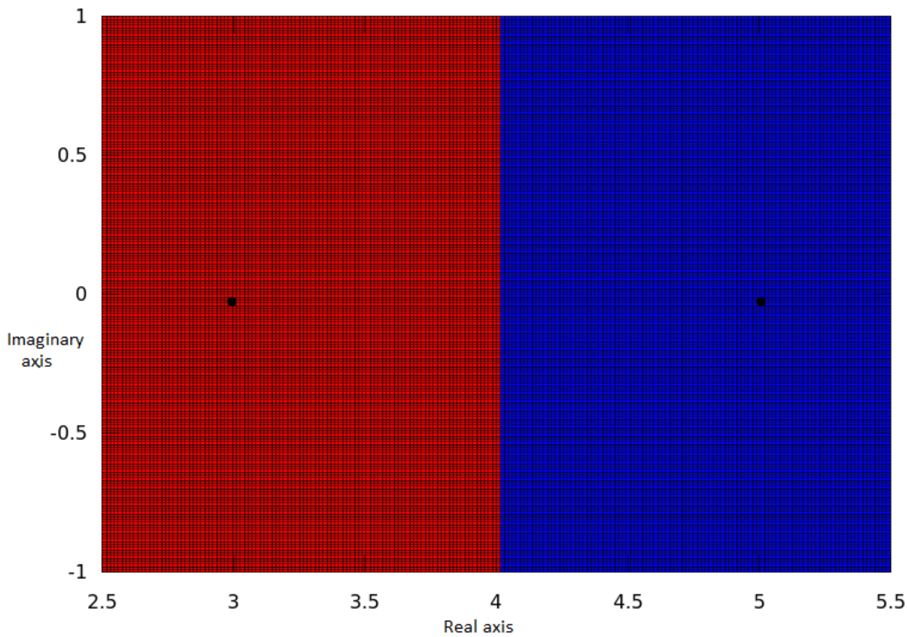


Figure 7: Newton Basin for $z^2 - 8z + 15$

Figure 4 shows the Newton basin of the polynomial$x^2 - 8x + 15$ which has two real roots, 3 and 5. The polynomial has two Newton basins, one Newton basin for each root. Any initial guesses in the basin coloured yellow will converge to the root 3. Any initial guesses in the basin coloured red will converge to the root 5.

