Iterative Methods to Solve Systems of Nonlinear Algebraic Equations

Md Shafiful Alam
Western Kentucky University, mdshafiful.alam694@topper.wku.edu

Follow this and additional works at: https://digitalcommons.wku.edu/theses

Part of the Dynamical Systems Commons, Non-linear Dynamics Commons, and the Numerical Analysis and Computation Commons

Recommended Citation
https://digitalcommons.wku.edu/theses/2305

This Thesis is brought to you for free and open access by TopSCHOLAR®. It has been accepted for inclusion in Masters Theses & Specialist Projects by an authorized administrator of TopSCHOLAR®. For more information, please contact topscholar@wku.edu.
ITERATIVE METHODS TO SOLVE SYSTEMS OF NONLINEAR ALGEBRAIC EQUATIONS

A Thesis
Presented to
The Faculty of the Department of Mathematics
Western Kentucky University
Bowling Green, Kentucky

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

By
Md Shafiful Alam

May 2018
ITERATIVE METHODS TO SOLVE SYSTEMS OF NONLINEAR ALGEBRAIC EQUATIONS

Date Recommended: April 17, 2018

Mark Robinson
Dr. Mark Robinson, Director of Thesis

Dominic Lanphier
Dr. Dominic Lanphier

Ngoc Nguyen
Dr. Ngoc Nguyen

Dean, Graduate School Date
To my parents Md Nazmul Alam and Aleya Begum.
I want to express my heartiest gratitude to Dr. Mark Robinson for his patient guidance, encouragement and support. I have been lucky to have associated with him in all four semesters I have been a student at Western Kentucky University. He is one of the nicest persons I have ever met. I would also like to thank Dr. Dominic Lanphier and Dr. Ngoc Nguyen for giving their valuable time to be in the thesis committee. I am grateful for everything I have learned from them.
# CONTENTS

1 Introduction 1

2 Equations in One Variable 4

2.1 Bisection Method 4
2.2 Fixed-Point Iteration Method 6
2.3 Newton-Raphson Method 7
2.4 Convergence of Newton’s Method 9
2.5 Secant Method 12
2.6 Chord Method 12
2.7 Rate of Convergence 12
2.8 Acceleration of Convergence of Newton’s Method 20
2.9 Newton’s Method for Minimization 29

3 Systems of Equations with Several Variables 32

3.1 Fixed Point Method for Several Variables 32
3.2 Newton’s Method for Several Variables 33
3.3 Acceleration of Convergence 48
3.4 Newton’s Method for Minimization 52

4 Newton’s Method in the Complex Plane 58

5 Conclusions 69

A Mathematica Codes 73
LIST OF FIGURES

2.1 Newton’s Method applied to $y = e^x - 1$ with initial approximation $x^{(0)}$ evaluating $x^{(1)}$ and then $x^{(2)}$ ........................................ 8

2.2 Graph of $y = \tan^{-1}(x)$ .................................................. 9

2.3 First two iterations of Newton’s method starting at $x = 1.5$ on the curve $y = \tan^{-1}(x)$ .................................................. 10

2.4 Graph of $f(x) = (x - 1)(x - 3)^2(x - 5)(x - 7)^2(x - 9)$ ............. 14

3.1 Intersection of $x_1x_2$ plane (gray) with the tangent plane (black) at (-4,-3) on the graph of $f_1(x_1, x_2) = x_1^2 + x_2^2 - 9$ (light gray) ........ 39

3.2 Intersection of $x_1x_2$ plane (gray) with the tangent plane (black) at (-4,-3) on the graph of $f_2(x_1, x_2) = -x_1 + x_2^2 - 3$ (light gray) ........ 39

3.3 The intersection of $x_1x_2$ plane and $f_1(x_1, x_2) = x_1^2 + x_2^2 - 9$ and $f_2(x_1, x_2) = -x_1 + x_2^2 - 3$, three solutions are $(-3,0)$, $(2,\sqrt{5})$ and $(2,-\sqrt{5})$ .... 39

3.4 The intersection of $x_1x_2$ plane and $f_1(x_1, x_2)$ and $f_2(x_1, x_2)$ and the tangent planes at (-4,-3) on these two surfaces $f_1$ and $f_2$ ............ 39

3.5 The graph of contour plots of different spectral radii ..................... 45

3.6 The region where the spectral radius is less than one .................... 45

3.7 The graph of three surfaces and their intersection ....................... 47

3.8 Graph of $f(x_1, x_2) = x_1x_2e^{-x_1^2-x_2^2}$ ............................ 55

3.9 The region where $1 \times 1$ leading principal submatrix of the Hessian has positive determinant .............................. 56

3.10 The region where $2 \times 2$ leading principal submatrix of the Hessian has positive determinant .............................. 56
3.11 The region which satisfies both conditions i.e. the Hessian is positive definite .................................................. 56

4.1 All pre-images up to level one (we call zero or the origin as level 0 of pre-images and its pre-images as level one pre-images) ............. 63
4.2 All pre-images up to level two ........................................ 63
4.3 All pre-images up to level three ...................................... 64
4.4 All pre-images up to level four ....................................... 64
4.5 All pre-images up to level five ...................................... 64
4.6 All pre-images up to level six ....................................... 64
4.7 All pre-images up to level seven .................................... 64
4.8 All pre-images up to level eight .................................... 64
4.9 All pre-images up to level nine ...................................... 65
4.10 Fractal image showing basin of attraction for each root of $z^3 - 1 = 0$ in different shade ............................................. 66
4.11 Fractal image showing basin of attraction for each root of $z^4 - 1 = 0$ .................................................. 67
4.12 Fractal image showing basin of attraction for each root of $z^5 - 1 = 0$ .................................................. 67
# LIST OF TABLES

2.1 Newton’s method for zeros of different multiplicity . . . . . . . . . . 14
2.2 Acceleration of Convergence by modification of Newton’s Method . . 21
2.3 Composition of Newton’s Method for different multiplicity of zeros . . 25
2.4 Acceleration of Convergence by modification of Newton’s Method . . 28

3.1 Newton’s Method for several variables . . . . . . . . . . . . . . . . 40
3.2 Newton’s Method for several variable component-wise . . . . . . . . 42
3.3 Newton’s Method for three variables . . . . . . . . . . . . . . . . . 47
3.4 Acceleration of Newton’s Method for two variables . . . . . . . . . . 49
3.5 Newton’s Method composed with itself for two variables . . . . . . 50
3.6 Newton’s Method for Minimization . . . . . . . . . . . . . . . . . . 57

4.1 Newton’s method for complex variable . . . . . . . . . . . . . . . . 59
4.2 Newton’s method . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
Iterative methods have been a very important area of study in numerical analysis since the inception of computational science. Their use ranges from solving algebraic equations to systems of differential equations and many more. In this thesis, we discuss several iterative methods, however our main focus is Newton’s method. We present a detailed study of Newton’s method, its order of convergence and the asymptotic error constant when solving problems of various types as well as analyze several pitfalls, which can affect convergence. We also pose some necessary and sufficient conditions on the function $f$ for higher order of convergence. Different acceleration techniques are discussed with analysis of the asymptotic behavior of the iterates. Analogies between single variable and multivariable problems are detailed. We also explore some interesting phenomena while analyzing Newton’s method for complex variables.
Chapter 1

Introduction

Solving equations is one of the basic topics in mathematics we learn in school. For example, we have learned how to solve systems of linear equations. Even though linear equations can be quite problematic to handle some times, it is not hard to get a clear view of the geometry involved. The same goes for nonlinear polynomial equations in one variable. However, in this case it is not always possible to come up with solutions analytically, especially when the degrees of the equations exceed four. Besides, there is the world of nonlinear equations with more than one variable, which turns out to be very hard to solve and the geometry involved beats our limited visualization capabilities. Even for the equations for which we are able to find solutions, the calculations become too complicated to be desirable. With the continuous development of science, mathematicians and engineers are facing more and more sophisticated and complicated equations, which are mostly nonlinear in nature usually involving more than one variable. In these situations, we usually depend on different numerical iterative methods to solve these problems with certain precision.

In this thesis, we discuss some of the established iterative methods to solve nonlinear equations in one variable like Bisection, Fixed-Point Iteration, Newton’s (Newton-Raphson), Secant and Chord Method. However, our primary focus is on one of the most powerful methods to solve equations or systems of equations, namely Newton’s method. Newton’s method is particularly popular because it provides faster convergence compared to others if the function is well behaved near the solution and the initial approximation is chosen carefully. We show that the method is not
as effective and may even fail to converge if the initial approximation is not chosen carefully. We show that some regions are better than others, from where a good initial approximation can be chosen for guaranteed convergence.

We also explore the rate of convergence and the asymptotic error constant. We analyze how the speed of convergence is affected by the multiplicity of the associated zero. It is also true that the speed of convergence of Newton’s method depends on the nature of the iteration function and its derivatives. We discuss an extension of this idea to pose conditions on the function itself. We show how the derivatives of a function $f$ can determine the order of convergence of Newton’s method. We also present some formulas to calculate the asymptotic error constant exactly and compare with numeric results.

In case of higher multiplicity of zero, the performance of Newton’s method is not something we can get excited about. We present ways to accelerate the convergence of the method for zeros of different multiplicity including simple zeros. Newton’s method can also be used to find minimizers. We analyze how we can guarantee the convergence of the method to a minimizer.

We extend our discussion to solving systems of equations in more than one variable. Newton’s method is the primary method that we use to analyze different phenomena. One of the major focuses of this thesis is the analogies between single variable problems and multivariable problems. We investigate how certain single variable phenomena can be translated to the higher dimensional cases. This makes it easier to analyze the behavior in higher dimensions. This also tells us what type of behavior we can expect from the method.

Similar to the single variable case, we analyze the convergence of Newton’s method for multivariable problems. Since we have more than one component in the higher dimensional cases, we analyze the convergence as a whole and component wise. We present techniques to calculate regions for good initial approximations to give
guaranteed convergence. We also show the acceleration techniques for systems of equations. We present discussions on minimization and how to guarantee convergence of Newton’s method to the minimizer.

Finally, we apply Newton’s method to problems involving complex variables. A complex equation can be solved as a single variable problem in $z$ and as a system of equations in two variables. We show that both ways generate exactly the same iterations, while one method has benefits over the others. The region of attraction for Newton’s method in the complex plane shows amazing fractal behavior. We analyze the basin of attraction for famous $n$-th root of unity problems because of their beautiful symmetric behavior. We also discuss some interesting behavior of Newton’s iterates in certain regions.
Chapter 2

Equations in One Variable

2.1 Bisection Method

The Bisection method is the most intuitive technique to find roots or solutions of an equation of the form $f(x) = 0$. The history of the method can be traced as early as 1700 B.C.E. Oldest use of this technique is found in the Yale Babylonian Collection, giving the approximation of $\sqrt{2}$ in base-60 number format, which is accurate up-to $10^{-5}$.[2]

This method is based on the Intermediate Value Theorem. The basic idea of the technique is as follows:
Suppose $f$ is a continuous function defined on the interval $[a, b]$, where $f(a)$ and $f(b)$ have opposite signs. The Intermediate Value Theorem implies that there exists a number $p$ in $(a, b)$ such that $f(p) = 0$. It does not say how many roots are there in this interval. However, regardless of the number of roots in the interval, the method will be able to converge to one of the roots, as long as the function is continuous in that interval.

The method implements a binary search algorithm. At every iteration, the interval is divided into two parts, each having length half of the original. Then it looks for the sub-interval, which contains a root by checking the sign of the function at the mid-point. The method replaces one of the endpoints of the interval, at which the function $f$ has the same sign as at the mid-point. The same procedure is repeated until a desirably small interval is found where we can consider the mid-point as the approximation of the root accurate up to the predefined accuracy. We need
appropriate stopping criteria so that the method does not run infinitely. Here are some of the most commonly used stopping criteria:

- When the absolute value of the difference between two consecutive iterates is less than a predefined small quantity, \( |p^{(k)} - p^{(k-1)}| < \epsilon \), where \( p^{(k)} \) is the approximation of solution at \( k \)-th iteration and \( \epsilon \) is the predefined accuracy.

- When the ratio of the absolute difference of two successive iterates and absolute value of current iterate is less than a predefined small quantity, \( \frac{|p^{(k)} - p^{(k-1)}|}{|p^{(k)}|} < \epsilon \), \( p^{(k)} \neq 0 \).

- When the absolute value of the function gets closer to zero or less than an acceptable small number, \( |f(p^{(k)})| < \epsilon \).

These stopping criteria are applicable to all types of iterative methods we are going to discuss. However, all the stopping criteria have their own difficulties. Without additional knowledge about \( f \) and the root, the second stopping criterion is considered to be the best for most of the methods as it comes closest to testing the relative error.

Setting a bound on the maximum number of iterations is considered to be a good practice. [2]

All that being said, the unique binary search algorithm of bisection method makes it possible to actually come up with an upper bound for the error by the formula \( |p^{(k)} - p| \leq \frac{b-a}{2^k} \), where \( p^{(k)} \) is the approximation of solution at \( k \)-th iteration, \( p \) is the actual solution and \( [a, b] \) is the initial interval.

Selection of the interval is of paramount importance for this method. If we do not know the whereabouts of the solutions, it might be hard to come up with an appropriate interval. The reason is, if there are even number of roots in the chosen interval \( [a, b] \), then both \( f(a) \) and \( f(b) \) may have the same sign. Again if there is no solution in the interval \( [a, b] \), we can observe same issue. In some of the cases, where the \( x \)-axis is tangent to the function, which is obviously a zero of the function,
we may not have different signs around that root. For the bisection method to work, the function must have different signs at the endpoints of the interval which contains the root. Another drawback of the method is the speed of convergence. If the chosen interval is not suitably small enough, the number of iterations required to get expected accuracy might become undesirably large. Also, better accuracy needs a higher number of iterations. However, the advantage of the method is that it will always converge to a root, where the method is applicable.

2.2 Fixed-Point Iteration Method

A number $p$ is called a fixed point of a function $f(x)$ if $f(p) = p$. Root finding problems can be viewed as fixed-point problems, which are easier to analyze and certain choices of fixed point can lead to very powerful root-finding technique. We say these two types of problems are equivalent because a root-finding problem $f(p) = 0$ can easily be expressed as a fixed-point problem by defining a function $g$ with a fixed point at $p$ in a number of ways. One example would be $g(x) = x - f(x)$. If the function $g$ has a fixed point at $p$, then $f(x) = x - g(x)$ has a zero at $p$. [2]

We present the following theorem from [2], which gives sufficient conditions for the existence and uniqueness of a fixed point:

**Theorem 2.1**

(i) If $g \in C[a, b]$ (continuously differentiable) and $g(x) \in [a, b]$ for all $x \in [a, b]$, then $g$ has at least one fixed point in $[a, b]$.

(ii) If, in addition, $g'(x)$ exists on $(a, b)$ and a positive constant $c < 1$ exists with $|g'(x)| \leq c$, for all $x \in (a, b)$, then there is exactly one fixed point in $[a, b]$.

Once we are certain that there exists a unique fixed point in an interval, then starting with an initial approximation $p^{(0)}$ from the interval, we can use the iteration function $p^{(k)} = g(p^{(k-1)})$, where $p^{(k-1)}$ and $p^{(k)}$ are the approximations at $(k-1)$-th and
$k$-th iteration respectively to generate a sequence of numbers $\{p^{(k)}\}_{k=0}^\infty$ to approximate the fixed point. Now, the question is, does this sequence of numbers converges to the desired fixed point? Furthermore, if it is converging, what is the rate/speed of the convergence? The answer to the first question can be found in the following theorem we present from [2]. We will discuss the rate of convergence at the end of this chapter.

**Theorem 2.2 (Fixed-Point Theorem)**

Let $g \in C[a, b]$ be such that $g(x) \in [a, b]$, for all $x \in [a, b]$. Suppose, in addition, that $g'$ exists on $(a, b)$ and that a constant $0 < c < 1$ exists with $|g'(x)| \leq c$, for all $x \in (a, b)$. Then for any initial approximation $p^{(0)}$ in $[a, b]$, the sequence defined by $p^{(k)} = g(p^{(k-1)}), k \geq 1$, converges to the unique fixed point $p$ in $[a, b]$.

We have mentioned that a function can be manipulated in several ways to convert a root-finding problem into a fixed-point problem. However, the rate of convergence depends on the nature of the iteration function. While choosing the iteration function, we should make sure that all the criteria mentioned in Theorem 2.2 are satisfied and the derivative $|g'(x)|$ is as small as possible near the fixed point.

### 2.3 Newton-Raphson Method

Newton-Raphson Method, popularly called Newton’s method, is one of the most powerful and popular root finding methods. The method was first introduced in the 17th century though the basic idea was around before that time.

This method tries to correct the initial approximation by a quotient of the value of the function and its first derivative at the current step. The formula can be given by

$$x^{(k)} = x^{(k-1)} - \frac{f(x^{(k-1)})}{f'(x^{(k-1)})}, f'(x^{(k-1)}) \neq 0.$$  \hspace{1cm} (2.1)

Here $x^{(k-1)}$ is the current approximation and $x^{(k)}$ is the new approximation corrected...
by the quotient $-\frac{f(x^{(k-1)})}{f'(x^{(k-1)})}$.

The formula can be derived from the first Taylor Polynomial, while the second order term of the second Taylor Polynomial gives the error bound. If we expand $f(x)$ around $x^{(0)}$ we get,

$$f(x) = f(x^{(0)}) + (x - x^{(0)})f'(x^{(0)}) + \frac{(x - x^{(0)})^2}{2} f''(\xi(x)),$$

where $\xi(x)$ lies between $x$ and $x^{(0)}$

Since $f(x) = 0$ at the solution, we can derive the formula for Newton’s method by ignoring the small second order term,

$$0 \approx f(x^{(0)}) + (x - x^{(0)}) f'(x^{(0)})$$

Rewriting this formula gives

$$x \approx x^{(0)} - \frac{f(x^{(0)})}{f'(x^{(0)})}$$

We can choose the new $x$ to be the first approximation $x^{(1)}$ and then using $x^{(1)}$ we can get $x^{(2)}$ and the procedure continues until the stopping criteria are met.

From the above picture, we can see that the intersection point of $x$-axis and the tangent of the function $f$ at $x^{(0)}$ ($x_0$ in the picture) is the next approximation $x^{(1)} = x^{(0)} - (x^{(0)} - x^{(1)}) = x^{(0)} - \frac{f(x^{(0)})}{f'(x^{(0)})}$ ($x_1$ in the picture). We continue in a similar fashion until we reach close enough to the actual root.
2.4 Convergence of Newton’s Method

For suitable functions Newton’s method can be extremely powerful and exhibit quadratic or higher order convergence. Quadratic convergence means that the order of convergence is two. Order of convergence is $\alpha$ when the absolute error at the current step is proportional to the $\alpha$-th power of the absolute error at previous step. However, the convergence of the method depends heavily on the initial approximation. It is possible to start with an initial approximation, for which the method may even fail to converge. The following is a classic example of such a case:

$$y = \tan^{-1}(x)$$

(2.2)

If we look at the graph of Equation 2.2, it might give us a clue why Newton’s method might fail to converge for certain initial approximations. We have a simple zero at $x = 0$, where the slope is one and as we go away from the root on both sides, the slope gets smaller and the graph starts becoming flatter. The significance of this phenomenon is that the tangent line at any of the points, where the curve is flatter, is not going to lead us to the zero. The tangent line is going to cut the $x$-axis on the opposite side of the $y$-axis, which is supposed to be our new approximation. But the distance of the actual zero from the new approximation will be greater than the distance from the initial approximation, which can be observed in the Figure.
2.3. Clearly, the sequence generated by Newton’s method when started at $x = 1.5$ is diverging. Naturally, the question arises, if there exists any region with the property that if the initial approximation is in this region, the sequence will converge. In [5], the authors mentioned that there is in fact such a region. Moreover, there exists an $x_c \in [1.39, 1.4]$, if the initial approximation $x^{(0)} = x_c$, then Newton’s method will produce the cycle $x^{(1)} = -x_c, x^{(2)} = x_c, x^{(3)} = -x_c, \ldots$. If $|x^{(0)}| < x_c$, then Newton’s method converges to $x^* = 0$ and if $|x^{(0)}| > x_c$, then Newton’s method diverges. Here we analyze the iteration function from Equation 2.1 to find the point,

$$-x_c = x_c - \frac{f(x_c)}{f'(x_c)}$$

$$\Rightarrow 2x_c = \frac{f(x_c)}{f'(x_c)}$$

$$\Rightarrow 2x_c = \frac{\tan^{-1}(x_c)}{\frac{1}{1+(x_c)^2}}$$

$$\Rightarrow 2x_c = (1 + (x_c)^2) \tan^{-1}(x_c)$$

$$\Rightarrow (1 + (x_c)^2) \tan^{-1}(x_c) - 2x_c = 0$$

Using the ”FindRoot” command in Mathematica we found the root of this equation to be $x_c = 1.391745200270735$. So in theory, if we choose $x_c$ as our initial approximation,
Newton’s method will keep producing approximations $-x_c$ and $x_c$ in an alternating manner until roundoff error contaminates the result. However, in my 1.8 GHz Core\textsuperscript{TM} i7 machine, the approximations did not change even after 100,000 iterations, which took about half an hour to execute. For any initial approximation $x^{(0)}$, the method will converge for $|x^{(0)}| < x_c$ and the method will diverge for $|x^{(0)}| > x_c$. Because of the nice symmetric geometry of the curve, it was possible to come up with such condition. However, it is not always possible to make such assertion with this accuracy.

Under certain conditions though we can find a region around the root (also referred as region of attraction or basin of attraction), where any initial approximation chosen from this region, the method will have guaranteed convergence. The method is described in Theorem 5.2.1 of [5] for several variables. We translate this for single variable problems here as follows:

\textbf{Theorem 2.3}

Let $f : \mathbb{R} \to \mathbb{R}$ be continuously differentiable in an interval $I \subset \mathbb{R}$. Assume there exists $x^* \in \mathbb{R}$ and $r, \beta > 0$ such that $N(x^*, r) \subset I$, ($N$ is the subinterval $[x^* - r, x^* + r] \subset I$), $f(x^*) = 0$, $f'(x^*)$ exists with $|f'(x^*)^{-1}| \leq \beta$, and $f' \in \text{Lip}_\gamma(N(x^*, r))$, which means $f'$ is Lipschitz continuous in $N$ with Lipschitz constant $\gamma$. Then there exists $\epsilon > 0$ such that for any initial approximation $x^{(0)} \in N(x^*, \epsilon)$ the sequence $x^{(1)}, x^{(2)}, x^{(3)}, \ldots$ generated by $x^{(k)} = x^{(k-1)} - \frac{f(x^{(k-1)})}{f'(x^{(k-1)})}$, $k = 1, 2, 3, \ldots$ is well defined and converges to $x^*$. $\epsilon$ is chosen as the minimum of $\{r, \frac{1}{2\beta\gamma}\}$.

However, the interval found by the above theorem is usually smaller than the actual region of attraction. For the $y = \tan^{-1}(x)$ problem, the interval calculated by Theorem 2.3 is $[-0.7698, 0.7698]$, whereas, our analysis coupled with numeric result show that the region of convergence is the open interval $(-1.391745200270735, 1.391745200270735)$. One thing to note here that for this theorem to be applicable, the first derivative of the function has to be nonzero at the actual root $x^*$. 

11
2.5 Secant Method

One of the main drawbacks of Newton’s method is that it needs to evaluate the derivative at each iteration, which is sometimes complicated and computationally undesirable. The secant method addresses this issue and hence is sometimes called Quasi-Newton’s Method. Interestingly though, this method was developed much earlier than Newton’s method. The formula for secant method can be given as

\[ x^{(k)} = x^{(k-1)} - \frac{f(x^{(k-1)})(x^{(k-1)} - x^{(k-2)})}{f(x^{(k-1)}) - f(x^{(k-2)})}, \quad f(x^{(k-1)}) - f(x^{(k-2)}) \neq 0. \]

2.6 Chord Method

Chord method is another variation of Newton’s method, which also addresses the issue of calculating the derivative at each iteration. This method calculates the derivative at the initial approximation and does not update the value at future iterations. So, the formula can be given by,

\[ x^{(k)} = x^{(k-1)} - \frac{f(x^{(k-1)})}{f'(x^{(0)})}, \quad f'(x^{(0)}) \neq 0. \]

Though the method is computationally desirable, the order of convergence might be something to worry about. We are going to employ this method in an acceleration technique.

2.7 Rate of Convergence

Here we present the formal definition of rate or order of convergence. Suppose \( \{p^{(k)}\}_{k=0}^{\infty} \) is a sequence that converges to \( p \), with \( p^{(k-1)} \neq p \) for all \( k \). If positive
constants $\lambda$ and $\alpha$ exist with

$$\lim_{k \to \infty} \frac{|p^{(k)} - p|}{|p^{(k-1)} - p|^\alpha} = \lambda,$$

then $\{p^{(k-1)}\}_{k=0}^\infty$ is said to converge to $p$ with an order of $\alpha$ and with an asymptotic error constant $\lambda$.[2]

- If $\alpha = 1$ with $\lambda < 1$, the sequence is said to be linearly convergent.
- If $\alpha = 2$, the sequence is said to be quadratically convergent.
- If $\alpha = 3$, the sequence is said to be cubically convergent and so on.

It is possible though to have order of convergence which is not a natural number.

Generally, a higher order convergent sequence converges faster than a lower order convergent sequence. Also, with the same order of convergence the speed may vary depending on the asymptotic error constant $\lambda$. Smaller value of asymptotic error constant corresponds with higher speed of convergence. However, the significance of asymptotic error constant is lower than the order of convergence. We would want all the sequences to converge as fast as possible. But most of the methods we are investigating in this thesis usually show linear or quadratic convergence. In some cases we will be able to get better than quadratic convergence.

For demonstration we have chosen the following function:

$$f(x) = (x - 1)(x - 3)^2(x - 5)(x - 7)^2(x - 9) \quad (2.3)$$

The function in Equation 2.3 has five distinct real zeros, three of them are simple zeros and two of them are of multiplicity two. In the next table, we present the approximation at each step for different zeros of the function when Newton’s method is applied. Among the five distinct zeros, 7 and 9 are actually mirror images of 1 and
3 respectively. So, it will be sufficient to analyze only the zeros 1, 3 and 5.

Table 2.1: Newton’s method for zeros of different multiplicity

<table>
<thead>
<tr>
<th>I</th>
<th>$x^* = 1$</th>
<th>$\lambda_{\text{quadratic}}$</th>
<th>$x^* = 3$</th>
<th>$\lambda_{\text{linear}}$</th>
<th>$x^* = 5$</th>
<th>$\lambda_{\text{cubic}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
<td>2.5</td>
<td></td>
<td>4.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.79006</td>
<td>0.83976</td>
<td>2.73086</td>
<td>0.538279</td>
<td>5.21267</td>
<td>1.70136</td>
</tr>
<tr>
<td>2</td>
<td>0.947549</td>
<td>1.19006</td>
<td>2.85709</td>
<td>0.530986</td>
<td>4.98847</td>
<td>1.19841</td>
</tr>
<tr>
<td>3</td>
<td>0.995762</td>
<td>1.54043</td>
<td>2.92573</td>
<td>0.51969</td>
<td>5.</td>
<td>1.1252</td>
</tr>
<tr>
<td>4</td>
<td>0.99997</td>
<td>1.69341</td>
<td>2.96203</td>
<td>0.511223</td>
<td>5.</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>1.</td>
<td>1.00000</td>
<td>2.98079</td>
<td>0.50602</td>
<td>5.</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1.</td>
<td>0.0</td>
<td>2.99033</td>
<td>0.503122</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>2.99515</td>
<td>0.501591</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>2.99757</td>
<td>0.500803</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>2.99878</td>
<td>0.500403</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>2.99939</td>
<td>0.500202</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>2.99977</td>
<td>0.500101</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>2.99985</td>
<td>0.500051</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>2.99992</td>
<td>0.500025</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>2.99996</td>
<td>0.500013</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>2.99998</td>
<td>0.500006</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>2.99999</td>
<td>0.500003</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$I$ = iteration number, $x^*$ = zero of the function and $\lambda$ = approximation to the asymptotic error constant.

We started with the initial approximation 0.5, 2.5 and 4.5 to converge to the zeros 1, 3 and 5 respectively. We chose the stopping criterion to be $|x^{(k)} - x^{(k-1)}| \leq 10^{-6}$, $k = 1, 2, 3, \ldots$. From the above table we can see that to converge to the zero 1, Newton’s method takes six iterations. The approximation to the asymptotic error constant
became zero at the sixth iteration because the quantity $|x^{(6)} - x^{(5)}| < \text{macheps}$ (anything less than macheps is usually treated by the machine as zero). "macheps" is the abbreviated form of Machine Epsilon, which is defined as the smallest positive number $\tau$ such that $1 + \tau > 1$ on the computer in question [5]. So, in the eyes of the computer the method did not move from the earlier iteration. Because of this, we will consider 1.70823 as the approximation to the asymptotic error constant. This is a simple zero and hence as we expected, Newton’s method shows quadratic convergence with an asymptotic error constant of approximately 1.70823.

However, the second zero ($x^* = 3$) has a multiplicity of two. In this case Newton’s method takes 16 iterations to meet the stopping criterion. It shows linear convergence with an asymptotic error constant approximately 0.5. Theoretically, this is exactly what we expect. If a function $f$ has a zero $p$ of multiplicity $m$, then Newton’s method converges to $p$ with a linear order of convergence with asymptotic error constant $\frac{m-1}{m}$. In this specific case, the multiplicity of the zero 3 is two, and hence the asymptotic error constant should be $\frac{2-1}{2} = 0.5$. We observe that as the multiplicity of the zero increases, the asymptotic error constant for linear convergence $\frac{m-1}{m}$ increases to one, which implies that the speed of convergence decreases. If we exclude the case of simple zeros, then the lowest multiplicity is two. So, Newton’s method will converge linearly to a zero with multiplicity greater than or equals to two of a function with an asymptotic error constant at least 0.5.

Finally, the zero at $x^* = 5$ has a special characteristic. This demonstrates that Newton’s method can show better than quadratic convergence in case of zeros with certain characteristics. This zero is a simple one and at the same time corresponds to an inflection point of the curve. In this particular case, Newton’s method shows cubic convergence with an asymptotic error constant approximately 1.1252. Really, what made the zero special is that $f'(x^*) \neq 0, f''(x^*) = 0$ and $f'''(x^*) \neq 0$. The idea is extended in the next paragraph.
Newton's method is actually a fixed-point iteration method with iteration function \( g(x) = x - \frac{f(x)}{f'(x)} \). In special cases the convergence of fixed-point iteration method can have higher order. These cases are described in Theorem 2.6 of [1] as follows:

**Theorem 2.4**

Let \( x^* \) be a fixed point of \( g(x) \), and \( g(x) \) be \( k \) times continuously differentiable for all \( x \) near \( x^* \), for some \( k \geq 2 \). In addition, suppose

\[
g'(x^*) = g''(x^*) = \cdots = g^{(k-1)}(x^*) = 0 \quad \text{and} \quad g^{(k)}(x^*) \neq 0
\]

Then for an initial approximation \( x^{(0)} \) sufficiently close to \( x^* \), the fixed point iteration method has order of convergence \( k \) with asymptotic error constant \( \lambda = \left| \frac{g^{(k)}(x^*)}{k!} \right| \).

Now, \( g'(x) = \frac{f(x)f''(x)}{f'(x)^2} \). At the zero \( x^* \), \( f(x^*) = 0 \). So, \( g'(x^*) = 0 \) (since this is a simple zero \( f'(x^*) \neq 0 \)). This tells us that for a simple zero, Newton's method will have at least quadratic convergence. It might have better convergence depending on the conditions \( g(x) \) and its derivatives satisfy. Let us examine the zero \( x^* = 5 \) in Equation 2.3.

In this case, we have \( f(5) = 0 \), \( f'(5) = -256 \), \( f''(5) = 0 \) and \( f'''(5) = 864 \). Now, we use these values to evaluate the derivative of \( g \). We see that \( g'(5) = g''(5) = 0 \) and \( g'''(5) = -\frac{27}{4} \). By the above theorem, we can infer that for this particular zero, we should have cubic convergence with asymptotic error constant \( \lambda = \left| \frac{g'''(5)}{3!} \right| = \left| -\frac{27}{6} \right| = 1.125 \). The numeric results we showed in the above table for the zero \( x^* = 5 \) shows that the order of convergence is cubic and the asymptotic error constant \( \lambda = 1.125 \), which is what we expect from theory.

The condition is given on the iteration function \( g \) and its derivatives considering Newton's method as a Fixed-Point Iteration method. We can extend this theorem for Newton's method only and pose conditions on \( f \) and its derivatives instead, which might be simpler to calculate in most of the cases. We have calculated the derivatives of \( g \) generally using Mathematica and the results are shown below. The derivatives
have the following general form \( g^{(k)}(x) = \frac{\text{Terms involving } f(x), f'(x), f''(x), \ldots f^{(k+1)}(x)}{f'(x)^{2k}} \). To see the pattern in general form we do not cancel any factor. The boldfaced derivatives on the left hand sides of the equations are highlighted just for clarity. The iteration function \( g \) and its derivatives:

\[
g(x) = x - \frac{f(x)}{f'(x)}
\]

\[
g'(x) = \frac{f(x)f''(x)}{f'(x)^3}
\]

\[
g''(x) = \frac{f(x)f^{(3)}(x)f'(x)^2}{f'(x)^8} + \frac{f'(x)^3f''(x)}{f'(x)^4} - \frac{2f(x)f'(x)f''(x)}{f'(x)^4}
\]

\[
g'''(x) = \frac{f(x)f^{(3)}(x)f''(x)}{f'(x)^8} + \frac{2f(x)f^{(3)}(x)f''(x)}{f'(x)^8} + \frac{3f'(x)^6f''(x)^2}{f'(x)^8} + \frac{6f(x)f'(x)^4f''(x)^3}{f'(x)^8} - \frac{6f(x)f^{(3)}(x)f'(x)^5f''(x)}{f'(x)^8}
\]

\[
g^{(iv)}(x) = \frac{f(x)f^{(5)}(x)f'(x)^{14}}{f'(x)^{16}} + \frac{3f^{(5)}(x)f'(x)^{15}}{f'(x)^{16}} - \frac{6f(x)f^{(3)}(x)^2f'(x)^{13}}{f'(x)^{16}} + \frac{12f(x)^3f'(x)^3f''(x)}{f'(x)^{16}} - \frac{24f(x)f'(x)^{11}f''(x)^4}{f'(x)^{16}} - \frac{8f(x)f^{(4)}(x)f'(x)^{12}f''(x)}{f'(x)^{16}} - \frac{14f(x)f^{(3)}(x)f'(x)^{14}f''(x)}{f'(x)^{16}} + \frac{36f(x)f^{(3)}(x)f'(x)^{12}f''(x)}{f'(x)^{16}}
\]

\[
g^{(v)}(x) = \frac{f(x)f^{(6)}(x)f'(x)^{30}}{f'(x)^{32}} + \frac{4f^{(5)}(x)f'(x)^{31}}{f'(x)^{32}} - \frac{20f(x)^2f'(x)^{30}}{f'(x)^{32}} - \frac{60f(x)^2f''(x)^4}{f'(x)^{32}} + \frac{120f(x)f'(x)^{29}f''(x)^5}{f'(x)^{32}} - \frac{10f(x)f^{(5)}(x)f'(x)f''(x)^2}{f'(x)^{32}} - \frac{20f(x)f^{(5)}(x)f'(x)f''(x)^2}{f'(x)^{32}} - \frac{25f^{(4)}(x)f'(x)^{30}f''(x)^2}{f'(x)^{32}} - \frac{60f(x)f^{(4)}(x)f'(x)^{28}f''(x)^2}{f'(x)^{32}} + \frac{100f(x)^2f'(x)^{29}f''(x)^2}{f'(x)^{32}} + \frac{90f(x)^2f'(x)f''(x)^2}{f'(x)^{32}} - \frac{240f(x)^2f'(x)^{27}f''(x)^3}{f'(x)^{32}}
\]

\[
g^{(vi)}(x) = \frac{f(x)f^{(7)}(x)f'(x)^{62}}{f'(x)^{64}} + \frac{5f^{(6)}(x)f'(x)^{63}}{f'(x)^{64}} - \frac{20f(x)f^{(4)}(x)^2f''(x)^{61}}{f'(x)^{64}} + \frac{90f(x)f^{(3)}(x)^3f'(x)^{60}}{f'(x)^{64}} + \frac{360f(x)^5f''(x)^{59}f''(x)^2}{f'(x)^{64}} - \frac{720f(x)^6f''(x)^{58}f''(x)^2}{f'(x)^{64}} - \frac{12f(x)^6f''(x)^{57}f''(x)^2}{f'(x)^{64}} - \frac{30f(x)^6f''(x)^{57}f''(x)^2}{f'(x)^{64}} - \frac{85f(x)^6f''(x)^{56}f''(x)^2}{f'(x)^{64}} + \frac{210f(x)^6f''(x)^{56}f''(x)^2}{f'(x)^{64}} - \frac{39f(x)^6f''(x)^{56}f''(x)^2}{f'(x)^{64}} + \frac{90f(x)^6f''(x)^{56}f''(x)^2}{f'(x)^{64}} + \frac{330f(x)^6f''(x)^{57}f''(x)^2}{f'(x)^{64}} + \frac{780f(x)^6f''(x)^{56}f''(x)^2}{f'(x)^{64}} - \frac{1080f(x)^6f''(x)^{57}f''(x)^2}{f'(x)^{64}} + \frac{180f(x)^6f''(x)^{58}f''(x)^2}{f'(x)^{64}} + \frac{360f(x)^6f''(x)^{58}f''(x)^2}{f'(x)^{64}}
\]

At the zero \( x^* \), we have \( f(x^*) = 0 \). So, the first derivative of the iteration function \( g'(x^*) \) is going to be zero. This is the reason we expect Newton’s method to have at least quadratic convergence for any zero with multiplicity one. If \( g''(x^*) \neq 0 \), then the convergence is only quadratic. However, if \( g''(x^*) = 0 \) as well, we can expect at least cubic convergence.
Now, if we look at the expression \( g''(x^*) \) and apply what we know so far (i.e. \( f(x^*) = 0 \) and \( f'(x^*) \neq 0 \)), it is going to be simplified to

\[
g''(x^*) = \frac{f'(x^*) f''(x^*)}{f(x^*)^3} \\
\implies g''(x^*) = \frac{f''(x^*)}{f'(x^*)}.
\]

So it is clear that for \( g''(x^*) \) to be zero, we must have \( f''(x^*) = 0 \). So we can infer that if \( f(x^*) = 0 \), \( f'(x^*) \neq 0 \) and \( f''(x^*) = 0 \), we should have at least cubic convergence. Now, if \( g''(x^*) \neq 0 \), then the convergence will be cubic and not better than that.