### Degree 3 polynomials

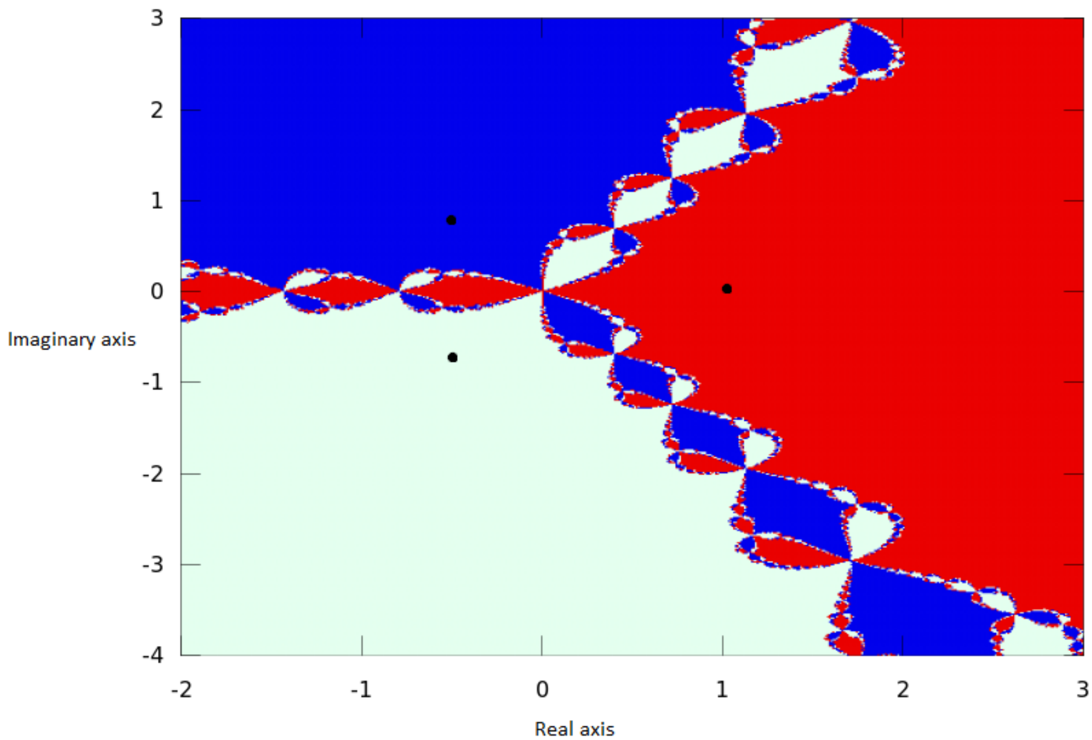


Figure 8: Newton Basin for f(z) = $z^3 - 1$

Fig. 7 shows the Newton basins for the polynomial $x^3 - 1$ which has one real root 1 and two complex roots $\dfrac{-1-\sqrt{3}i}{2}$ and $\dfrac{-1+\sqrt{3}i}{2}$. The polynomial has three Newton basins, one for each root. Any initial guess in the red coloured region will converge towards root 1. Similarly, any initial guess in the blue coloured region will converge towards root $\dfrac{-1+\sqrt{3}i}{2}$ and initial guess in the light blue coloured region will converge towards root $\dfrac{-1-\sqrt{3}i}{2}$.

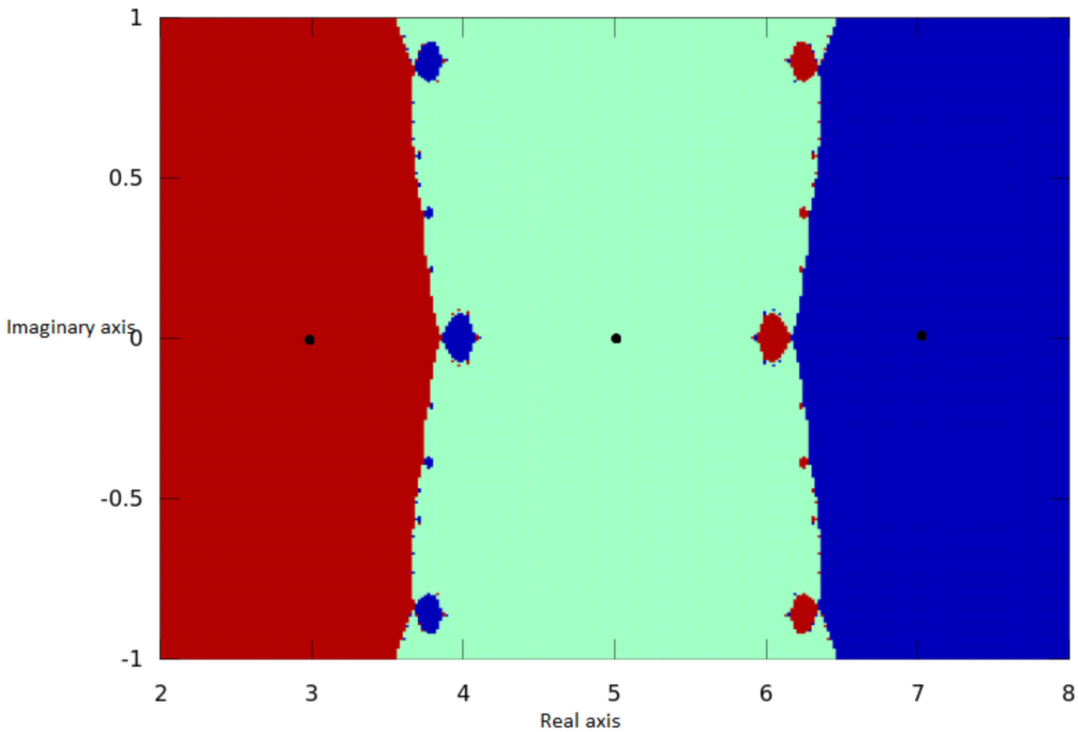


Figure 9: Newton Basin for f(x) = $x^3$ − 15$x^2$ + 71x − 105

Fig. 8 shows the Newton basins for the polynomial $x^3 - 15x^2 + 71x - 105$ which has three real roots 3,5 and 7. The polynomial has three Newton basins, one for each root. Any initial guess in the red coloured region will converge towards root 3. Similarly, any initial guess in the blue coloured region will converge towards root 7 and initial guess in the light green coloured region will converge towards root 5.

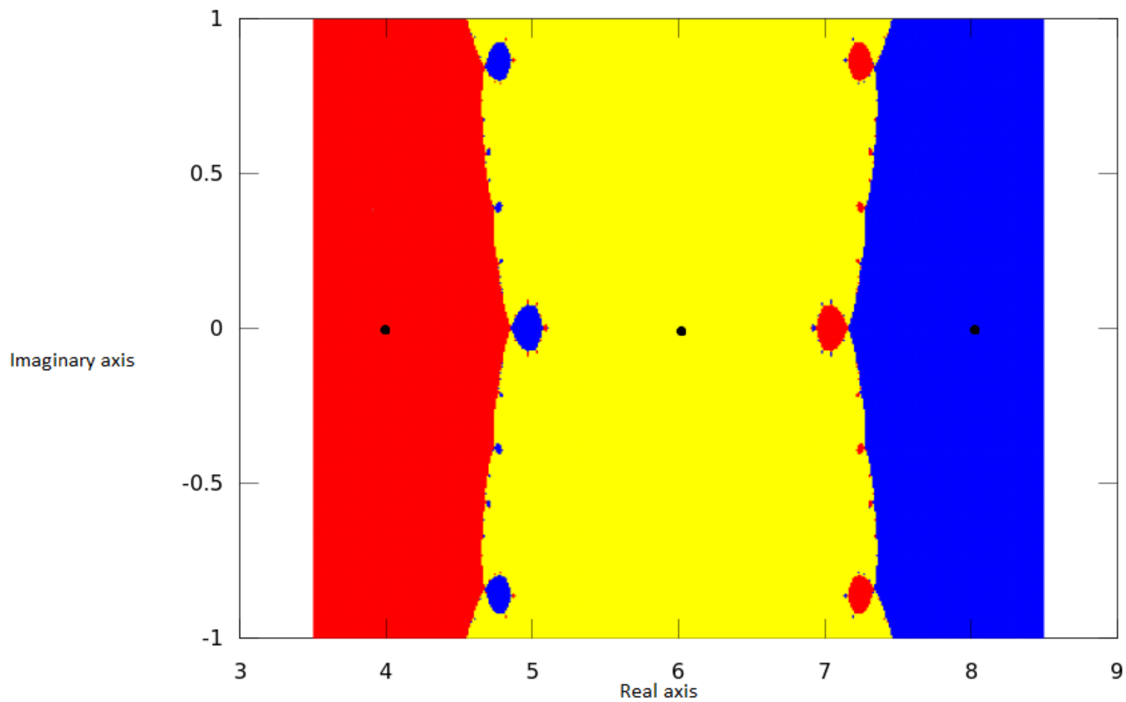


Figure 10: Newton Basin for f(z) = $z^3$ − $18z^2$ + 104z − 192

Fig. 9 shows the Newton basins for the polynomial $z^3 - 18z^2 + 104$z − 192 which has three real roots 4,6 and 8. The polynomial has three Newton basins, one for each root. Any initial guess in the red coloured region will converge towards root 4. Similarly, any initial guess in the blue coloured region will converge towards root 8 and initial guess in the yellow coloured region will converge towards root 6.