Let us continue in the same fashion and analyze the expression of \( g'''(x^*) \). Applying \( f(x^*) = 0 \), \( f'(x^*) \neq 0 \), \( f''(x^*) = 0 \), we get,

\[
g'''(x^*) = \frac{2f'(x^*) f''(x^*)}{f(x^*)^4} \\
\implies g'''(x^*) = \frac{2f'(x^*)}{f(x^*)}.
\]

Again, we infer that \( g'''(x^*) \) will be nonzero if \( f''(x^*) \neq 0 \). On the other hand, \( g'''(x^*) = 0 \) if and only if \( f''(x^*) = 0 \). A statement can be made that if \( f(x^*) = 0 \), \( f'(x^*) \neq 0 \), \( f''(x^*) = 0 \) and \( f'''(x^*) \neq 0 \) then Newton’s method will have cubic convergence. This trend continues and we can actually make a general conjecture here.

But, before doing that let’s analyze what the terms of the higher order derivatives of \( g \) generally look like and what happens to them when evaluated at the zero \( x^* \).

In our analysis, the second term of the derivatives are going to be the key. Comparing the second terms of each of the derivatives of \( g \) (except for \( g'(x) \)), we can see that there is a pattern. So, the second term of \( g^{(n+1)}(x) \) can be generalized as,

\[
\frac{1}{f'(x)^2} \left( n \cdot f^{(n+1)}(x) f'(x)^{2n+1-1} \right).
\]

When we evaluate this at \( x^* \), we can conclude that it is nonzero if and only if \( f^{(n+1)}(x^*) \) is nonzero (we know that \( f'(x^*) \) is nonzero). In fact, this is the only term in \( g^{(n+1)}(x^*) \),
which is nonzero. All other terms are zero at $x^*$, because they involve at least one of the $f(x)$, $f''(x)$, $f'''(x), \ldots, f^{(n)}(x)$. It may be mentioned here that we will analyze $g^{(n+1)}(x)$ only when $g'(x^*) = g''(x^*) = \cdots = g^{(n)}(x^*) = 0$. If $g'(x^*) = g''(x^*) = \cdots = g^{(n-1)}(x^*) = 0$ and $g^{(n)}(x^*) \neq 0$, we will stop our analysis and make a conclusion by Theorem 2.4. So, from our above analysis, we can conclude that to get $g^{(n)}(x^*) = 0$, we need to have $f(x^*) = f''(x^*) = f'''(x^*) = \cdots = f^{(n-1)}(x^*) = f^{(n)}(x^*) = 0$. So, basically we can write $g^{(n+1)}(x)$ as follows,

$$g^{(n+1)}(x) = \cdots + \frac{n \cdot f^{(n+1)}(x)f'(x)(2^{(n+1)-1})}{f'(x)2^{(n+1)-1}} + \cdots$$

We can write this generally for $n$,

$$g^{(n)}(x) = \cdots + \frac{(n-1) \cdot f^{(n)}(x)f'(x)(2^{(n)-1})}{f'(x)2^{(n)-1}} + \cdots$$

Evaluating at $x^*$ we get,

$$g^{(n)}(x^*) = \frac{(n - 1) \cdot f^{(n)}(x^*)f'(x^*)2^{n-1}}{f'(x^*)^{2n}}$$

$$g^{(n)}(x^*) = \frac{(n - 1) \cdot f^{(n)}(x^*)}{f'(x^*)}$$

So, it is clear that if $f^{(n)}(x^*) \neq 0$, then $g^{(n)}(x^*) \neq 0$. Obviously, $f'(x^*) \neq 0$ and $f(x^*) = f''(x^*) = f'''(x^*) = \cdots = f^{(n-1)}(x^*) = 0$. So, we can conclude from Theorem 2.4 that the fixed-point scheme represented by Newton’s iteration function is going to converge to $x^*$ with an order of convergence $n$ and an asymptotic error constant $|\frac{g^{(n)}(x^*)}{(n)!}|$. Now, we formally present the conditions on $f$ in the following corollary.

**Corollary 2.4.1**

If $f(x)$ has a simple zero at $x^*$, i.e. $f(x^*) \neq 0$, and $f(x^*) = f''(x^*) = f'''(x^*) = \cdots = f^{(n-1)}(x^*) = 0$ and $f^{(n)}(x^*) \neq 0$ then Newton’s method, with iteration function $g(x) = x - \frac{f(x)}{f'(x)}$, will converge to $x^*$ with an order of convergence $n$. Also the asymptotic error constant ($\lambda$) can be given by $|\frac{(n-1)f^{(n)}(x^*)}{n!f'(x^*)}|$. 

19
2.8 Acceleration of Convergence of Newton’s Method

We have seen in Corollary 2.4.1 that depending on the derivatives of $f$ evaluated at the root, we might be able to get better than quadratic convergence. However, generally Newton’s method is supposed to give us only quadratic convergence for a simple zero and linear convergence for a zero with higher multiplicity. There are several ways we can accelerate the convergence of different iterative methods. Aitken’s $\Delta^2$ method and Steffensen’s method (which is an updated version of Aitken’s $\Delta^2$ method) are very good examples of such methods [2]. We can also introduce a few modifications to the iteration function of Newton’s method to accelerate the convergence. In this section, we are going to present a few such modifications.

In Theorem 10.2.4 of [7] one such method (let us name it multi-step Newton’s method) is given for vector valued functions, which we can translate for the single variable case in the following manner,

**Theorem 2.5**

Let $f : D \subset \mathbb{R} \to \mathbb{R}$ be differentiable in an open interval $S = (x^* - \epsilon, x^* + \epsilon) \subset D$ and satisfy,

$$\|f'(x) - f'(x^*)\| \leq \gamma \|x - x^*\|, \forall x \in S.$$ 

Assume further, that $f(x^*) = 0$ and that $f'(x^*) \neq 0$. Then $x^*$ is a point of attraction and the interval $S$ is a region of attraction of the following modified iteration function of Newton’s method, which shows at least cubic convergence,

$$x^{(k)} = x^{(k-1)} - \frac{f(x^{(k-1)})}{f'(x^{(k-1)})}.$$  \hspace{1cm} (2.4)

We have applied this technique to different sample problems like $f(x) = (x - 1)^2$, $f(x) = (x - 1)(x - 2)(x - 3)$, $f(x) = \sin x$ and so on. $f(x) = (x - 1)^2$ is a function with a zero of multiplicity two. The original Newton’s method \(x^{(k)} = x^{(k-1)} - \frac{f(x^{(k-1)})}{f'(x^{(k-1)})}\)
gives us linear convergence for this problem with an asymptotic error constant $\frac{1}{2}$. In this case, the iteration function 2.4 fails to accelerate the convergence and the approximation to the asymptotic error constant seems the same. This is expected because the problem does not satisfy the required condition $f'(x^*) \neq 0$ mentioned in Theorem 2.5, so we can not apply the theorem. However, the second function satisfies all the requirements provided in Theorem 2.5 and we see a very good improvement as the result of the mentioned theorem suggests. This function has three zeros, among which $x^* = 1$ and $x^* = 3$ give quadratic convergence and the zero $x^* = 2$ gives us cubic convergence with original Newton’s iteration function. Here, we present the numerical results for the function $f(x) = (x - 1)(x - 2)(x - 3)$.

Table 2.2: Acceleration of Convergence by modification of Newton’s Method

<table>
<thead>
<tr>
<th>$I$</th>
<th>$x^* = 1$</th>
<th>$\lambda_{\text{Quadratic}}$</th>
<th>$x^* = 1$</th>
<th>$\lambda_{\text{Cubic}}$</th>
<th>$x^* = 2$</th>
<th>$\lambda_{\text{Cubic}}$</th>
<th>$x^* = 2$</th>
<th>$\lambda_{\text{Quintic}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Newton’s</td>
<td>Accelerated Newton’s</td>
<td>Newton’s</td>
<td>Accelerated Newton’s</td>
<td>Newton’s</td>
<td>Accelerated Newton’s</td>
<td>Newton’s</td>
<td>Accelerated Newton’s</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.4</td>
<td>2.4</td>
<td>3</td>
<td>4.49514</td>
<td>2.00801</td>
<td>2.0</td>
</tr>
<tr>
<td>1</td>
<td>0.545455</td>
<td>0.454545</td>
<td>0.692985</td>
<td>0.307015</td>
<td>1.75385</td>
<td>3.84615</td>
<td>2.19854</td>
<td>19.3883</td>
</tr>
<tr>
<td>2</td>
<td>0.848953</td>
<td>0.731066</td>
<td>0.962555</td>
<td>1.29396</td>
<td>2.03646</td>
<td>2.44432</td>
<td>2.00237</td>
<td>7.69665</td>
</tr>
<tr>
<td>3</td>
<td>0.974674</td>
<td>1.11005</td>
<td>0.999806</td>
<td>3.68726</td>
<td>1.999999</td>
<td>2.000801</td>
<td>2.000001</td>
<td>5.99968</td>
</tr>
<tr>
<td>4</td>
<td>0.999092</td>
<td>1.41635</td>
<td>0.49514</td>
<td>1.0</td>
<td>2.00801</td>
<td>2.000001</td>
<td>2.000001</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>0.999999</td>
<td>1.49683</td>
<td>1.0</td>
<td>0.0</td>
<td>2.00801</td>
<td>2.000001</td>
<td>2.000001</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>1.0</td>
<td>1.49998</td>
<td>1.0</td>
<td>0.0</td>
<td>2.00801</td>
<td>2.000001</td>
<td>2.000001</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>1.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>2.00801</td>
<td>2.000001</td>
<td>2.000001</td>
<td>0.0</td>
</tr>
</tbody>
</table>

$I$ = Iteration number, $x^*$ = zero of the function and $\lambda$ = approximation to the asymptotic error constant.

We choose $x^{(0)} = 0$ as the initial approximation for the zero $x^* = 1$ and $x^{(0)} = 2.4$ for $x^* = 2$. The stopping criterion is set to $|x^{(k)} - x^{(k-1)}| \leq 10^{-6}$. In the numerical results, we can see that the original Newton’s method converges to the zero $x^* = 1$ quadratically with an approximation to the asymptotic error constant $\lambda = 1.49998$. By Theorem 2.4 the asymptotic error constant is supposed to be $\frac{f''(1)}{2f'(1)} = \frac{3}{2}$, which seems reasonable. With the modification, the method converges cubically with an approximate asymptotic error constant $\lambda = 4.49514$. In this case, the expected asymptotic error constant by Theorem 2.4 is $\frac{9}{2}$, which is again very close to what we
observe in the numerical results.

For the root \( x^* = 2 \), we get cubic convergence by Newton’s method without any modification, as \( f(x) \) has an inflection point at 2, which implies \( f''(x) = 0 \). Theorem 2.4 tells us that we should get an asymptotic error constant of 2 and our approximation is 1.99995. When we apply the modification, we see a quintic convergence with the approximation to the asymptotic error constant 5.99668, which we calculated theoretically to be 6. The numerical results coupled with analytical result inferred from Theorem 2.4 is pretty convincing that we can achieve a significant improvement in the convergence rate with the use of iteration scheme presented by Equation 2.4. Though the scheme is not appropriate for the problems with zeros of higher multiplicity, it is a good improvement for simple zeros considering that the scheme needs to evaluate the derivative \( f'(x) \) only once and reuse it.

Now, let us analyze the iteration function analytically. The first three derivatives of the iteration function \( g(x) \) from Equation 2.4 can be given by,

\[
g'(x) = \frac{1}{f'(x)}(f(x - \frac{f(x)}{f'(x)})f'(x) + f(x)(f'(x) - f'(x - \frac{f(x)}{f'(x)})))f''(x)
\]

\[
g''(x) = \frac{1}{f'(x)}(f'(x)^3f''(x) + 4f(x)f'(x - \frac{f(x)}{f'(x)})f''(x)f''(x)^2 - f(x)^2f''(x)^2f''(x - \frac{f(x)}{f'(x)})
+ f'(x)^3(f^{(3)}(x)(f(x- \frac{f(x)}{f'(x)})+f(x)) - f'(x- \frac{f(x)}{f'(x)})f''(x)) - f'(x)^3(2f(x- \frac{f(x)}{f'(x)})f''(x)^2 + f(x)(2f''(x)^2 + f^{(3)}(x)f'(x - \frac{f(x)}{f'(x)})))
\]

\[
g'''(x) = \frac{1}{f'(x)^2}(2f^{(3)}(x)f'(x)^6 + 9f(x)^2f'(x)f''(x)^3f''(x - \frac{f(x)}{f'(x)}) + 3f'(x)^3(2f(x - \frac{f(x)}{f'(x)})f''(x)^3 + f(x)f''(x)(2f''(x)^2 + 4f^{(3)}(x)f'(x - \frac{f(x)}{f'(x)}) - f''(x)f''(x - \frac{f(x)}{f'(x)})))
- 3f(x)f'(x)^2f''(x)(6f'(x - \frac{f(x)}{f'(x)})f''(x)^2 + f(x)f^{(3)}(x)f''(x - \frac{f(x)}{f'(x)})) - f(x)^3f^{(3)}(x - \frac{f(x)}{f'(x)})f''(x)^3 + f'(x)^5(-3f''(x)^2 + f^{(4)}(x)(f(x- \frac{f(x)}{f'(x)})+f(x)) - 2f^{(3)}(x)f'(x - \frac{f(x)}{f'(x)})) + f'(x)^4(f'(x - \frac{f(x)}{f'(x)})(6f''(x)^2 - f(x)f^{(4)}(x)) - 6f^{(3)}(x)(f(x - \frac{f(x)}{f'(x)}) + f(x)))f''(x)))
\]

At \( x = x^* \) the function \( f(x^*) = 0 \), so evaluating the derivatives at \( x^* \), we get,

\[
g'(x^*) = g''(x^*) = 0 \quad \text{and} \quad g'''(x^*) = \frac{3f''(x^*)^2}{f'(x^*)^2}.
\]
So, clearly Theorem 2.4 implies that if $f'(x^*) \neq 0$ using the modification mentioned in Equation 2.4 should give us at least cubic convergence. If $f''(x^*) \neq 0$, we should get cubic convergence with an asymptotic error constant $\frac{f''(x^*)^2}{2f'(x^*)^2}$. These results agree with our numerical results.

We have analyzed higher order derivatives of the iteration function $g(x)$ using Mathematica and evaluated them at $x = x^*$. The result is as follows,

If $f''(x^*) = 0$ and $f^{(3)}(x^*) \neq 0$ then $g'(x^*) = g''(x^*) = g^{(3)}(x^*) = g^{(4)}(x^*) = 0$ and $g^{(5)}(x^*) = \frac{20f^{(3)}(x^*)^2}{f'(x^*)^2}$. So, we can expect quintic convergence with an asymptotic error constant $\frac{f^{(3)}(x^*)^2}{6f'(x^*)^2}$.

If $f^{(3)}(x^*) = 0$ and $f^{(4)}(x^*) \neq 0$ then $g'(x^*) = g''(x^*) = g^{(3)}(x^*) = g^{(4)}(x^*) = g^{(5)}(x^*) = g^{(6)}(x^*) = 0$ and $g^{(7)}(x^*) = \frac{105f^{(5)}(x^*)^2}{f'(x^*)^2}$. So, we should have convergence of order seven with an asymptotic error constant $\frac{f^{(4)}(x^*)^2}{48f'(x^*)^2}$.

If $f^{(4)}(x^*) = 0$ and $f^{(5)}(x^*) \neq 0$ then $g'(x^*) = g''(x^*) = g^{(3)}(x^*) = g^{(4)}(x^*) = g^{(5)}(x^*) = g^{(6)}(x^*) = g^{(7)}(x^*) = g^{(8)}(x^*) = 0$ and $g^{(9)}(x^*) = \frac{504f^{(5)}(x^*)^2}{f'(x^*)^2}$. So, we should have convergence of order nine with an asymptotic error constant $\frac{f^{(5)}(x^*)^2}{720f'(x^*)^2}$.

Advancing this way, we see that the order of the convergence for the modified Newton’s method seems to follow the rule $2q - 1$, $q = 2, 3, 4, ...$, where $q$ is the order of convergence for the basic Newton’s method. This trend continues for at least $q = 8$.

A more general approach of this scheme was also presented in NR 10.2-2 in [7] to obtain higher order convergence. Analyzing the iteration function tells us that we actually perform two Newton’s steps with the same derivative $f'(x)$. In other words, we update the derivative $f'(x)$ at every two iterations. This involves a similar idea used in the Chord method, which was described in section 2.6. The chord method basically calculates the derivative $f'(x)$ only once and carries out all the iterations with the same value of $f'(x)$. Though this helps to reduce the computational cost, it fails to achieve higher order of convergence. However, using this idea, evaluating the derivative only occasionally we can achieve a very good improvement in the speed of
convergence. The method can be given by,

\[ x^{(k,0)} = x^{(k)}, \quad x^{(k,i)} = x^{(k,i-1)} - \frac{f(x^{(k,i-1)})}{f'(x^{(k,i-1)})}, \quad i = 1, 2, 3, \ldots, m + 1, \quad x^{(k+1)} = x^{(k,m+1)}. \]

This is an \( m \)-step method, in which \( m \) original Newton’s steps are taken between each Newton’s step with updated derivatives. This method has convergence of order \( m+2 \).

Now we present another modification mentioned in [7] in NR 10.2-3, which is applicable to the problems with linear convergence. This is basically a composition of Newton’s method with itself. If we consider the iteration function of Newton’s method is \( g(x) \), then the iteration function of this modification would simply be,

\[ x^{(k)} = g(g(x^{(k-1)})) = g(x^{(k-1)} - \frac{f(x^{(k-1)})}{f'(x^{(k-1)})}) \]

\[ = x^{(k-1)} - \frac{f(x^{(k-1)})}{f'(x^{(k-1)})} \left( \frac{f(x^{(k-1)})}{f'(x^{(k-1)})} \right). \]

In [7], the authors mentioned that composition of two iterative processes \( g_1(x) \) and \( g_2(x) \), where \( \|g_i(x) - x^*\| \leq \gamma_i \|x - x^*\|^p_i \) for \( i = 1, 2, 3, \ldots \) in a neighborhood of \( x^* \) should produce a process with order of convergence at least \( p_1p_2 \). In this thesis we are looking at only Newton’s method. In this case, we are supposed to get an order of convergence at least \( \alpha^2 \), where \( \alpha \) is the order of convergence of the original Newton’s iteration scheme \( g(x) \).

Let us look at a numerical results obtained by this iteration scheme for the function \( f(x) = (x - 1)(x - 3)^2(x - 5)(x - 7)^2(x - 9) \).
Table 2.3: Composition of Newton’s Method for different multiplicity of zeros

<table>
<thead>
<tr>
<th>( I )</th>
<th>( x^* = 1 )</th>
<th>( \lambda_{Quartic} )</th>
<th>( x^* = 3 )</th>
<th>( \lambda_{Linear} )</th>
<th>( x^* = 5 )</th>
<th>( \lambda_{Nonic} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
<td>2.5</td>
<td>4.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.947549</td>
<td>0.839223</td>
<td>2.85709</td>
<td>0.285819</td>
<td>4.98847</td>
<td>5.90189</td>
</tr>
<tr>
<td>2</td>
<td>0.99997</td>
<td>4.01836</td>
<td>2.96203</td>
<td>0.265678</td>
<td>5.0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.  0.</td>
<td>2.99033</td>
<td>0.25459</td>
<td>5.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.  0.</td>
<td>2.99757</td>
<td>0.251198</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>2.99939</td>
<td>0.250303</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>2.99985</td>
<td>0.250076</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>2.99996</td>
<td>0.250019</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>2.99999</td>
<td>0.250005</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>3.  0.25</td>
<td>0.250001</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>3.  0.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>3.  0.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\( I \) = iteration number, \( x^* \) = zero of the function and \( \lambda \) = approximation to the asymptotic error constant.

We have already seen the effect of original Newton’s method on this function. For the zero \( x^* = 1 \), the Newton’s method converged quadratically with an approximation to the asymptotic error constant 1.70823. To meet the stopping criterion \(|x^{(k)} - x^{(k-1)}| \leq 10^{-6}\) it took six iterations. Here, with the composition of Newton’s method it took only three iterations to hit the same stopping criterion and converged with quartic order with an asymptotic error constant approximately 4.01836.

For \( x^* = 3 \), we obtained linear convergence by the original Newton’s method with an approximation to the error constant 0.50003 (Theoretically we were supposed to get 0.5). The number of required iterations to meet stopping criterion was 16, while the composition of Newton’s method stopped after 11 iterations. This is definitely a very good improvement, even though both methods converge linearly. The acceleration is reflected in the asymptotic error constant as it seems to be approaching 0.25. We suspect this is because 0.5 × 0.5 = 0.25. We will try to see if this is the case analytically same way we did for original Newton’s method.

Let us consider that \( f(x) \) has a zero \( p \) of multiplicity \( m \), then we can define the function as \( f(x) = h(x)(x - p)^m \), where \( h(x) \neq 0 \) at \( x = p \). The Newton’s iteration
function composed with itself can be given by,

\[ g(x) = x - \frac{f(x)}{f'(x)} - \frac{f(x) - f'(x)}{f'(x) - f''(x)} = x - \frac{f(x)}{f'(x)} - \frac{f(x) - f'(x)}{f'(x) - f''(x)}. \]

For the above function \( f(x) \), the iteration scheme can be written as,

\[
g(x) = x + \frac{h(x)(p-x)}{(x-p)h'(x) + mh(x)} - \frac{mh(x)(p-x)h''(x)}{(x-p)h'(x) + mh(x)} + \frac{mh(x)(p-x)h''(x)}{(x-p)h'(x) + mh(x)}.
\]

The derivative of \( g(x) \) is a very large fraction. For simple display, let us consider the numerator is \( A \) and the denominator is \( B \).

\[
A = h(x)h'(x((p+(m-1)x)h(x) + x(p-x)h'(x)) + (p-x)((p-x)h''(x) - 2mh'(x))((m - 1)mh(x) + (p-x)((p-x)h''(x) - 2mh'(x))((m - 1)mh(x) + (p-x)((p-x)h''(x) - 2mh'(x)) - 2m^2h'((p+(m-1)x)h(x) + x(p-x)h'(x))h(x) + (p-x)2h'(x)^2((m - 1)mh((p+(m-1)x)h(x) + x(p-x)h'(x)) + (p-x)(p-x)h''((p+(m-1)x)h(x) + x(p-x)h'(x)) - 2mh'((p+(m-1)x)h(x) + x(p-x)h'(x))))\]

\[
B = (mh(x) + (p-x)h'(x))(p-x)h'(x)((p-x)h''((p+(m-1)x)h(x) + x(p-x)h'(x)) - mh((p+(m-1)x)h(x) + x(p-x)h'(x)) + (p-x)(p-x)h''((p+(m-1)x)h(x) + x(p-x)h'(x)) - (m - 1)(p-x)h''((p+(m-1)x)h(x) + x(p-x)h'(x)))^2.
\]

Evaluating \( A \) and \( B \) at \( x = p \), we get \( A = (m - 1)^2 \) and \( B = m^2 \). So, we can see that for a zero of higher multiplicity, if the original Newton's method converges with asymptotic error constant \( \lambda \) then the modified (composition) Newton's method converges with an asymptotic error constant of \( \lambda^2 \).

Finally, for \( x^* = 5 \) we obtained cubic convergence, while with the modification mentioned above, we can achieve convergence of order 9 for \( x^* = 5 \). Clearly we are getting improved rates of convergence. It might be worthwhile to analyze the benefit we gain in terms of computational cost we have to incur. We can see that a linearly convergent iteration scheme does not have any order improvement even though it improves the speed (smaller asymptotic error constant). When the original Newton's method converges quadratically, the composite Newton's method
gives us an improvement at least to quartic order, however, we have to double the efforts because of composition. For the functions, which already have better than quadratic convergence, we see a very good improvement in the order of convergence. As the order of convergence for the composite Newton’s method follows a geometric progression, we can conclude that the better iteration scheme we start with the better order of convergence we can expect from the composition.

Our final modification to Newton’s method is a very interesting one. This is a problem given as an exercise in [2] in exercise set 2.4. The iteration function is given as $g(x) = x - \frac{mf(x)}{f'(x)}$. This fairly simple looking iteration function is supposed to accelerate the convergence of zeros with higher multiplicity. This iteration function works for the zeros, whose multiplicity is more than one and known beforehand. First we will analyze if this is true analytically, and then provide a numerical results.

Let us consider that $f(x)$ has a zero $p$ of multiplicity $m$, then same as earlier analysis we choose to write the function as $f(x) = h(x)(x - p)^m$, where $h(x) \neq 0$ at $x = p$. So, with modification Newton’s iteration function becomes,

$$g(x) = x - \frac{mf(x)}{f'(x)}.$$ 

For the function $g(x)$ can be written as

$$g(x) = \frac{x(x-p)h'(x)+mph(x)}{(x-p)h'(x)+mh(x)}.$$  

$$g'(x) = \frac{(p-x)(mh(x)((p-x)h''(x)-2h'(x))-(m-1)(p-x)h'(x)x)}{((x-p)h'(x)+mh(x))^2}.$$  

At $x = p$ this evaluates to zero i.e. $g'(p) = 0$. So, we clearly see that we must have at least quadratic convergence. To exactly find out what order of convergence this method has, we look at the second derivative.

$$g''(x) = \frac{1}{(mh(x)+(x-p)h'(x))^3}((m(m(2h'(x) + (p-x)((p-x)h''(x) - 4h''(x)))))h(x)^2 - (p - x)(-4mh'(x)^2 + (p-x)(3mh''(x) + (p-x)h''(x))h'(x) - 2(p-x)^2h''(x)^2h(x) + (p - x)^2h'(x)^2(2mh'(x) + (x-p)h''(x)))].$$
Evaluating at \( x = p \) we get \( g''(p) = \frac{2h'(p)}{mh(p)} \). So, by Theorem 2.4 we are certain that we will get at least quadratic convergence with an asymptotic error constant of \( \lambda = \frac{|g''(p)|}{2!} = \frac{h'[p]}{mh[p]} \). Now let’s look at an example to see if this holds. This time we will look at the function \( f(x) = (x - 1)^2(x - 2)^5(x - 10) \).

Table 2.4: Acceleration of Convergence by modification of Newton’s Method

<table>
<thead>
<tr>
<th>I</th>
<th>( x^* = 1 )</th>
<th>( \lambda_{\text{Linear}} )</th>
<th>( x^* = 1 )</th>
<th>( \lambda_{\text{Quadratic}} )</th>
<th>( x^* = 2 )</th>
<th>( \lambda_{\text{Linear}} )</th>
<th>( x^* = 2 )</th>
<th>( \lambda_{\text{Quadratic}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>2.45763</td>
<td>0.508475</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0.217391</td>
<td>0.782609</td>
<td>0.434783</td>
<td>0.565217</td>
<td>4.49153</td>
<td>0.830508</td>
<td>5</td>
<td>2.45763</td>
</tr>
<tr>
<td>2</td>
<td>0.400452</td>
<td>0.766089</td>
<td>0.727289</td>
<td>0.853633</td>
<td>4.07453</td>
<td>0.832633</td>
<td>2.04663</td>
<td>0.222644</td>
</tr>
<tr>
<td>3</td>
<td>0.552754</td>
<td>0.745972</td>
<td>0.903187</td>
<td>1.30176</td>
<td>3.72874</td>
<td>0.833316</td>
<td>2.00076</td>
<td>0.351187</td>
</tr>
<tr>
<td>4</td>
<td>0.677249</td>
<td>0.721643</td>
<td>0.982154</td>
<td>1.90401</td>
<td>3.4402</td>
<td>0.833094</td>
<td>2</td>
<td>0.374585</td>
</tr>
<tr>
<td>5</td>
<td>0.776416</td>
<td>0.692744</td>
<td>0.999234</td>
<td>2.40387</td>
<td>3.19875</td>
<td>0.832239</td>
<td>2</td>
<td>0.372566</td>
</tr>
<tr>
<td>6</td>
<td>0.85252</td>
<td>0.659619</td>
<td>0.999999</td>
<td>2.54866</td>
<td>2.99592</td>
<td>0.830913</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.907989</td>
<td>0.623883</td>
<td>1</td>
<td>2.55552</td>
<td>2.82585</td>
<td>0.829228</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.945832</td>
<td>0.588716</td>
<td>1</td>
<td>0</td>
<td>2.6832</td>
<td>0.827274</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>0.999999</td>
<td>0.500005</td>
<td></td>
<td>2.02985</td>
<td>0.802646</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>0.999999</td>
<td>0.500003</td>
<td></td>
<td>2.02394</td>
<td>0.802145</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>0.999999</td>
<td>0.500001</td>
<td></td>
<td>2.01919</td>
<td>0.801735</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td></td>
<td>2.00001</td>
<td>0.800001</td>
<td>2</td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td></td>
<td>2.00001</td>
<td>0.8</td>
<td>2</td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td></td>
<td>2.00001</td>
<td>0.8</td>
<td>2</td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td></td>
<td>2.00001</td>
<td>0.8</td>
<td>2</td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

I = iteration number, \( x^* \) = zero of the function and \( \lambda \) = approximation to the asymptotic error constant.

We choose \( x^{(0)} = 0 \) as the initial approximation for the zero \( x^* = 1 \) and \( x^{(0)} = 5 \) for \( x^* = 2 \). The stopping criterion is set to \( |x^{(k)} - x^{(k-1)}| \leq 10^{-6} \). The original Newton’s method needed 25 iterations for the zero \( x^* = 1 \) and 64 iterations for the zero \( x^* = 2 \) to reach the stopping criterion. The error constant seems to approach 0.5 and 0.8, as we suggested theoretically. In the numerical results, we can clearly see that this modified Newton’s method converges quadratically for both zeros with asymptotic error constant \( \lambda = 2.55552 \) and \( \lambda = 0.372566 \) approximately. Theoretically we were supposed to get 2.55556 for the zero \( x^* = 1 \) and 0.375 for the zero \( x^* = 2 \). Both the
numerical results seem to be very close to what we expected theoretically.

2.9 Newton’s Method for Minimization

Beside solving equations or systems of equations, Newton’s method can be applied to minimization problems as well. Really, this is another root finding problem in disguise. We know that the first derivative of a continuously differentiable function $f(x)$ is equal to zero at the minimizer, which implies we need to find the zeros of $f’(x)$. This we can achieve by the iteration scheme,

$$x^{(k)} = x^{(k-1)} - \frac{f’(x^{(k-1)})}{f''(x^{(k-1)})}, \text{ where } f''(x^{(k-1)}) \neq 0.$$ 

So, for a sufficiently close initial approximation, Newton’s method will converge to a zero of $f’(x)$. However, $f’(x) = 0$ does not guarantee a minimizer. This only implies that at the point $x = x^*$ where $f’(x) = 0$ we have a stationary point. To guarantee that we have a minimizer at $x = x^*$, we need $f''(x^*) > 0$. However, at times we can have a minimizer when $f''(x^*) = 0$ as well. Guaranteeing if a zero $x^*$ of $f’(x)$ is a minimizer, maximizer, or an inflection point needs more conditions when $f''(x^*) = 0$. Also, even if the zero is a minimizer, it is practically impossible to know if it is a global minimum of a function [5]. However, we are only interested in finding a local minimizer, so our study is going to revolve around how we can make sure that we are converging to a minimizer and not a maximizer and how we can guarantee convergence of the method. We are going to analyze the iteration function to see if we can pose some conditions so that the method converges to the minimizer only.

We can rewrite the iteration equation as follows,

$$x^{(k)} - x^{(k-1)} = -\frac{f’(x^{(k-1)})}{f''(x^{(k-1)})}$$

$$\Rightarrow (x^{(k)} - x^{(k-1)})^2 = -\frac{f’(x^{(k-1)})(x^{(k)} - x^{(k-1)})}{f''(x^{(k-1)})}. \quad (2.5)$$
The left hand side of the equation is clearly positive or zero. So, the numerator on the right hand side should be zero or have opposite sign as the denominator. i.e. $f'(x^{(k-1)})(x^{(k)} - x^{(k-1)}) < 0$ if and only if $f''(x^{(k-1)}) > 0$. The numerator becomes zero when we are at the zero of $f'(x)$. Let us analyze the case where the numerator is less than zero.

$$f'(x^{(k-1)})(x^{(k)} - x^{(k-1)}) < 0 \iff (x^{(k)} - x^{(k-1)}) > 0 \text{ when } f'(x^{(k-1)}) < 0 \text{ and } (x^{(k)} - x^{(k-1)}) < 0 \text{ when } f'(x^{(k-1)}) > 0$$

$$\implies x^{(k)} > x^{(k-1)} \text{ when } f''(x^{(k-1)}) < 0 \text{ and } x^{(k)} < x^{(k-1)} \text{ when } f''(x^{(k-1)}) > 0.$$  

This implies if $f$ is increasing at $x^{(k-1)}$, our next approximation is towards the left (in the opposite direction of a possible maximum), this is also true when $f$ is decreasing, our next approximation is towards the right (decreasing function, so we might have a minimum to the right).

Similarly, $f'(x^{(k-1)})(x^{(k)} - x^{(k-1)}) > 0$ in Equation 2.5 if and only if $f''(x^{(k-1)}) < 0$.

Now, $f'(x^{(k-1)})(x^{(k)} - x^{(k-1)}) > 0$ implies $f'(x^{(k-1)})$ and $x^{(k)} - x^{(k-1)}$ have the same sign i.e. if $f$ is increasing, our next approximation goes to the right towards a possible maximum and if $f$ is decreasing, our next approximation goes to the left towards a possible maximum. Finally if $f''(x^{(k-1)}) = 0$, the method fails. So, basically, the second derivative at an approximation dictates which way our next iteration is going to move. If $f''(x^{(k-1)}) > 0$ at $x^{(k-1)}$, the iteration scheme is clearly moving towards the direction of a minimum.

So, we can conclude that there should be an interval $D$ around $x^*$ with $f'(x^*) = 0$ such that the function will have a minimizer at $x^*$ if $f''(x) > 0 \forall x \in D$ except possibly at $x^*$.

The question may arise though as to when do we stop iterating. In this case, since we are looking for a minimizer, our goal is to continue iterating as long as
$f(x^{(k)}) < f(x^{(k-1)})$. However, the method might jump through the minimizer without reaching the point, which may also produce $f(x^{(k)}) \geq f(x^{(k-1)})$. To avoid that we could check if $|f'(x^{(k)})| \leq \text{tolerance}$. There could be a scale issue as well in this approach, which can be avoided by using $f'(x^{(k)}) x^{(k)}$ instead of $f'(x^{(k)})$ [5]. Besides, $|x^{(k)} - x^{(k-1)}| \leq \text{tolerance}$ gives us a way of checking if the iterations are almost at the same place. It is possible to use more than one stopping criterion for a better result.

For example, $f(x) = x^4$ has a minimum at $x = 0$. Note that $f''(x) = 12x^2$. By our analysis, Newton’s method should converge to the minimum if we start with an initial approximation $x \in D$ such that $f''(x) > 0$ for all $x \in D$ i.e. $12x^2 > 0$, $\implies x^2 > 0$. So, we see that for any real value of $x$ except at $x = 0$, we should converge to the minimizer $x = 0$, which also tells us that the minimizer is a global one.
Chapter 3

Systems of Equations with Several Variables

In the last chapter, we have discussed different methods to solve nonlinear equations. Some of these methods can also be applied to solve nonlinear systems of equations. In this chapter, we will discuss different methods to solve systems of equations with one or more variable. Our main focus will be cases involving more than one variable. However the same technique can be readily applied to the single variable problems as well.