### Degree 4 polynomials

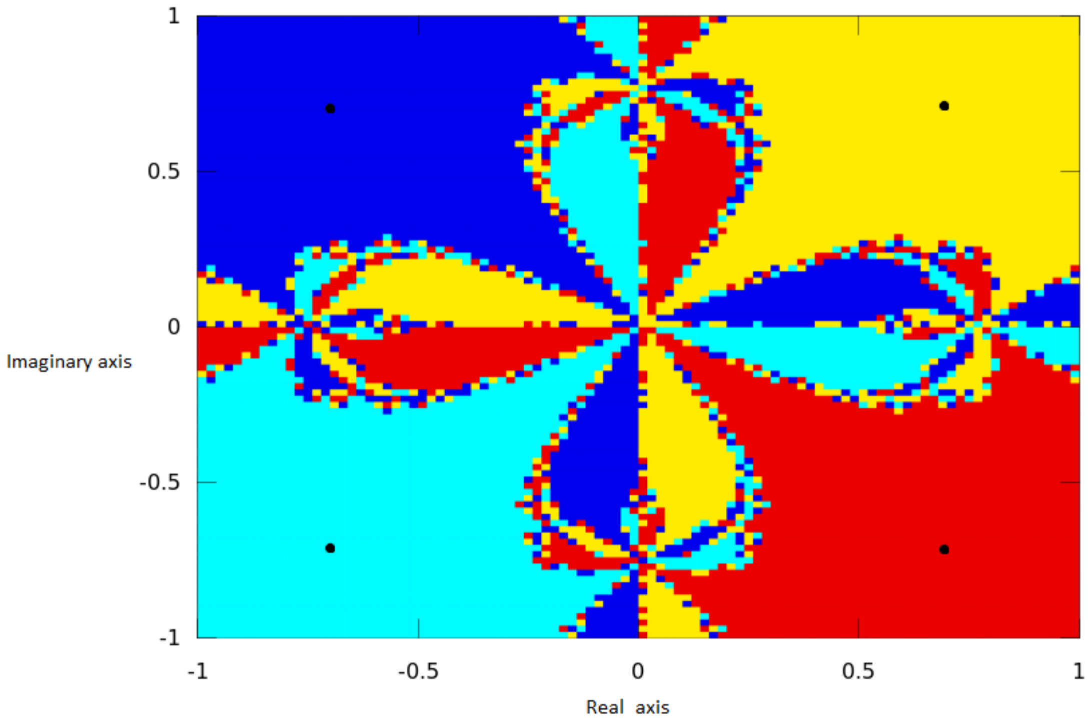


Figure 11: Newton Basin for f(z) = $z^4 + 1$

Fig. 10 shows the Newton basins for the polynomial $z^4 + 1$ which has four complex roots in which each pair is conjugate of each other. The roots are $-0.70 + 0.70i$, $-0.70 - 0.70i$, $0.70 + 0.70i$ and $0.70 - 0.70i$. The polynomial has four Newton basins, one for each root. Any initial guess in the red coloured region will converge towards root $0.70 - 0.70i$. Similarly, any initial guess in the yellow coloured region will converge towards root $0.70 + 0.70i$, any initial guess in the blue coloured region will converge towards root $-0.70 + 0.70i$ and any initial guess in the light blue coloured region will converge toward the root $-0.70 - 0.70i$.