Generally, a system of nonlinear equations is approximated by converting it to a system of linear equations if possible. However, at times we need more direct approach to solve these types of problems. Luckily, the methods in one variable have variants for solving systems of equations with several variables. The first method we are going to look at is the fixed point method.

3.1 Fixed Point Method for Several Variables

Let us consider a system of nonlinear equations:

\[ f_1(x_1, x_2, \ldots, x_n) = 0 \]
\[ f_2(x_1, x_2, \ldots, x_n) = 0 \]
\[ \vdots \]
\[ f_n(x_1, x_2, \ldots, x_n) = 0. \]

Each of the functions \( f_i \) can be thought of a component function of the vector valued function \( \vec{F}(x_1, x_2, \ldots, x_n) \). The \( x_i \)'s also can be expressed as \( \vec{x} \). So the above
A system of equations can be expressed as

\[ \vec{F}(\vec{x}) = \vec{0}. \]

Similar to the one variable case, a function \( \vec{G} \) from \( D \subset \mathbb{R}^n \) into \( \mathbb{R}^n \) has a fixed point at \( \vec{p} \in D \) if \( \vec{G}(\vec{p}) = \vec{p} \).

The following theorem [2] extends the fixed point theorem for one variable to the \( n \)-variable case.

**Theorem 3.1 (A special case of Contraction Mapping Theorem)**

Let \( D = \{(x_1, x_2, \ldots, x_n)^t|a_i \leq x_i \leq b_i, \text{ for each } i = 1, 2, \ldots, n\} \) for some collection of constants \( a_1, a_2, \ldots, a_n \) and \( b_1, b_2, \ldots, b_n \). Suppose \( G \) is a continuous function from \( D \subset \mathbb{R}^n \) into \( \mathbb{R}^n \) with the property that \( \vec{G}(\vec{x}) \in D \) whenever \( \vec{x} \in D \). Then \( \vec{G} \) has a fixed point in \( D \). Moreover, suppose that all the component functions of \( \vec{G} \) have continuous partial derivatives and a constant \( K < 1 \) exists with

\[ |\frac{\partial g_i(\vec{x})}{\partial x_j}| \leq \frac{K}{n}, \text{ whenever } \vec{x} \in D \]

for each \( j = 1, 2, \ldots, n \) and each component function \( g_i \). Then the sequence \( \{\vec{x}(k)\}_{k=0}^{\infty} \) defined by an arbitrarily selected \( \vec{x}(0) \) in \( D \) and generated by

\[ \vec{x}(k) = \vec{G}(\vec{x}(k-1)), \text{ for each } k \geq 1 \]

converges to the unique fixed point \( \vec{p} \in D \) and

\[ \|\vec{x}(k) - \vec{p}\|_\infty \leq \frac{K^k}{1-K} \|\vec{x}(1) - \vec{x}(0)\|_\infty. \]

**3.2 Newton’s Method for Several Variables**

We have seen how Newton’s method works for solving equations of one variable. Now we are going to analyze the method for systems of nonlinear equations in more than one variable. It turns out that we can represent the method for the multivariable
case without loss of generality. The iteration function of the method in single-variable case is given by

\[ x^{(k)} = x^{(k-1)} - \frac{f(x^{(k-1)})}{f'(x^{(k-1)})}. \]

For several variable case we can translate this iteration function as follows:

\[ \vec{x}^{(k)} = \vec{x}^{(k-1)} - J(\vec{x}^{(k-1)})^{-1} \vec{F}(\vec{x}^{(k-1)}). \] (3.1)

Here, \( \vec{x} = (x_1, x_2, x_3, \ldots, x_n)^T \), \( \vec{F}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), f_3(\vec{x}), \ldots, f_n(\vec{x}))^T \) and the \( J(\vec{x}) \) = Jacobian matrix of \( \vec{F}(\vec{x}) \). To elaborate, let us consider the following system of nonlinear equations,

\[
\begin{align*}
f_1(x_1, x_2, \ldots, x_n) &= 0 \\
f_2(x_1, x_2, \ldots, x_n) &= 0 \\
&\vdots \\
f_n(x_1, x_2, \ldots, x_n) &= 0.
\end{align*}
\]

We represent the whole system by the shorthand \( \vec{F}(\vec{x}) = \vec{0} \). The corresponding iteration function can be presented as,

\[
\begin{pmatrix}
x_1^{(k)} \\
x_2^{(k)} \\
\vdots \\
x_n^{(k)}
\end{pmatrix} = 
\begin{pmatrix}
x_1^{(k-1)} \\
x_2^{(k-1)} \\
\vdots \\
x_n^{(k-1)}
\end{pmatrix} - 
\begin{pmatrix}
\frac{\partial f_1(x^{(k-1)})}{\partial x_1} & \frac{\partial f_1(x^{(k-1)})}{\partial x_2} & \cdots & \frac{\partial f_1(x^{(k-1)})}{\partial x_n} \\
\frac{\partial f_2(x^{(k-1)})}{\partial x_1} & \frac{\partial f_2(x^{(k-1)})}{\partial x_2} & \cdots & \frac{\partial f_2(x^{(k-1)})}{\partial x_n} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial f_n(x^{(k-1)})}{\partial x_1} & \frac{\partial f_n(x^{(k-1)})}{\partial x_2} & \cdots & \frac{\partial f_n(x^{(k-1)})}{\partial x_n}
\end{pmatrix}^{-1} \times 
\begin{pmatrix}
f_1(\vec{x}^{(k-1)}) \\
f_2(\vec{x}^{(k-1)}) \\
\vdots \\
f_n(\vec{x}^{(k-1)})
\end{pmatrix}.
\] (3.2)

In the one-variable cases we have presented a nice little picture showing how each iteration is progressing. The tangent line had a big part to play and we could easily visualize what is going on. But, in the cases of multivariable equations and systems this is not as simple as the single variable cases \( (\mathbb{R} \rightarrow \mathbb{R}) \). It would have been pretty amazing to see or even imagine what the graphs look like and how they interact with
each other. But, as humans we do not possess the capability to exactly visualize higher than two dimensional systems ($\mathbb{R}^2 \rightarrow \mathbb{R}^2$). However, we can visualize some aspects of three dimensional systems. Also, we can employ different techniques in Mathematica to get a good sense what is going on in three dimensional problems ($\mathbb{R}^3 \rightarrow \mathbb{R}^3$). Anything higher than that we have to make use of projections and make sense from the mathematics involved. It is interesting to see that there are certain analogies that can be translated from the one-dimensional case to two and three dimensional cases, which we can then use to make sense of higher dimensional systems. We will try to establish the analogies in this chapter.

When we deal with a two dimensional system, we require a system of two equations in two variables, where each of the equations involves a function of the form $f_i(x_1, x_2)$. Each of the functions represents a surface on the three dimensional space and each of the equations $f_i(x_1, x_2) = 0$ represents the intersection of the $x_1x_2$ plane with the surface, which is a curve on the $x_1x_2$ plane. So, basically we are interested in the intersection(s) of two such curves, which is/are the solution(s) of the system.

To solve a system of two variables, we start with an initial approximation $(x_1^{(0)}, x_2^{(0)})$, we draw a tangent plane at $(x_1^{(0)}, x_2^{(0)}, f_i(x_1^{(0)}, x_2^{(0)}))$ to the surface. We then take the intersection of the tangent plane with the $x_1x_2$—plane, which produces a line. Two tangent planes will produce two lines on the $x_1x_2$—plane and their intersection will be the next approximation of Newton’s method. It is clear that there are certain similarities between the one and two dimensional cases in terms of how the new approximation is obtained. The same analogy should be applicable to all other dimensions as well.

Let us first illustrate the progression symbolically, then we will give an example with numerical results to verify. To analyze generally, we consider the following system of equations,
\[
\begin{pmatrix}
  f_1(x_1, x_2) \\
  f_2(x_1, x_2)
\end{pmatrix} = \begin{pmatrix}
  0 \\
  0
\end{pmatrix}.
\] (3.3)

The first Taylor polynomials of these two component functions expanded around our initial approximation \((x_1^{(0)}, x_2^{(0)})\) give us two functions, which are the tangent planes to the two surfaces \(f_i(x_1, x_2)\). We express the intersection of these planes with the \(x_1x_2\) plane as the following system of linear equations, which represent the two lines on the \(x_1x_2\) plane,

\[
\begin{align*}
  f_1(x_1^{(0)}, x_2^{(0)}) + (x_1 - x_1^{(0)}) \frac{\partial f_1(x_1^{(0)}, x_2^{(0)})}{\partial x_1} + (x_2 - x_2^{(0)}) \frac{\partial f_1(x_1^{(0)}, x_2^{(0)})}{\partial x_2} &= 0, \\
  f_2(x_1^{(0)}, x_2^{(0)}) + (x_1 - x_1^{(0)}) \frac{\partial f_2(x_1^{(0)}, x_2^{(0)})}{\partial x_1} + (x_2 - x_2^{(0)}) \frac{\partial f_2(x_1^{(0)}, x_2^{(0)})}{\partial x_2} &= 0.
\end{align*}
\]

As we mentioned earlier, the intersection of these two lines is supposed to give us the new approximation \((x_1^{(1)}, x_2^{(1)})\). Solving these two equations for \(x_1\) and \(x_2\), we get,

\[
\begin{align*}
  x_1 &= x_1^{(0)} - \frac{f_1 \times \frac{\partial f_2}{\partial x_2} - f_2 \times \frac{\partial f_1}{\partial x_2}}{\frac{\partial f_2}{\partial x_1} \times \frac{\partial f_1}{\partial x_2} - \frac{\partial f_1}{\partial x_1} \times \frac{\partial f_2}{\partial x_2}}, \\
  x_2 &= x_2^{(0)} - \frac{f_1 \times \frac{\partial f_2}{\partial x_1} - f_2 \times \frac{\partial f_1}{\partial x_1}}{\frac{\partial f_2}{\partial x_1} \times \frac{\partial f_1}{\partial x_2} - \frac{\partial f_1}{\partial x_1} \times \frac{\partial f_2}{\partial x_2}}.
\end{align*}
\] (3.4) (3.5)

For simplicity we avoided repeatedly writing \((x_1^{(0)}, x_2^{(0)})\), but each of the \(f_1, f_2, \frac{\partial f_1}{\partial x_1}, \frac{\partial f_2}{\partial x_2}\), and \(\frac{\partial f_2}{\partial x_1}\) is evaluated at \((x_1^{(0)}, x_2^{(0)})\).

Let us rewrite Equation 3.4 and 3.5 in the matrix form,

\[
\begin{pmatrix}
  x_1 \\
  x_2
\end{pmatrix} = \begin{pmatrix}
  x_1^{(0)} \\
  x_2^{(0)}
\end{pmatrix} - \begin{pmatrix}
  f_1 \times \frac{\partial f_2}{\partial x_2} - f_2 \times \frac{\partial f_1}{\partial x_2} \\
  f_1 \times \frac{\partial f_2}{\partial x_1} - f_2 \times \frac{\partial f_1}{\partial x_1}
\end{pmatrix} \begin{pmatrix}
  \frac{\partial f_2}{\partial x_1} \times \frac{\partial f_1}{\partial x_2} - \frac{\partial f_1}{\partial x_1} \times \frac{\partial f_2}{\partial x_2}
\end{pmatrix}.
\] (3.6)
So, theoretically these $x_1$ and $x_2$ are the new approximations $x_1^{(1)}$ and $x_2^{(1)}$ obtained by Newton’s method. Now, we will verify if that is true. The Newton’s method corresponding to the system 3.3 can be given by,

\[
\begin{pmatrix}
  x_1^{(k)} \\
  x_2^{(k)}
\end{pmatrix} = \begin{pmatrix}
  x_1^{(k-1)} \\
  x_2^{(k-1)}
\end{pmatrix} - \left( \begin{array}{cc}
  \frac{\partial f_1(x^{(k-1)})}{\partial x_1} & \frac{\partial f_1(x^{(k-1)})}{\partial x_2} \\
  \frac{\partial f_2(x^{(k-1)})}{\partial x_1} & \frac{\partial f_2(x^{(k-1)})}{\partial x_2}
\end{array} \right)^{-1} \begin{pmatrix}
  f_1(x^{(k-1)}) \\
  f_2(x^{(k-1)})
\end{pmatrix}.
\] (3.7)

Rewriting Equation 3.7 to get $x_1^{(1)}$ and $x_2^{(1)}$ we get,

\[
\begin{pmatrix}
  x_1^{(1)} \\
  x_2^{(1)}
\end{pmatrix} = \begin{pmatrix}
  x_1^{(0)} \\
  x_2^{(0)}
\end{pmatrix} - \left( \begin{array}{cc}
  \frac{\partial f_1(x^{(0)})}{\partial x_1} & \frac{\partial f_1(x^{(0)})}{\partial x_2} \\
  \frac{\partial f_2(x^{(0)})}{\partial x_1} & \frac{\partial f_2(x^{(0)})}{\partial x_2}
\end{array} \right)^{-1} \begin{pmatrix}
  f_1(x^{(0)}) \\
  f_2(x^{(0)})
\end{pmatrix}.
\]

\[
\Rightarrow \begin{pmatrix}
  x_1^{(1)} \\
  x_2^{(1)}
\end{pmatrix} = \begin{pmatrix}
  x_1^{(0)} \\
  x_2^{(0)}
\end{pmatrix} - \left( \begin{array}{cc}
  \frac{\partial f_2}{\partial x_2} & \frac{\partial f_1}{\partial x_2} \\
  -\frac{\partial f_2}{\partial x_1} & \frac{\partial f_1}{\partial x_1}
\end{array} \right) \begin{pmatrix}
  \frac{\partial f_1}{\partial x_2} \\
  \frac{\partial f_1}{\partial x_1}
\end{pmatrix} \begin{pmatrix}
  f_1 \\
  f_2
\end{pmatrix}.
\]

Each of the $f_1, f_2, \frac{\partial f_1}{\partial x_1}, \frac{\partial f_1}{\partial x_2}, \frac{\partial f_2}{\partial x_1}$ and $\frac{\partial f_2}{\partial x_2}$ is evaluated at $(x_1^{(0)}, x_2^{(0)}), \text{i.e. at } x^{(0)}$.

\[
\Rightarrow \begin{pmatrix}
  x_1^{(1)} \\
  x_2^{(1)}
\end{pmatrix} = \begin{pmatrix}
  x_1^{(0)} \\
  x_2^{(0)}
\end{pmatrix} - \left( \begin{array}{cc}
  \frac{\partial f_2}{\partial x_2} & -\frac{\partial f_2}{\partial x_1} \\
  \frac{\partial f_1}{\partial x_2} & -\frac{\partial f_1}{\partial x_1}
\end{array} \right) \begin{pmatrix}
  \frac{\partial f_1}{\partial x_2} \\
  \frac{\partial f_1}{\partial x_1}
\end{pmatrix} \begin{pmatrix}
  f_1 \\
  f_2
\end{pmatrix}.
\] (3.8)

So, comparing Equation 3.6 and 3.8, we can say that $x_1$ and $x_2$ in Equation 3.6 are exactly same as $x_1^{(1)}$ and $x_2^{(1)}$ in Equation 3.8. Now let us look at an example. Let us
consider the following pair of functions.

\[ f_1(x_1, x_2) = x_1^2 + x_2^2 - 9 \]
\[ f_2(x_1, x_2) = -x_1 + x_2^2 - 3. \]

The corresponding system of equations is,

\[
\begin{pmatrix}
  x_1^2 + x_2^2 - 9 \\
  -x_1 + x_2^2 - 3
\end{pmatrix}
= \begin{pmatrix} 0 \\ 0 \end{pmatrix}.
\]

The \( x_1x_2 \) plane cuts the graph of \( z = f_1(x_1, x_2) \) in a circle and \( z = f_2(x_1, x_2) \) in a parabola. This circle and parabola intersect in three points \((-3, 0), (2, \sqrt{5})\) and \((2, -\sqrt{5})\), which are the solutions of the above system. To solve this system by Newton’s method, we start with an initial approximation \((-4, -3)\) and draw tangent planes on the surfaces \( z = f_1(x_1, x_2) \) and \( z = f_2(x_1, x_2) \). The first step of Newton’s method in the three dimensional view is shown in the two Figures 3.1 and 3.2 and the actual problem in two dimensional view is shown in the Figures 3.3 and 3.4.

We can see that the black tangent planes intersect the \( x_1x_2 \) plane in two lines. Using the equations for \( x_1 \) and \( x_2 \) we obtained in the previous page, we can find the intersection point of these two lines, which is \((-\frac{22}{7}, -\frac{33}{21})\). This point is our new approximation.

We follow the same procedure to get the next approximation. Among these three solutions, \((2, \sqrt{5})\) and \((2, -\sqrt{5})\) are mirror images with respect to the \( x_1 \) axis and the Jacobian is nonsingular at both the solutions. So, we expect Newton’s method to behave similarly at both of these solutions. Therefore, it will be sufficient to analyze \((-3, 0)\) and \((2, -\sqrt{5})\). We are presenting the numerical results obtained using Mathematica below. For the solution \((-3, 0)\), we take our initial approximation
Figure 3.1: Intersection of $x_1x_2$ plane (gray) with the tangent plane (black) at (-4,-3) on the graph of $f_1(x_1,x_2) = x_1^2 + x_2^2 - 9$ (light gray)

Figure 3.2: Intersection of $x_1x_2$ plane (gray) with the tangent plane (black) at (-4,-3) on the graph of $f_2(x_1,x_2) = -x_1 + x_2^2 - 3$ (light gray)

Figure 3.3: The intersection of $x_1x_2$ plane and $f_1(x_1,x_2) = x_1^2 + x_2^2 - 9$ and $f_2(x_1,x_2) = -x_1 + x_2^2 - 3$, three solutions are (-3, 0), $(2, \sqrt{5})$ and $(2, -\sqrt{5})$

Figure 3.4: The intersection of $x_1x_2$ plane and $f_1(x_1,x_2)$ and $f_2(x_1,x_2)$ and the tangent planes at (-4,-3) on these two surfaces $f_1$ and $f_2$
(-4, -3) and for (2, \sqrt{5}) we start with (4, 3). The stopping criterion we choose for 
both the cases is \|\vec{x}(k+1) - \vec{x}^*\|_2 \leq 10^{-6}. To calculate the asymptotic error constant 
as well we use 2-norm. The formula for the asymptotic error constant can be given by,

\[
\lambda_{Linear} = \lim_{k \to \infty} \frac{\|\vec{x}(k+1) - \vec{x}^*\|_2}{\|\vec{x}(k) - \vec{x}^*\|_2}
\]

\[
\lambda_{Quadratic} = \lim_{k \to \infty} \frac{\|\vec{x}(k+1) - \vec{x}^*\|_2}{\|\vec{x}(k) - \vec{x}^*\|_2^2}
\]

However, for analyzing we will use the following formulas to approximate the 
asymptotic error constants,

\[
\lambda_{Linear} = \frac{\|\vec{x}(k+1) - \vec{x}^*\|_2}{\|\vec{x}(k) - \vec{x}^*\|_2}
\]

\[
\lambda_{Quadratic} = \frac{\|\vec{x}(k+1) - \vec{x}^*\|_2}{\|\vec{x}(k) - \vec{x}^*\|_2^2}
\]

Table 3.1: Newton’s Method for several variables

<table>
<thead>
<tr>
<th>I</th>
<th>(x_1^* = -3)</th>
<th>(x_2^* = 0)</th>
<th>(\lambda_{Linear})</th>
<th>(x_1^* = 2)</th>
<th>(x_2^* = \sqrt{5})</th>
<th>(\lambda_{Quadratic})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-4</td>
<td>-3</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-3.14286</td>
<td>-1.47619</td>
<td>0.468993</td>
<td>2.44444</td>
<td>2.40741</td>
<td>0.10392</td>
</tr>
<tr>
<td>2</td>
<td>-3.00386</td>
<td>-0.736787</td>
<td>0.4968</td>
<td>2.03354</td>
<td>2.24913</td>
<td>0.158656</td>
</tr>
<tr>
<td>3</td>
<td>-3</td>
<td>-0.368392</td>
<td>0.49999</td>
<td>2.00022</td>
<td>2.23616</td>
<td>0.181429</td>
</tr>
<tr>
<td>4</td>
<td>-3</td>
<td>-0.184196</td>
<td>0.5</td>
<td>2</td>
<td>2.23607</td>
<td>0.186321</td>
</tr>
<tr>
<td>5</td>
<td>-3</td>
<td>-0.0920979</td>
<td>0.5</td>
<td>2</td>
<td>2.23607</td>
<td>0.</td>
</tr>
<tr>
<td>6</td>
<td>-3</td>
<td>-0.046049</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-3</td>
<td>-0.0230245</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
</tr>
<tr>
<td>19</td>
<td>-3</td>
<td>-5.62121 \times 10^{-6}</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>-3</td>
<td>-2.81061 \times 10^{-6}</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>-3</td>
<td>-1.40530 \times 10^{-6}</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>-3</td>
<td>-7.02651 \times 10^{-7}</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(I = \) iteration number, \(x_1^* \) and \(x_2^* \) are coordinates of the solution and 
\(\lambda = \) approximations to the asymptotic error constant.

From the numerical results we see that to meet the stopping criterion in the case of
the solution \((-3, 0)\) we needed 22 iterations, which is a bit slow. The approximation to the asymptotic error constant suggests that the convergence is linear. In the case of single variable problems, we saw that Newton’s method is supposed to give us at least quadratic convergence, if the zero is simple. If the zero has multiplicity more than one, we obtained linear convergence. Zeros of higher multiplicity means that the first derivative of the function is zero at the root. In the case of multivariables, the analogous case of the zero derivative would be the singular Jacobian matrix. The Jacobian is the matrix of partial derivatives of the functions with respect to all the variables. So, the Jacobian matrix of the problem in question is,

\[
\begin{pmatrix}
2x_1 & 2x_2 \\
-1 & 2x_2
\end{pmatrix}, \tag{3.10}
\]

Evaluating the Jacobian at \((-3, 0)\) and \((2, \sqrt{5})\) give us,

\[
\begin{pmatrix}
-6 & 0 \\
-1 & 0
\end{pmatrix} \text{ and } \begin{pmatrix}
4 & 2\sqrt{5} \\
-1 & 2\sqrt{5}
\end{pmatrix}.
\]

Clearly, the first matrix is singular because the determinant is zero and also because all the columns are not independent. However, the second matrix is nonsingular, which is analogous to the nonzero derivative or simple zero in the single variable case. Comparing these with the numerical results makes perfect sense now. For the second solution we obtained quadratic convergence, which we are supposed to get. Similarly, for the first solution, we obtained linear convergence. Since the Jacobian is singular at this solution, that’s exactly what we expected.

Now we will look at the first solution \((-3, 0)\) component-wise, i.e. what is the speed of convergence along the \(x_1\) and \(x_2\) axes. We present the following numerical results.
Table 3.2: Newton’s Method for several variable component-wise

<table>
<thead>
<tr>
<th></th>
<th>( x_1^* = -3 )</th>
<th>( \lambda_{\text{Quadratic for } x_1^*} )</th>
<th>( x_2^* = 0 )</th>
<th>( \lambda_{\text{Linear for } x_2^*} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-4</td>
<td>0.142857</td>
<td>-3</td>
<td>0.492063</td>
</tr>
<tr>
<td>1</td>
<td>-3.14286</td>
<td>0.189189</td>
<td>-1.47619</td>
<td>0.499114</td>
</tr>
<tr>
<td>2</td>
<td>-3.00386</td>
<td>0.199692</td>
<td>-0.736787</td>
<td>0.499997</td>
</tr>
<tr>
<td>3</td>
<td>-3.</td>
<td>0.199951</td>
<td>-0.184196</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>-3.</td>
<td>0.019995</td>
<td>0.0920979</td>
<td>0.5</td>
</tr>
<tr>
<td>5</td>
<td>-3.</td>
<td>0.</td>
<td>-0.046049</td>
<td>0.5</td>
</tr>
<tr>
<td>6</td>
<td>-3.</td>
<td>0.</td>
<td>-0.0230245</td>
<td>0.5</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**I** = iteration number, \( x_1^* \) and \( x_2^* \) are coordinates of the solution and \( \lambda \) = approximations to the asymptotic error constant.

From these numerical results we see that the speed of convergence is not linear for both components, rather for \( x_1^* \) the speed is quadratic. But, when we approximate the asymptotic error constant for Newton’s method for the whole system, then the slowest component becomes dominant. It is also interesting to see that the asymptotic error constant of the dominant component becomes the general asymptotic error constant.

Now let us analyze the region of convergence or the region of attraction. Looking at the Newton’s iteration function we can immediately conclude that the Jacobian matrix evaluated at the initial approximation should be non-singular. Otherwise, the method will definitely fail. We need to avoid the regions where the Jacobian is singular. To find out where the Jacobian is singular, we look at the determinant of the matrix 3.10. The determinant is \( 4x_1x_2 + 2x_2 \). The Jacobian is singular where \( 4x_1x_2 + 2x_2 = 0 \).

Solving this equation we get \( x_2 = 0 \) and \( x_1 = -\frac{1}{2} \). So, the Jacobian is singular on these lines. To get convergence, we should avoid choosing points that fall on these lines. But to determine a region of attraction around a solution we present the
Theorem 5.2.1 of [5] for several variables:

**Theorem 3.2**

Let $\vec{F} : \mathbb{R}^n \to \mathbb{R}^n$ be continuously differentiable in an open convex set $D \subset \mathbb{R}^n$. Assume that there exists $\vec{x}^* \in \mathbb{R}^n$ and $r, \beta > 0$ such that $N(\vec{x}^*, r) \subset D, (N$ is the disk around $\vec{x}^*$ with radius $r), \vec{F}(\vec{x}^*) = 0$, $J(\vec{x}^*)^{-1}$ [J is the Jacobian] exists with $\|J(x^*)^{-1}\| \leq \beta$, and $J \in \text{Lip}_\gamma(N(x_*, r))$, which means $J$ is Lipschitz continuous in $N$ with Lipschitz constant $\gamma$. Then there exists $\epsilon > 0$ such that for any initial approximation $x_0 \in N(\vec{x}^*, \epsilon)$ the sequence $x_1, x_2, x_3, \ldots$ generated by $x^{(k)} = x^{(k-1)} - J(x^{(k-1)})^{-1}F(x^{(k-1)})$, $k = 1, 2, 3, \ldots$ is well defined and converges to $x_*$ and obeys $\|x^{(k)} - \vec{x}^*\| \leq \beta \gamma \|x^{(k-1)} - \vec{x}^*\|^2$, $k = 1, 2, 3, \ldots$. $\epsilon$ is chosen as the minimum of $\{r, \frac{1}{2\beta\gamma}\}$.

It is clear from the above theorem that we can not find a region for the solution $(-3, 0)$, because the Jacobian is singular at this point. But we can find a region for the other two solutions. It can be easily shown that the Lipschitz constant is 4 for the entire plane and $\|J(x^*)^{-1}\|_2 = \sqrt{\rho([J^{-1}(x^*)]^T \times [J^{-1}(x^*)])} = 0.2140045769864111$, where $\rho(A) = \text{spectral radius of the matrix } A$. The spectral radius of a matrix is defined as the maximum of the absolute values of the eigenvalues of that matrix. So, the $\epsilon = \min\{\infty, \frac{1}{2\beta}\} = \min\{\infty, \frac{1}{2\times0.2140045769864111\times4}\} = 0.58409965693368$. So, by the theorem we have a region around each of the solutions $(2, \sqrt{5})$ and $(2, -\sqrt{5})$ of radius 0.58409965693368, any initial approximation chosen from this region will have guaranteed convergence of quadratic order to that solution. In practice the region is much wider though.

Now, we present the Ostrowski’s theorem mentioned in [6], which gives sufficient conditions for a solution to be a point of attraction. The theorem says,

**Theorem 3.3**

Assume that $G : \mathbb{R}^n \to \mathbb{R}^n$, where $\vec{G}(\vec{x}) = \vec{x} - J_F(\vec{x})^{-1}\vec{F}(\vec{x})$, $J_F = \text{Jacobian of } \vec{F}$ is differentiable at the fixed point $\vec{x}^*$ (as we can consider Newton’s iteration function
as a fixed-point iteration scheme) and that $\rho(JG(\bar{x}^*)) < 1$, where $\rho = \text{spectral radius}$. Then $\bar{x}^*$ is a point of attraction of the Newton’s iteration scheme.

Here the point of attraction $\bar{x}^*$ refers to the fixed point $\bar{x}^*$ of the iteration scheme $\bar{x}^{(k)} = G(\bar{x}^{(k-1)})$, where $G : \mathbb{R}^n \to \mathbb{R}^n$, if there is an open neighborhood $S$ of $\bar{x}^*$ such that whenever $x^{(0)} \in S$, the iterates $\bar{x}^{(k)}$ are well defined and converge to $\bar{x}^*$ [6].

Now, we apply Theorem 3.3 in analyzing the point of attraction of the system of equations 3.9. The Newton’s iteration scheme omitting the superscripts can be given by,

$$\bar{G}(\bar{x}) = \bar{x} - J^{-1}(\bar{x})\bar{F}(\bar{x}).$$

For this specific system,

$$\bar{G}(\bar{x}) = \begin{pmatrix} 3x_1^2 + 2x_1 - 6 \\ 2x_1 + 1 \\ x_1^2 - 6x_1x_2^2 + 6x_1 - 3x_2^2 + 9 \\ 4x_1x_2 + 2x_2 \end{pmatrix},$$

$$J_G(\bar{x}) = \begin{pmatrix} \frac{2(x_1^2 + x_1 - 6)}{(2x_1 + 1)^2} & 0 \\ \frac{x_1^2 + x_1 - 6}{(2x_1 + 1)^2} & \frac{-x_1^2 + 2x_1(x_2^2 - 3) + x_3^2 - 9}{2(2x_1 + 1)x_2^2} \end{pmatrix}.$$

The determinant of $J_G$ can be given by,

$$\text{Det}(J_G) = \frac{(-x_1^2 + x_1 - 6)((x_1 + 3)^2 - (2x_1 + 1)x_2^2)}{(2x_1 + 1)^3x_2^2}.$$ 

The eigenvalues of $J_G$ are

$$\frac{2(x_1^2 + x_1 - 6)}{(2x_1 + 1)^2} \text{ and } \frac{-x_1^2 + 2x_1(x_2^2 - 3) + x_3^2 - 9}{2(2x_1 + 1)x_2^2}.$$

We have mentioned earlier that spectral radius is defined as the maximum of the absolute values of the eigenvalues. We have applied this property to plot the level curves of $\rho(J_G)$ corresponding to different spectral radii in Figure 3.5 and the region where the absolute value of both eigenvalues are simultaneously less than one in Figure 3.6.
The smallest interior of the two black squares represents the spectral radius 0.1 and as we start to move outwards the spectral radius increases and we showed the lines or collection of lines, where the spectral radius is 0.1, 0.25, 0.5, 0.75, 1.0, 5.0, 10.0, 50.0 and 1000.0 in Figure 3.5.

Theoretically, the regions shown in Figure 3.6 have a good chance of being in the neighborhood of a probable point of attraction. In fact, we should be able to find a neighborhood around each of the solutions $2, \sqrt{5}$ and $2, -\sqrt{5}$, from where an initial approximation should converge to the corresponding solution. However, the condition provided in Ostrowski’s theorem is not a necessary condition for convergence [6].

Recall that the solutions of the system are $(-3, 0), (2, \sqrt{5})$ and $(2, -\sqrt{5})$ and we have a singular Jacobian at the solution $(-3, 0)$. Figure 3.5 indicates that as we move towards the solutions $(2, \sqrt{5})$ and $(2, -\sqrt{5})$, the spectral radius approaches zero. This can be easily verified by looking at the numerators of the eigenvalues, which are zero at these two solutions. Also, as we move towards the lines $x_1 = -\frac{1}{2}$ and $x_2 = 0$, the spectral radius approaches infinity because the Jacobian $J_F$ is singular at these two lines. The denominators of the determinant and the eigenvalues tell us that this is what we can expect as they are zero at $x_1 = -\frac{1}{2}$.

![Figure 3.5: The graph of contour plots of different spectral radii](image1)

![Figure 3.6: The region where the spectral radius is less than one](image2)
Now let us look at the null space and range of the Jacobian at all the solutions. At $(-3, 0)$ the null space is the span of \[
\begin{pmatrix}
0 \\
1
\end{pmatrix},
\] which basically denotes the $x_2$ axis. The range is the span of \[
\begin{pmatrix}
-6 \\
-1
\end{pmatrix},
\] which is the line going through the points $(0, 0)$ and $(-6, -1)$. The dimension of the null space is one and the dimension of the range is one. The significance of the direction of null the space is given in [3] and [4]. We can see from the Table 3.2 that the method shows linear convergence rate in the direction of the null space and quadratic convergence in the orthogonal direction of the null space. In this case the null space and the range are not orthogonal. However, we can see that the null space direction is dominating the convergence order as being the slowest one.

It will be worthwhile to see how Newton’s method follows the same principle in the case of three variables as well. Let us consider the following system of equations

\[
\begin{align*}
    f_1(x_1, x_2, x_3) &= 4x_1^2 + x_2^2 - x_3^2 + 4 = 0 \\
    f_2(x_1, x_2, x_3) &= x_1^2 + x_2^2 + x_3^2 - 4 = 0 \\
    f_3(x_1, x_2, x_3) &= x_2 = 0.
\end{align*}
\]

A three dimensional contour plot of $f_i(\vec{x}) = 0$ is shown in Figure 3.7. They intersect at $(0, 0, 2)$ and $(0, 0, -2)$. We present the numerical results in Table 3.3.
Table 3.3: Newton’s Method for three variables

<table>
<thead>
<tr>
<th>$I$</th>
<th>$x^*_1 = 0$</th>
<th>$x^*_2 = 0$</th>
<th>$x^*_3 = -2$</th>
<th>$\lambda_{\text{Linear}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>10</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>20.5</td>
<td>0</td>
<td>-32.5</td>
<td>3.63871</td>
</tr>
<tr>
<td>2</td>
<td>10.25</td>
<td>0</td>
<td>-16.3115</td>
<td>0.479017</td>
</tr>
<tr>
<td>3</td>
<td>5.125</td>
<td>0</td>
<td>-8.27838</td>
<td>0.460395</td>
</tr>
<tr>
<td>4</td>
<td>2.5625</td>
<td>0</td>
<td>-4.38078</td>
<td>0.431584</td>
</tr>
<tr>
<td>5</td>
<td>1.28125</td>
<td>0</td>
<td>-2.64933</td>
<td>0.410348</td>
</tr>
<tr>
<td>6</td>
<td>0.640625</td>
<td>0</td>
<td>-2.07906</td>
<td>0.449717</td>
</tr>
<tr>
<td>7</td>
<td>0.320313</td>
<td>0</td>
<td>-2.0015</td>
<td>0.496241</td>
</tr>
<tr>
<td>8</td>
<td>0.160156</td>
<td>0</td>
<td>-2.0</td>
<td>0.499994</td>
</tr>
<tr>
<td>9</td>
<td>0.0800781</td>
<td>0</td>
<td>-2</td>
<td>0.5</td>
</tr>
<tr>
<td>10</td>
<td>0.0400391</td>
<td>0</td>
<td>-2</td>
<td>0.5</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>$2.443790 \times 10^{-6}$</td>
<td>0</td>
<td>-2</td>
<td>0.5</td>
</tr>
<tr>
<td>25</td>
<td>$1.221895 \times 10^{-6}$</td>
<td>0</td>
<td>-2</td>
<td>0.5</td>
</tr>
<tr>
<td>26</td>
<td>$6.109476 \times 10^{-7}$</td>
<td>0</td>
<td>-2</td>
<td>0.5</td>
</tr>
</tbody>
</table>

$I =$ iteration number, $x^*_1$, $x^*_2$ and $x^*_3$ are coordinates of the solution and $\lambda =$ approximations to the asymptotic error constant.

In this example we see some interesting behavior. The initial approximation was chosen as $(1, 10, -1)$ and the stopping criterion was $\|\vec{x}^{(k)} - \vec{x}^{(k-1)}\| \leq 10^{-6}$. From the result we see that, $x_1$ and $x_3$ were shot further away from the solution and for $x_2$ it reached 0 in one iteration. Also, the speeds of convergence of $x_1$ and $x_3$ seem

Figure 3.7: The graph of three surfaces and their intersection
to be different. By the eighth iteration it seems that $x_3$ reached $-2$ (at least the
with number of digits shown in Mathematica) where $x_1$ took 26 iterations to meet
the stopping criterion. However, the overall speed and order of convergence of the
method is dominated by the speed of $x_1$, which was expected [3] and [4]. To conclude,
Newton’s method converges to $(0, 0, -2)$ linearly with an asymptotic error constant
0.5.

3.3 Acceleration of Convergence

We have discussed three methods to accelerate the convergence of Newton’s method
in the single variable case. For multivariable systems of equations we are going to
present two of them.

Let us first present the Theorem 10.2.4 of [7] for acceleration of Newton’s method
for vector valued functions.