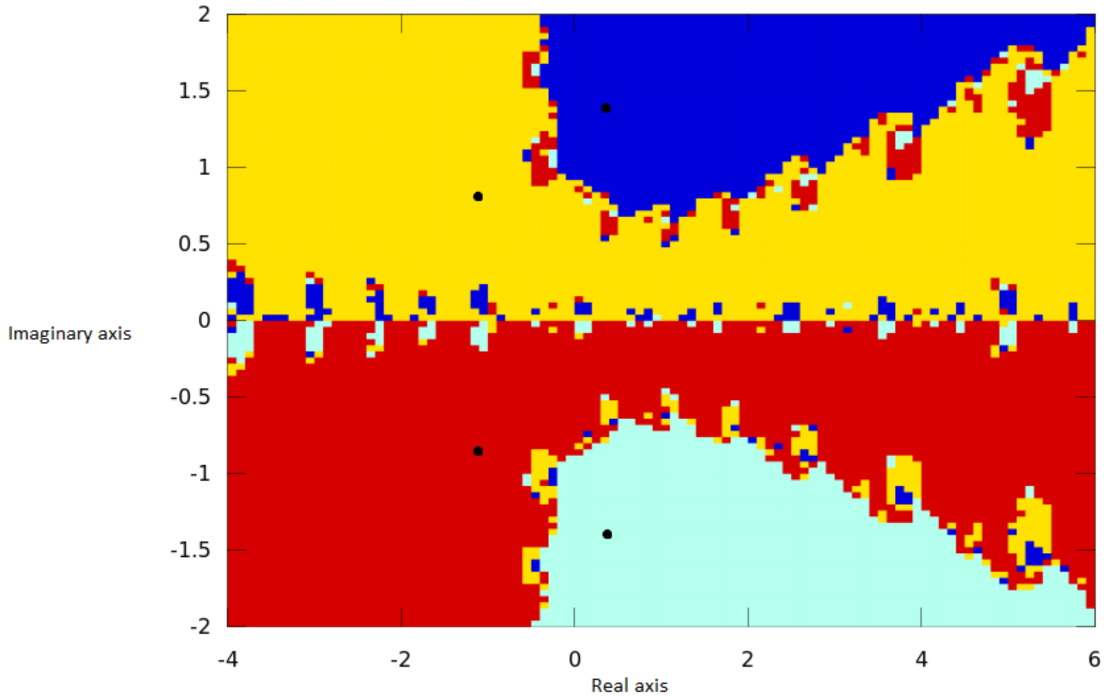


Figure 12: Newton Basin for f(z) = $z^4$ + $2z^3$ + $3z^2$ + 4z + 5

Fig. 11 shows the Newton basins for the polynomial $z^4$ + $2z^3$ + $3z^2$ + 4z + 5 which has four complex roots in which each pair is conjugate of each other. The roots are $0.28 + 1.416i$, $0.28 - 1.416i$, $-1.28 + 0.85i$ and $-1.28 - 0.85i$. The polynomial has four Newton basins, one for each root. Any initial guess in the red coloured region will converge towards root $-1.28 - 0.85i$. Similarly, any initial guess in the yellow coloured region will converge towards root $-1.28 + 0.85i$, any initial guess in the blue coloured region will converge towards root $0.28 + 1.416i$ and any initial guess in the light blue coloured region will converge toward the root $0.28 - 1.416i$.

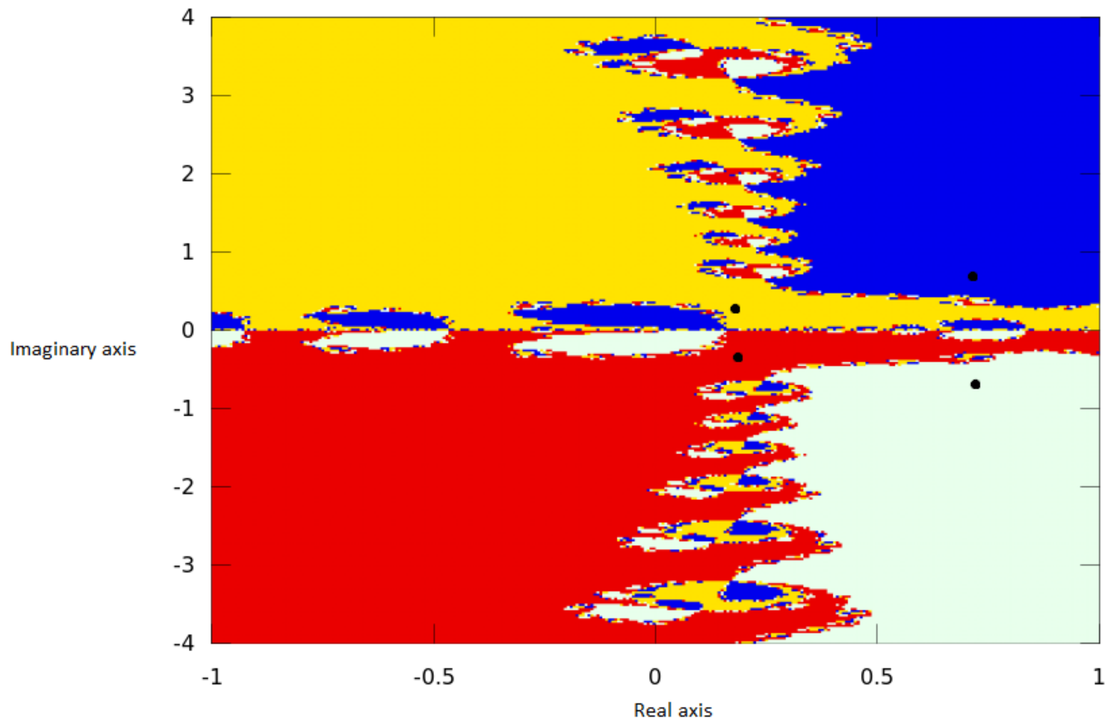


Figure 13: Newton Basin for f(z) = $z^4$ − $0.98z^3$ + $0.4802z^2$ + 0.2705z + 0.0762

Fig. 12 shows the Newton basins for the polynomial $z^4 - 0.98z^3 +$ $0.4802z^2$ + 0.2705z + 0.0762 which has four complex roots in which each pair is conjugate of each other. The roots are $-0.2 - 0.2i$, $-0.2 + 0.2i$, $0.69 + 0.69i$ and $0.69 - 0.69i$. The polynomial has four Newton basins, one for each root. Any initial guess in the red coloured region will converge towards root $0.2 - 0.2i$. Similarly, any initial guess in the yellow coloured region will converge towards root $-0.2 - 0.2i$, any initial guess in the blue coloured region will converge towards root $0.69 + 0.69i$ and any initial guess in the white coloured region will converge toward the root $0.69 - 0.69i$.