**Theorem 3.4**

Let $\vec{F} : D \subset \mathbb{R}^n \to \mathbb{R}^n$ be $F$-differentiable in an open ball $S = (x^*, \delta) \subset D$ and satisfy,

$$\|\vec{F}'(\vec{x}) - \vec{F}'(\vec{x}^*)\| \leq \gamma \|\vec{x} - \vec{x}^*\|, \forall \vec{x} \in S.$$

Assume further, that $\vec{F}(\vec{x}^*) = \vec{0}$ and that $\vec{F}'(\vec{x}^*)$ is nonsingular. Then $\vec{x}^*$ is a point
of attraction and the ball $S$ is a region of attraction of the modified Newton’s method
given by the following iteration scheme, which shows at least cubic convergence,

$$\vec{x}^{(k)} = \vec{x}^{(k-1)} - \vec{F}'(\vec{x}^{(k-1)})^{-1} [\vec{F}(\vec{x}^{(k-1)}) + \vec{F}(\vec{x}^{(k-1)} - \vec{F}'(\vec{x}^{(k-1)})^{-1} \vec{F}(\vec{x}^{(k-1)})]].$$  \hspace{1cm} (3.11)

A function $F : D \subset \mathbb{R}^n \to \mathbb{R}^m$ is $F$-differentiable (Frechet differentiable) [7] at
$x \in \text{int}(D)$ if there is a linear operator $A \in L(\mathbb{R}^n, \mathbb{R}^m)$ such that

$$\lim_{h \to 0} (\frac{1}{\|h\|}) \|F(x + h) - F(x) - Ah\| = 0.$$
Here the derivative $\vec{F}'$ can be thought of as the Jacobian matrix $J$ of the function $\vec{F}$.

Now let us present the numerical results for this acceleration technique.

Table 3.4: Acceleration of Newton’s Method for two variables

<table>
<thead>
<tr>
<th>$I$</th>
<th>$x_1^* = 2$</th>
<th>$x_2^* = \sqrt{5}$</th>
<th>$\lambda_{\text{Cubic}}$</th>
<th>$x_1^* = -3$</th>
<th>$x_2^* = 0$</th>
<th>$\lambda_{\text{Linear}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4</td>
<td>3</td>
<td>-4</td>
<td>-3</td>
<td>-1.0668</td>
<td>0.350169</td>
</tr>
<tr>
<td>1</td>
<td>2.17558</td>
<td>2.30407</td>
<td>0.0191876</td>
<td>-3.0379</td>
<td>-1.10668</td>
<td>0.374721</td>
</tr>
<tr>
<td>2</td>
<td>2.00037</td>
<td>2.23621</td>
<td>0.0598522</td>
<td>-3</td>
<td>-0.414941</td>
<td>0.375</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2.23607</td>
<td>0.0693907</td>
<td>-3</td>
<td>-0.155603</td>
<td>0.375</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>-3</td>
<td>-0.058351</td>
<td>0.375</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td>-3</td>
<td>-0.0218816</td>
<td>0.375</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td>-3</td>
<td>-0.0082056</td>
<td>0.375</td>
</tr>
<tr>
<td>:</td>
<td></td>
<td></td>
<td></td>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
<tr>
<td>998</td>
<td></td>
<td></td>
<td></td>
<td>-3.0000773554</td>
<td>0.50885</td>
<td>3.12964</td>
</tr>
<tr>
<td>999</td>
<td></td>
<td></td>
<td></td>
<td>-3.0000242095</td>
<td>3.12964</td>
<td>3.12964</td>
</tr>
</tbody>
</table>

$I = \text{iteration number}, x_1^* \text{ and } x_2^* \text{ are coordinates of the solution and}$

$\lambda = \text{approximations to the asymptotic error constant}.$

We have already seen in Table 3.1 that the original Newton’s method converges quadratically to the solution $(2, \sqrt{5})$ starting with an initial approximation $(4, 3)$ in five iterations with an approximate asymptotic error constant $0.186321$. The stopping criterion was set to $\|\vec{x}^{(k)} - \vec{x}^{(k-1)}\| \leq 10^{-6}$. Using the same initial approximation and stopping criterion, the above modification achieves cubic convergence and the method reaches the stopping criterion in three iterations with an approximation to the error constant $0.0693907$. This result agrees with the theorem and the results in the single variable problems.

For the solution $(-3, 0)$, we started with $(-4, -3)$ as the initial approximation, but after even 999 iterations the method does not seem to converge. The theorem is not applicable for solutions where the Jacobian is singular. So, we just wanted to explore what happens if we apply the method for this solution. Surprisingly, we see a huge degradation in the speed of convergence, which was not the case in single variable case. The round off error may have caused this.

The second acceleration method we looked at was the composition of Newton’s
method to itself. We achieved quite fast convergence rate with this technique in the single variable case. Here is the iteration scheme, as mentioned in NR10.2-3 in [7],

\[
\vec{x}^{(k)} = \vec{x}^{(k-1)} - J(\vec{x}^{(k-1)})^{-1} \vec{F}(\vec{x}^{(k-1)}) - J(\vec{x}^{(k-1)} - J(\vec{x}^{(k-1)})^{-1} \vec{F}(\vec{x}^{(k-1)}))^{-1} \vec{F}(\vec{x}^{(k-1)}) - J(\vec{x}^{(k-1)})^{-1} \vec{F}(\vec{x}^{(k-1)})).
\]

Consider the same system of equations we have been analyzing for two variables.

\[
\begin{align*}
    f_1(x_1, x_2) &= x_1^2 + x_2^2 - 9 \\
    f_2(x_1, x_2) &= -x_1 + x_2^2 - 3.
\end{align*}
\]

Among the three solutions, we analyze \((-3, 0)\) and \((2, \sqrt{5})\) as we expect the other solution \((2, -\sqrt{5})\) to behave like \((2, \sqrt{5})\) because of its symmetry. The acceleration we gain is presented in Table 3.5.

<table>
<thead>
<tr>
<th>(I)</th>
<th>(x_1^* = -3)</th>
<th>(x_2^* = 0)</th>
<th>(\lambda_{Linear})</th>
<th>(x_1^* = 2)</th>
<th>(x_2^* = \sqrt{5})</th>
<th>(\lambda_{Quartic})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-4</td>
<td>-3</td>
<td>2.32996</td>
<td>4</td>
<td>3</td>
<td>0.00171339</td>
</tr>
<tr>
<td>1</td>
<td>-3.00386</td>
<td>-0.736787</td>
<td>0.249995</td>
<td>2.03354</td>
<td>2.24913</td>
<td>0.00631692</td>
</tr>
<tr>
<td>2</td>
<td>-3.184196</td>
<td>-0.046049</td>
<td>0.25</td>
<td>2.03354</td>
<td>2.24913</td>
<td>0.00631692</td>
</tr>
<tr>
<td>3</td>
<td>-3.0115122</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-3.00287806</td>
<td>0.25</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-3.00019537</td>
<td>0.249993</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>-3.000122193</td>
<td>0.625443</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>-3.000526989</td>
<td>4.26366</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>-3.000132525</td>
<td>0.254372</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>-3.000132525</td>
<td>0.254372</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(I\) = iteration number, \(x_1^*\) and \(x_2^*\) are coordinates of the solution and \(\lambda\) = approximations to the asymptotic error constant.

The initial approximations and stopping criteria are the same for this acceleration scheme as well ((-4, -3) for the solution (-3, 0) and (4, 3) for the solution (2, \sqrt{5}), while the tolerance for stopping criterion is \(10^{-6}\)). We can see from Table 3.5 that the order of convergence for (-3, 0) does not improve, while the approximation to the asymptotic error constant mostly jumps around 0.25. Though there are few
fluctuations, we can assume that the round off error might be the reason. This makes sense if we examine the Mathematica code (attached in Appendix) to see the calculations involved. The progression of calculations in the code is as follows:

- Calculate Jacobian $J$ of $\vec{F}$.
- Calculate inverse of the Jacobian $J^{-1}$.
- Calculate basic Newton’s iteration to get an intermediate approximation by
  \[ x_{\text{intermediate}}^{(k-1)} = x^{(k-1)} - J^{-1} \times \vec{F}. \]
- Evaluate $\vec{F}$ and $J$ at $x_{\text{intermediate}}^{(k-1)}$.
- Calculate change in the approximations by
  \[ -J^{-1}(x^{(k-1)}) \times \vec{F}(x^{(k-1)}) - J^{-1}(x_{\text{intermediate}}^{(k-1)}) \times \vec{F}(x_{\text{intermediate}}^{(k-1)}). \]
- Finally, we calculate the next approximation by adding or subtracting the change in approximations to the current approximations.

So, these many calculations, and especially the calculation of inverse matrices, could very easily contaminate the approximations as the number of iterations increases. However, with all this being said, we can clearly see an improvement in the number of iterations from the basic Newton’s method. It took 22 iterations to hit the stopping criterion with an approximate asymptotic error constant 0.5, while it takes 17 iterations with the composite modification and the approximate error constant also seems to be approaching 0.25. Recall that we have a singular Jacobian at the solution $(-3, 0)$ and that is the reason we observed linear convergence in the first place.

For the other solution though, the method seems to hit the stopping criterion in two iterations with an approximate asymptotic error constant 0.00631692. The order of convergence is also improved from quadratic to quartic, which agrees with the theorem presented.
In this case, it is actually impossible to see the asymptotic behavior of the error constant because we only see two iterations. It would have been nice though to have formulas to calculate the asymptotic error constant exactly, just like we did for the single variable case. However, in the case of higher dimensional problems $\mathbb{R}^n \to \mathbb{R}^n$ it is not as easy as the single variable case, because it would require calculations of higher order derivatives of $F$. The Jacobian could be considered as the first order derivative, which can also be considered as a tensor of order two. Similarly, when calculating the second derivative, for each of the component functions $f_i$, we will get a Hessian matrix. So, the second derivative is actually a tensor of order 3. Also, higher order derivatives would mean tensors of higher order. Calculation of resultant/determinants of tensors requires a lot of machine power, which casts doubt on its worthiness.

3.4 Newton’s Method for Minimization

We have presented a condition in the single variable case such that Newton’s method converges to the minimizer only. Now, we are going to see what this translates to higher dimensional cases. Let us consider a twice continuously differentiable function $f : \mathbb{R}^n \to \mathbb{R}$. We want to find a minimizer of the function. In the single variable case we converted this problem to a root finding problem, where we wanted to find the zeros of $f'(x)$. In this case also, the problem is the same as finding the solution of the system of $n$ nonlinear equations in $n$ unknowns represented by,

$$\nabla f(x_1, x_2, \ldots, x_n) = 0,$$

(3.12)

where $\nabla f$ is the gradient of the function $f$. To solve this system, Newton’s iteration scheme can be given by,

$$\bar{x}^{(k)} = \bar{x}^{(k-1)} - [\nabla^2 f(\bar{x}^{(k-1)})]^{-1} \times \nabla f(\bar{x}^{(k-1)}).$$
We need $\nabla^2 f(\bar{x}^{(k-1)})$ to be nonsingular, otherwise the method will fail. Here $\nabla^2 f$ is the Hessian matrix of $f$ and sometimes expressed as $H_f$ as well. Similar to the single variable problem, $\nabla f = 0$ could mean a minimizer, a maximizer, or a saddle point.

To guarantee that we have a minimizer at $\bar{x} = \bar{x}^*$, we need to have $\nabla^2 f(\bar{x}^*)$ to be a positive definite matrix, which is analogous to the $f''(x^*) > 0$ in the single variable problem. This claim is also supported by the discussion in Section 5.5 of [5].

A symmetric $n \times n$ matrix $A$ is called positive definite if $\bar{x}A\bar{x}^T > 0$ for every $n$-dimensional vector $\bar{x} \neq 0$ [2]. One of the useful properties of a positive definite matrix is that all of its leading principal submatrices have positive determinants [2]. A leading principal submatrix of an $n \times n$ matrix $A$ has the following form.

$$
\begin{pmatrix}
  a_{11} & a_{12} & \ldots & a_{1k} \\
  a_{21} & a_{22} & \ldots & a_{2k} \\
  \vdots & \vdots & \ddots & \vdots \\
  a_{k1} & a_{k2} & \ldots & a_{kk}
\end{pmatrix}
$$

for some $1 \leq k \leq n$.

This property makes it easy to test for positive definiteness of a matrix with smaller numbers of rows and columns.

If we knew the zero, we could evaluate the Hessian matrix to determine if the solution is a minimizer or not, but we are trying to solve the problem $\nabla f = 0$ and at the same time trying to guarantee if the solution is a minimizer. So, we approach this problem in the same way as we did for the single variable problem. As we said earlier, the multivariable analogy of $f''(x)$ is the Hessian matrix. So, the higher dimension analogy of the condition can be presented as follows.

Let the first and second partial derivatives exist of a function $f(x_1, x_2, \ldots, x_n)$ and let $\nabla f(\bar{x}) = 0$ at $\bar{x}^*$. Let us assume that there is a region $D$ around $\bar{x}^*$ where $\nabla^2 f(\bar{x})$ is positive definite for all $\bar{x} \in D$ except possibly at $\bar{x}^*$. Then the critical point $x^*$ is a minimizer and for any initial approximation $\bar{x} \in D$ except at $x^*$, and Newton’s
method will converge to the minimizer.

Evaluation of the Hessian matrix is computationally very expensive, so usually it is replaced by approximations calculated in different ways. But, since minimization is not the focus of our thesis we are going to omit discussions on that.

Consider the following minimization problem.

\[ f(x_1, x_2) = x_1^2 + x_2^2 - 9. \]

Now

\[ \nabla f = \begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix}. \]

The Hessian matrix is,

\[ \nabla^2 f = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}. \]

This is clearly a positive definite matrix as the leading principal submatrices are \( \begin{pmatrix} 2 \end{pmatrix} \) and \( \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \). Both of these matrices have positive determinant. So, the Hessian matrix is positive definite independent of \( x_1 \) and \( x_2 \), which implies that for any choice of initial approximation the method should converge to the minimizer \( (0, 0) \). Also, the minimizer is a global minimizer.

Now, let us look at the following minimization problem,

\[ f(\vec{x}) = x_1 x_2 e^{-x_1^2 - x_2^2}. \]

To find the critical points we need to solve the following system of equations,

\[ \nabla f(\vec{x}) = 0 \]

\[ \Rightarrow \begin{pmatrix} x_2 e^{-x_1^2 - x_2^2} - 2x_1 x_2 e^{-x_1^2 - x_2^2} \\ x_1 e^{-x_1^2 - x_2^2} - 2x_1 x_2 e^{-x_1^2 - x_2^2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \]
This system has five solutions \((0, 0), \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right), \left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right), \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)\) and \(\left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)\). Among these five solutions we have two maximizers at \(\left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)\) and \(\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)\), two minimizers at \(\left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)\) and \(\left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)\) and one saddle point at \((0, 0)\). Figure 3.8 shows them clearly.

The Hessian matrix can be given by,

\[
\begin{pmatrix}
 e^{-x_1^2-x_2^2}(4x_1^3x_2 - 6x_1x_2) & e^{-x_1^2-x_2^2}(-2x_1^2 + 4x_2^2x_1^2 + 1 - 2x_2^2)
 \\
 e^{-x_1^2-x_2^2}(-2x_1^2 + 4x_2^2x_1^2 + 1 - 2x_2^2) & e^{-x_1^2-x_2^2}(4x_1x_3^3 - 6x_1x_2)
\end{pmatrix}
\]

As we mentioned earlier, if the Hessian matrix is positive definite in a region around a stationary point, then the stationary point is a minimizer and Newton’s method will converge to that minimizer for any initial approximation in that region. We can actually calculate the region using properties of a positive definite matrix. The Hessian will be the positive definite if and only if,

\[
4x_1^3x_2e^{-x_1^2-x_2^2} - 6x_1x_2e^{-x_1^2-x_2^2} > 0 \quad \text{and} \quad 4x_1^2e^{-x_1^2-2x_2^2} + 20x_1^2x_2^2e^{-2x_1^2-2x_2^2} - e^{-2x_1^2-2x_2^2} + 4x_2^2e^{-2x_1^2-2x_2^2} - 8x_1x_2^4e^{-2x_1^2-2x_2^2} - 8x_1^2x_2^2e^{-2x_1^2-2x_2^2} - 8x_1^2x_2^4e^{-2x_1^2-2x_2^2} > 0.
\]

We have plotted these inequalities in Mathematica and the results are presented in Figures 3.9, 3.10 and 3.11.

We can follow the same procedure to find a region that guarantees convergence to the maximizers, where the Hessian is negative definite. Now let us present a numerical
Figure 3.9: The region where $1 \times 1$ leading principal submatrix of the Hessian has positive determinant

Figure 3.10: The region where $2 \times 2$ leading principal submatrix of the Hessian has positive determinant

Figure 3.11: The region which satisfies both conditions i.e. the Hessian is positive definite
results,

Table 3.6: Newton’s Method for Minimization

<table>
<thead>
<tr>
<th>I</th>
<th>$x_1^*$ = $\frac{1}{\sqrt{2}}$</th>
<th>$x_2^*$ = $-\frac{1}{\sqrt{2}}$</th>
<th>$\lambda_{Quadratic}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
<td>-0.5</td>
<td>0.</td>
</tr>
<tr>
<td>1</td>
<td>0.75</td>
<td>-0.75</td>
<td>707107</td>
</tr>
<tr>
<td>2</td>
<td>0.705224</td>
<td>-0.705224</td>
<td>0.723661</td>
</tr>
<tr>
<td>3</td>
<td>0.707104</td>
<td>-0.707104</td>
<td>0.490696</td>
</tr>
<tr>
<td>4</td>
<td>0.707107</td>
<td>-0.707107</td>
<td>0.499983</td>
</tr>
</tbody>
</table>

$I$ = iteration number, $x_1^*$ and $x_2^*$ are coordinates of the minimizer and
$\lambda$ = approximations to the asymptotic error constant.