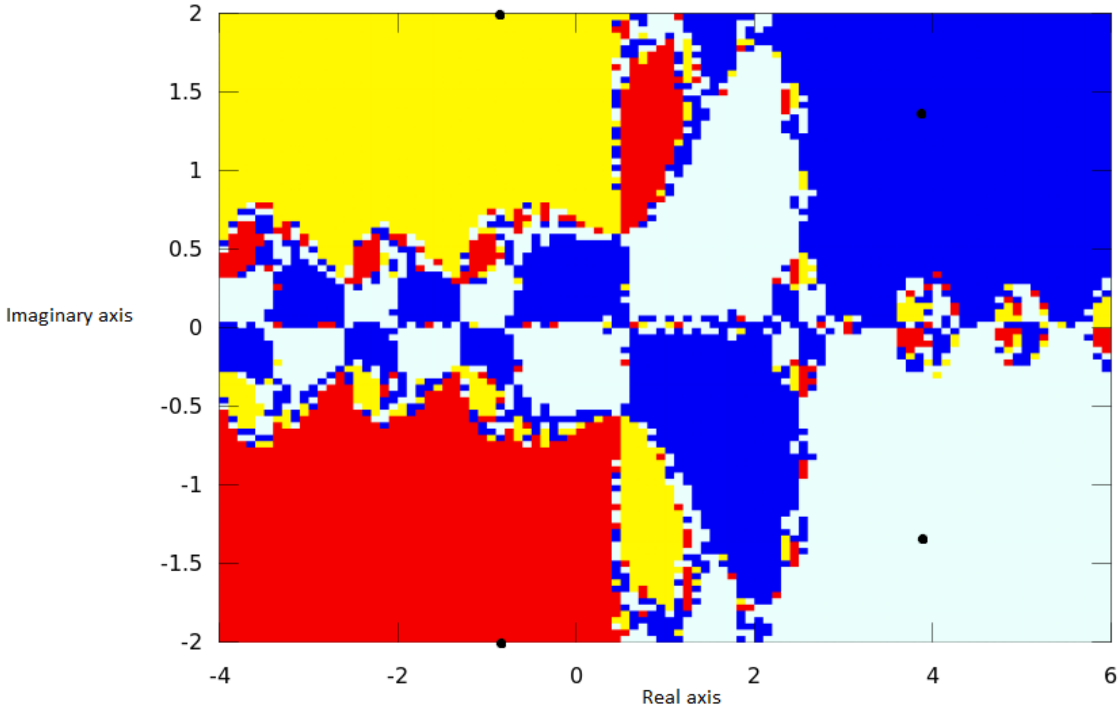


Figure 14: Newton Basin for f(z) = $z^4 - 6z^3 + 8z^2 -$ 8z + 90

Fig. 12 shows the Newton basins for the polynomial $z^4 - 6z^3 + 8z^2 -$ 8z + 90 which has four complex roots in which each pair is conjugate of each other. The roots are $3.93+1.42i$, $3.93-1.42i$, $-0.93+2.07i$ and $-0.93-2.07i$. The polynomial has four Newton basins, one for each root. Any initial guess in the red coloured region will converge towards root $-0.93-2.07i$. Similarly, any initial guess in the yellow coloured region will converge towards root $-0.93 + 2.07i$, any initial guess in the blue coloured region will converge towards root $3.93 + 1.42i$ and any initial guess in the white coloured region will converge toward the root $3.93 - 1.42i$.

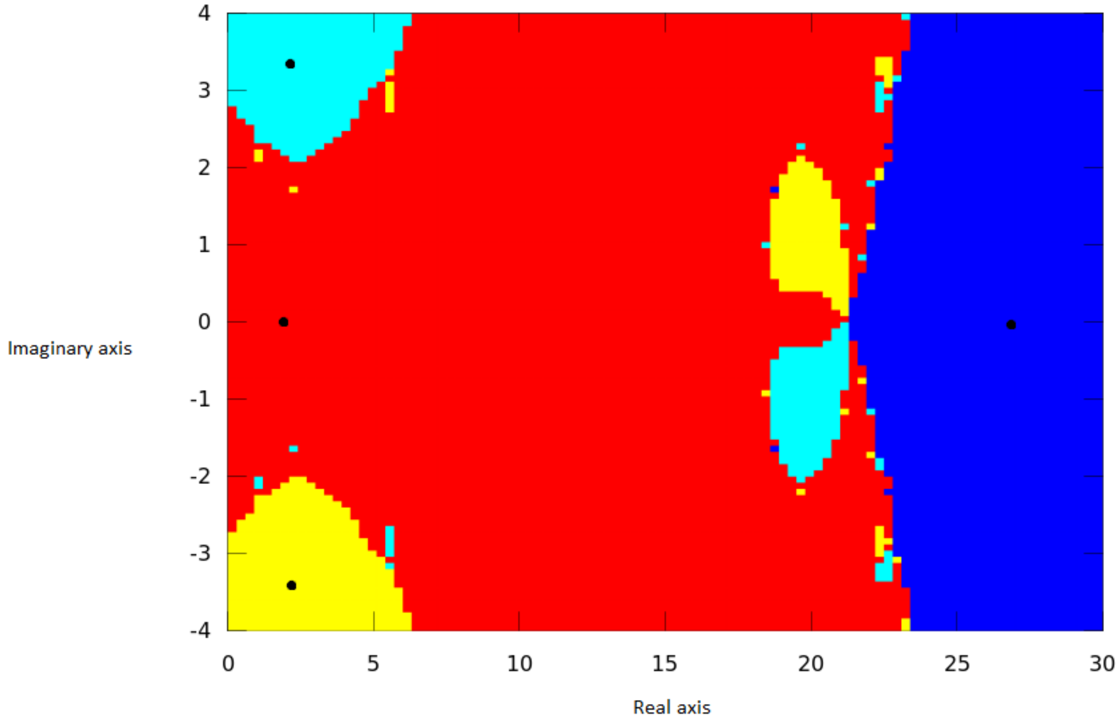


Figure 15: Newton Basin for f(z) = $z^4$ − $34z^3$ + $206z^2$ − 744z + 945

Fig. 13 shows the Newton basins for the polynomial $z^4$ − $34z^3$ + $206z^2$ − 744z + 945 which has two complex roots $2.24 + 3.39i$ and $2.24 - 3.39i$. It also has two real roots which are 27.43 are 2.085. The polynomial has four Newton basins, one for each root. Any initial guess in the red coloured region will converge towards root 2.085. Similarly, any initial guess in the blue coloured region will converge towards root 27.43, any initial guess in the yellow coloured region will converge towards root $2.24 - 3.39i$ and any initial guess in the light blue coloured region will converge toward the root $2.24 + 3.39i$.

### Degree 5 polynomials

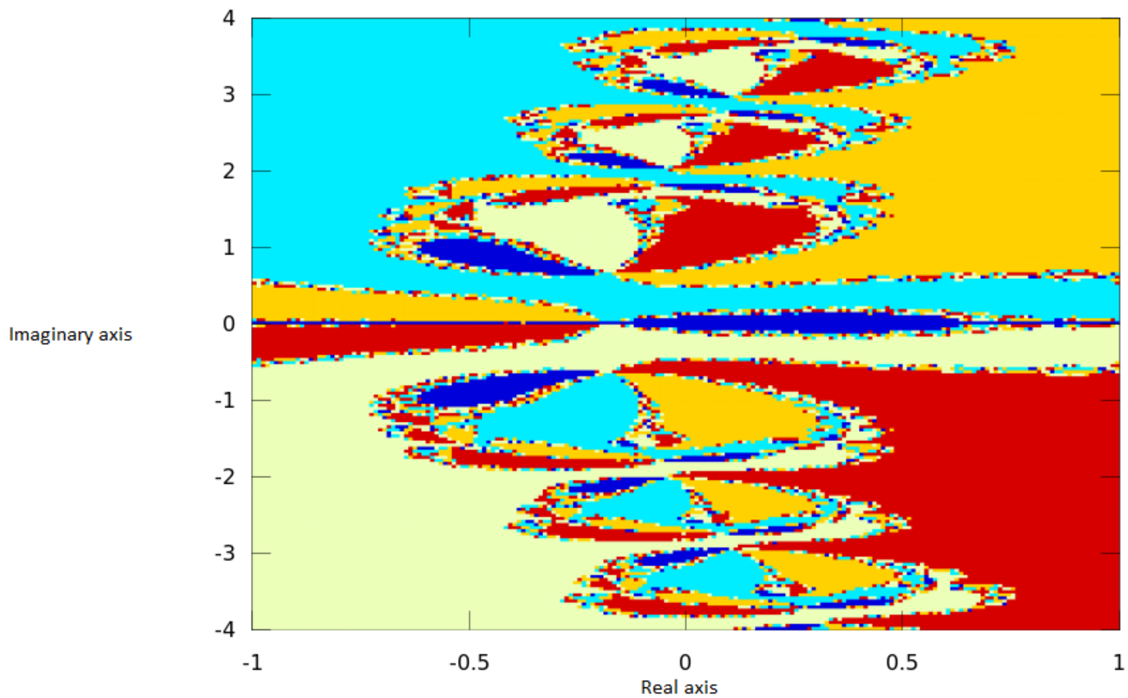


Figure 16: Newton Basin for f(x) = $x^5$ + $9x^4$ + $7x^3$ + $9x^2$ + 3x + 245

Fig. 13 shows the Newton basins for the polynomial $x^5$ + $9x^4$ + $7x^3$ + $9x^2$ + 3x + 245 which has four complex roots $1.392 + 1.5797i$, $1.392 - 1.5797i$, $-1.7245 + 1.9119i$ and $-1.7245 - 1.9119i$. It also has a real root which is -8.3353. The polynomial has five Newton basins, one for each root. Any initial guess in the red coloured region will converge towards root $1.392 - 1.5797i$. Similarly, any initial guess in the yellow coloured region will converge towards root $1.392+1.5797i$, any initial guess in the cream coloured region will converge towards root $-1.7245 - 1.9119i$, any initial guess in the sky blue coloured region will converge toward the root $-1.7245 + 1.9119i$ and any initial guess in dark blue region will converge towards -8.3353.

## Conclusion

We analysed the Newton Method and found that it works excellent for an intelligent initial guess(when it is quite close to the root) in a well behaved function. But,it has some limitations also .This method is quadratically convergent when initial guess is close to the one of the roots of the function.Around this initial guess function f(x) and its derivative f'(x) should be continuous and differentiable at all the points inside the domain.Favourably,f'(x) should not be very small and f"(x) must not be very large. Using power of computation and Octave, we computed and analysed Newton Basins and wastelands for different polynomials. We further analysed the sequential iteration of Newton's method for an initial guess towards a root of the polynomial using Octave.

## References


- Wikipedia, 'Newton's Method'
  $http://en.wikipedia.org/wiki/Newton's\_method$

- David E. Joyce, 'Newton's Method and Basins'
  $http://aleph0.clarku.edu/\sim djoyce/newton/method.html$

- Jonathan S. Renning, 'Convergence rate of Newton's Method'
  $http://www.math-cs.gordon.edu/courses/ma342/handouts/rate.pdf$

- Courtney Watts, 'Evaluating Newton's method for real and complex roots of various polynomials
  $http://cms.uhd.edu/faculty/redlt/cwseniorproject.pdf$

- Digital Image Processing Using Matlab,Second Edition,By Rafael C.Gonzalez

- Programming In Matlab,By Marc E Herniter

- Matlab Programming For Engineers,Third Edition,By Stephen J.Chapman

- Thomas Calculus,Eleventh Edition ,By Maurice D.Weir,Joel Hass,Frank R.Giordano

- www.octave.org

- mathworld.wolfram.com ... Numerical Methods Root-Finding

- www.math.ubc.ca/ anstee/math104/newtonmethod.pdf

- www.sosmath.com/calculus/diff/der07/der07.html

- www.math.sc.edu/cgi-bin/sumcgi/Newton.pl

- tutorial.math.lamar.edu/Classes/CalcI/NewtonsMethod.aspx

- https://www.math.ucdavis.edu/ thomases/W11_16C1_lec_3_11_11.pdf

- www.math.ubc.ca/ anstee/math104/newtonmethod.pdf

- www3.ul.ie/ mlc/support/CompMaths2/files/NewtonExample.pdf

- www3.ul.ie/ mlc/support/Loughborough %20website/chap13/13_3.pdf

- personal.maths.surrey.ac.uk/st/S.Gourley/NewtonRaphson.pdf

- www.b-u.ac.in/sde_book/msc_mathnum.pdf

- www.maths.manchester.ac.uk/ cds/internal/tables/numerical.pdf
- www.math.colostate.edu/ gerhard/classes/331/lab/newton.html
- www.math.pitt.edu/ sussmanm/2070/lab_04/lab_04.html
- www.math.washington.edu/ greenbau/Math_464/page.html