For the numerical results in Table 3.6, we started with the initial approximation
$(0.5, -0.5)$ and the stopping criterion was set to $\|\vec{x}^{(k)} - \vec{x}^{(k-1)}\| < 10^{-6}$. We can see
that the method met the stopping criterion after four iterations. In fact the method
converged quadratically with an approximation to the asymptotic error constant
0.499983. Since, the Hessian matrix is non-singular at $(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$ we expected quadratic
convergence. If the Hessian is singular and still we have a minimum then the expected
convergence would be linear.
Chapter 4

Newton’s Method in the Complex Plane

So far we have dealt with solving equations in one variable or systems of nonlinear equations in more than one variable. However, all the roots we have discussed are real. Needless to say that we can have complex roots for equations. Also, we know that if the coefficients of a polynomial equation are real, complex roots occur in pairs where each one of a pair is a complex conjugate of the other. Just like functions of a real variable, Newton’s method is applicable to equations of complex variables. For example, Newton’s method can be used to find the zeros of the function $f(z)$, where $z$ is a complex variable and can be replaced by $x + iy$. Here, we will look at the famous problem known as roots of unity. Let’s first analyze the following problem,

$$f(z) = z^3 - 1 = 0. \tag{4.1}$$

To solve this equation by Newton’s method we can follow two techniques. We can solve the complex-valued function normally using Newton’s method with the iteration function,

$$z^{(k)} = z^{(k-1)} - \frac{f(z^{(k-1)})}{f'(z^{(k-1)})}. \tag{4.2}$$

Or we can convert the equation to a system of nonlinear equations of two variables and solve the system following the same method as Equation 3.3. We can think of $z$ as $x + iy$, where $x$ and $y$ are real numbers. $x$ is the real part of $z$ written as $Re(z)$ and $y$ is the imaginary part of $z$ written as $Im(z)$. Now, the equation $z^3 - 1 = 0$
becomes,
\[(x + iy)^3 - 1 = 0\]
\[\implies (x^3 - 3xy^2 - 1) + i(3x^2y - y^3) = 0.\]

Now, we can actually separate the real and imaginary part of the equations and convert this into a system of nonlinear equations as follows:
\[
\begin{pmatrix}
  x^3 - 3xy^2 - 1 \\
  3x^2y - y^3
\end{pmatrix}
= \begin{pmatrix}
  0 \\
  0
\end{pmatrix}.
\]
(4.3)

We will see if solving the Equation 4.1 and solving the System 4.3 are equivalent. We know that the cube roots of unity are 1, \(-\frac{1}{2} + i\frac{\sqrt{3}}{2}\) and \(-\frac{1}{2} - i\frac{\sqrt{3}}{2}\). Let’s analyze the zeros 1 and \(-\frac{1}{2} + i\frac{\sqrt{3}}{2}\). Now if we solve this using Equation 4.2, Newton’s method should converge to 1 and \(-\frac{1}{2} + i\frac{\sqrt{3}}{2}\). For the initial approximation we choose \(z^{(0)} = 3\) and \(z^{(0)} = i\). Here are the numerical results,

Table 4.1: Newton’s method for complex variable

<table>
<thead>
<tr>
<th>(l)</th>
<th>(z^* = -\frac{1}{2} + i\frac{\sqrt{3}}{2})</th>
<th>(\lambda_{Quadratic})</th>
<th>(z^* = 1)</th>
<th>(\lambda_{Quadratic})</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>(i)</td>
<td>0.969771</td>
<td>3</td>
<td>0.259259</td>
</tr>
<tr>
<td>1</td>
<td>(-0.333333 + 0.666667i)</td>
<td>1.49378</td>
<td>1.0124</td>
<td>0.860947</td>
</tr>
<tr>
<td>2</td>
<td>(-0.582222 + 0.924444i)</td>
<td>0.889345</td>
<td>1.12002</td>
<td>0.624611</td>
</tr>
<tr>
<td>3</td>
<td>(-0.508791 + 0.868166i)</td>
<td>0.9917</td>
<td>1.0124</td>
<td>0.860947</td>
</tr>
<tr>
<td>4</td>
<td>(-0.5 + 0.866025i)</td>
<td>0.500099</td>
<td>1.00015</td>
<td>0.983716</td>
</tr>
<tr>
<td>5</td>
<td>(-0.5 + 0.866025i)</td>
<td>2.55643</td>
<td>1.00015</td>
<td>0.983716</td>
</tr>
<tr>
<td>6</td>
<td>(-0.5 + 0.866025i)</td>
<td>1.00015</td>
<td>0.999798</td>
<td>0.847538</td>
</tr>
<tr>
<td>7</td>
<td>(-0.5 + 0.866025i)</td>
<td>1.00015</td>
<td>0.999798</td>
<td>0.847538</td>
</tr>
</tbody>
</table>

\(l = \) iteration number, \(z^* = \) zero of the function and \(\lambda = \) approximation to the asymptotic error constant.

Now, let’s make use of the system of nonlinear equations. In this case the same initial approximation translates to \(x^{(0)} = 3\) and \(y^{(0)} = 0\) for the solution \((1, 0)\), which is \(z^* = 1\). For the zero \(z^* = -\frac{1}{2} + i\frac{\sqrt{3}}{2}\), we expect to get the solution \((-\frac{1}{2}, \frac{\sqrt{3}}{2})\) and the initial approximation is \(x^{(0)} = 0\) and \(y^{(0)} = 1\). We present the numerical results here.
Table 4.2: Newton’s method

<table>
<thead>
<tr>
<th>I</th>
<th>$x^* = -\frac{1}{2}$</th>
<th>$y^* = \sqrt{3}$</th>
<th>$\lambda_{\text{Quadratic}}$</th>
<th>$x^* = 1$</th>
<th>$y^* = 0$</th>
<th>$\lambda_{\text{Quadratic}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>-0.333333</td>
<td>0.666667</td>
<td>0.969771</td>
<td>2.03704</td>
<td>0</td>
<td>0.259259</td>
</tr>
<tr>
<td>2</td>
<td>-0.582222</td>
<td>0.924444</td>
<td>1.49378</td>
<td>1.0124</td>
<td>0</td>
<td>0.407603</td>
</tr>
<tr>
<td>3</td>
<td>-0.508791</td>
<td>0.898810</td>
<td>1.00225</td>
<td>1.00015</td>
<td>0</td>
<td>0.624611</td>
</tr>
<tr>
<td>4</td>
<td>-0.500069</td>
<td>0.865982</td>
<td>0.9917</td>
<td>0.999798</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-0.500069</td>
<td>0.865982</td>
<td>0.9917</td>
<td>0.999798</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>-0.500069</td>
<td>0.865982</td>
<td>0.9917</td>
<td>0.999798</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>-0.500069</td>
<td>0.865982</td>
<td>0.9917</td>
<td>0.999798</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>-0.500069</td>
<td>0.865982</td>
<td>0.9917</td>
<td>0.999798</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$I =$ iteration number, $x^*$ and $y^*$ are the solutions of the system and $\lambda =$ approximation to the asymptotic error constant.

From the above two tables we see that these two techniques produce exactly the same approximations at each iteration. But using the iteration function involving the complex variable $z$ provides us a simple way of calculating the approximations. Moreover, it provides a problem in one variable instead of a system of nonlinear equations involving two variables. We have analyzed the roots of $z^4 - 1 = 0$ and $z^5 - 1 = 0$ and found that the approximations at each iteration are exactly same.

Now, we will try to establish symbolically if these two techniques are supposed to give us the same approximations theoretically. We will analyze the same problem for this purpose. The iteration function of Newton’s method given by Equation 4.2 for cubic root of $z$ can be given by,

$$g(z) = z - \frac{z^3 - 1}{3z^2}$$

$$\implies g(z) = \frac{3z^3 - z^3 + 1}{3z^2}$$

$$\implies g(z) = \frac{2z^3 + 1}{3z^2}.$$

Expressing $z = x + iy$ and $g(z) = u + iv$, we get,

$$u + iv = \frac{2(x+iy)^3 + 1}{3(x+iy)^2}$$

$$\implies u + iv = \frac{(2x^3 - 6xy^2 + 1) + i(6x^2y - 2y^3)}{(3x^2 - 3y^2)^2 + i6xy}$$

$$\implies u + iv = \frac{(2x^3 - 6xy^2 + 1)(3x^2 - 3y^2) + 6xy(6x^2y - 2y^3)}{(3x^2 - 3y^2)^2 + (6xy)^2} + i\left(\frac{6x^2y - 2y^3 - 6xy(2x^3 - 6xy^2 + 1)}{(3x^2 - 3y^2)^2 + (6xy)^2}\right).$$
\[
\Longrightarrow u + iv = \frac{2x^5 + 4x^3y^2 + x^2 + 2xy^4 - y^2}{3(x^2 + y^2)^2} + i\frac{2x^4y + 4x^2y^3 - 2xy + 2y^5}{3(x^2 + y^2)^2}.
\]

(4.4)

Clearly \((u, v) = \left( \frac{2x^5 + 4x^3y^2 + x^2 + 2xy^4 - y^2}{3(x^2 + y^2)^2}, \frac{2x^4y + 4x^2y^3 - 2xy + 2y^5}{3(x^2 + y^2)^2} \right)\) is the approximation Newton’s method produces starting with any \((x, y)\).

Now, we will check if Newton’s method for the system 4.3 produces same result.

We can write the iteration scheme as follows,

\[
\begin{pmatrix}
  u \\
  v
\end{pmatrix}
= \begin{pmatrix}
  x \\
  y
\end{pmatrix}
- \begin{pmatrix}
  3x^2 - 3y^2 & -6xy \\
  6xy & 3x^2 - 3y^2
\end{pmatrix}^{-1}
\begin{pmatrix}
  x^3 - 3xy^2 - 1 \\
  3x^2y - y^3
\end{pmatrix}
\]

\[
\begin{pmatrix}
  u \\
  v
\end{pmatrix}
= \begin{pmatrix}
  x \\
  y
\end{pmatrix}
- \begin{pmatrix}
  \frac{3x^2 - 3y^2}{9x^4 + 18x^2y^2 + 9y^4} & \frac{6xy}{9x^4 + 18x^2y^2 + 9y^4} \\
  \frac{6xy}{9x^4 + 18x^2y^2 + 9y^4} & \frac{3x^2 - 3y^2}{9x^4 + 18x^2y^2 + 9y^4}
\end{pmatrix}
\begin{pmatrix}
  x^3 - 3xy^2 - 1 \\
  3x^2y - y^3
\end{pmatrix}
\]

\[
\begin{pmatrix}
  u \\
  v
\end{pmatrix}
= \begin{pmatrix}
  x \\
  y
\end{pmatrix}
- \begin{pmatrix}
  \frac{6xy(3x^2y - y^3)}{9x^4 + 18x^2y^2 + 9y^4} + \frac{(3x^2 - 3y^2)(x^3 - 3xy^2 - 1)}{9x^4 + 18x^2y^2 + 9y^4} \\
  \frac{(3x^2 - 3y^2)(3x^2y - y^3)}{9x^4 + 18x^2y^2 + 9y^4} - \frac{6xy(3x^2y - y^3)}{9x^4 + 18x^2y^2 + 9y^4}
\end{pmatrix}
\begin{pmatrix}
  x^3 - 3xy^2 - 1 \\
  3x^2y - y^3
\end{pmatrix}
\]

\[
\begin{pmatrix}
  u \\
  v
\end{pmatrix}
= \begin{pmatrix}
  \frac{2x^5 + 4x^3y^2 + x^2 + 2xy^4 - y^2}{3(x^2 + y^2)^2} \\
  \frac{2x^4y + 4x^2y^3 - 2xy + 2y^5}{3(x^2 + y^2)^2}
\end{pmatrix}. \quad (4.5)
\]

So, Equations 4.4 and 4.5 tell us that we end up with the exact same approximations \((u, v)\). Similarly, it can be shown for any \(n\) these two methods produce exactly the same approximations for the equation \(z^n - 1 = 0\), which is also supported by our numerical results.

Let’s analyze the region of attraction for the same problem. We will use the system of Equations 4.3 to see where the Jacobian is singular. The Jacobian of the system
can be given by,
\[
\begin{pmatrix}
3x^2 - 3y^2 & -6xy \\
6xy & 3x^2 - 3y^2
\end{pmatrix}.
\]

The determinant of the Jacobian can be given by,
\[
(3x^2 - 3y^2)^2 + 36x^2y^2.
\]

To find where the Jacobian is singular, we set the determinant equal to zero,
\[
(3x^2 - 3y^2)^2 + 36x^2y^2 = 0
\implies (3x^2 + 3y^2)^2 = 0
\implies 3x^2 + 3y^2 = 0.
\]

Since \(x\) and \(y\) are both real numbers, this tells us that \(x = 0\) and \(y = 0\). The Jacobian is nonsingular in the entire \(xy\) plane except at \((0, 0)\). In fact, the iteration function 4.2 has a pole at \(z = 0\). So, Newton’s method will fail if this point is chosen as the initial approximation. It might seem that all other initial approximations should give us a convergent method. However, this is not the case. All the points at which the Newton’s iteration function is zero (let’s call these pre-images of zero) will give us trouble as well because if we start with one of these pre-images, our next approximation by Newton’s method will be zero and Newton’s method will fail. Similarly, the pre-images of the pre-images of zeros will also have the same problem because it will take two iterations to obtain zero as an approximation. This goes on recursively. So, we would want to avoid all these pre-images of zero, pre-images of pre-images of zero and so on. Let’s analyze this phenomenon for this particular function \(f(z) = z^3 - 1\). The iteration function can be given as,
\[
z^{(k)} = z^{(k-1)} - \frac{(z^{(k-1)})^3 - 1}{3(z^{(k-1)})^2}
\implies z^{(k)} = \frac{2(z^{(k-1)})^3 + 1}{3(z^{(k-1)})^2}.
\]

Avoiding the superscripts, we can write the Newton’s iteration function generally as,
\[ g(z) = \frac{2z^3+1}{3z^2}. \]

Let’s call the list of pre-images “\(z_p\)”. At the first step, we solve \(g(z) = \frac{2z^3+1}{3z^2} = 0\) for \(z\) to get three solutions. So, our list of pre-images for the moment is \(z_p = \{-0.793701, 0.39685 - 0.687365i, 0.39685 + 0.687365i\}\). We will call these pre-images as level one pre-images. In the next step, we will set \(g(z) = -0.793701\) to get three more pre-images. We will do the same for each of the pre-images of level one and solve for \(z\). Clearly, we will get nine pre-images. We label these as level two pre-images. So, appending these nine newfound pre-images, our updated list of pre-images become \(z_p = \{-0.793701, 0.39685 - 0.687365i, 0.39685 + 0.687365i, -1.43378, 0.121612 - 0.577875i, 0.121612 + 0.577875i, -0.561261 - 0.183618i, 0.439649 + 0.394257i, 0.716888 - 1.24169i, -0.561261 + 0.183618i, 0.439649 - 0.394257i, 0.716888 + 1.24169i\}\). Then, we follow the same procedure to obtain the pre-images of these nine and label them as level three pre-images. We continue in this manner until we obtain enough pre-images to see a pattern. For this specific analysis, we have used Mathematica to find pre-images up to level nine and plotted them to see the distribution in the complex plane. The following Figures 4.1 to 4.9 show the back propagation level by level.

As we can see that for the first few levels, it is hard to find a pattern. But, as we plot more and more levels, we see a beautiful fractal behavior showing up. The images are not complete, because we used pre-images until level nine. We can assume that the images will be sharper as we include more levels. If we could plot all the pre-images,
Figure 4.3: All pre-images up to level three

Figure 4.4: All pre-images up to level four

Figure 4.5: All pre-images up to level five

Figure 4.6: All pre-images up to level six

Figure 4.7: All pre-images up to level seven

Figure 4.8: All pre-images up to level eight
we would have seen a perfect fractal image indicating all the points we should avoid choosing as initial approximations for Newton’s method to get convergence.

We know the cube roots of unity are $1, \ -\frac{1}{2} + \frac{\sqrt{3}}{2}i$ and $-\frac{1}{2} - \frac{\sqrt{3}}{2}i$. The first root lies on the positive side of $x$ axis. The second and third roots lie on the second and third quadrant making an angle of $\frac{2\pi}{3}$ and $-\frac{2\pi}{3}$ radians with the positive $x$ axis. Looking at Figure 4.9, we can see that the neighboring points around each of the roots are free from any trouble. In fact, all the pre-images seem to lie nearby the line found by rotating the line containing any of the roots at angle of $\pi$ radians. Choosing an initial approximation near any of the roots, in other words choosing an initial approximation avoiding the trouble area should give us a convergent Newton’s method. Another question worth asking is which root the sequence of approximations should converge to. Usually it is a very hard question to answer. But, in case of $n$-th root of unity, the basin of attraction follows an interesting pattern, which we can guess by looking at the pattern we saw for pre-images. Here we present the following picture generated by Mathematica code from [8] indicating the region of attraction for each of the roots of the equation $z^3 - 1 = 0$:

Each color represents the region of attraction for one of the roots. Just like
Figure 4.10: Fractal image showing basin of attraction for each root of $z^3 - 1 = 0$ in different shade

Figure 4.9 the basin of attraction shows fractal behavior near the regions where the pre-images lie. This is understandable because we can expect the complement of a fractal graph to be fractal.

One interesting observation is that the fractal behavior happens near or around a line (we can imagine three axes through origin around/near which all the fractal phenomena are happening and also the pre-images are distributed near and around these axes), which bisects the angle between two consecutive lines that contain roots. Let’s name these fictitious axes as axes of pre-images. This phenomenon holds for all values of $n \geq 2$. In addition to that it seems from Figure 4.9 and 4.10 that the axis of pre-images can be found by rotating the root-containing line by $\pi$ radians. In fact for all odd $n$ this seems to hold. But, for even $n$, the lines opposite of a line that
contains root, also contains another root. However, the axes of pre-images always bisect an angle between two consecutive lines that contain roots. We know that if we draw a line from the origin to each root of $z^n - 1 = 0$, the angle between each two consecutive lines will be $\frac{2\pi}{n}$. We present two more images of basin of attraction of the roots of the equation $z^n - 1 = 0$ for $n = 4$ and $n = 5$ generated by Mathematica code from [8].

Similarly, the angle between two consecutive axes of pre-images is also $\frac{2\pi}{n}$.

Before wrapping up our discussion, we are going to look into one more interesting phenomenon observed when Newton’s method is applied to $n$th roots of unity. If an initial approximation is chosen on the line through origin that contains a root or on the line that divides the angle between two consecutive roots (the angle created by the lines that join the origin and the roots), then Newton’s iteration function will generate a sequence of approximations, which stay on the same line until they converge if there is a root on the line or continue to stay on the line if there is no root on the line. This can be shown analytically. The iteration function of Newton’s method can be given by,

$$g(z) = z - \frac{z^n - 1}{nz^{n-1}}$$
\[ g(z) = \frac{(n-1)z^n + 1}{nz^{n-1}}. \]

Now, consider \( z = x + iy \) is our initial approximation, where \( x \) and \( y \) are real numbers. This can be written as \( z = r\cos \theta + ir\sin \theta = re^{i\theta} \). So, the iteration function becomes,

\[
\frac{(n-1)z^n + 1}{nz^n} \times z = \frac{(n-1)(re^{i\theta})^n + 1}{n(re^{i\theta})^n} \times z = \frac{(n-1)r^n(e^{i\theta})^n + 1}{n r^n(e^{i\theta})^n} \times z = \frac{(n-1)r^n e^{in\theta} + 1}{n r^n e^{i\theta}} \times z.
\]

Now, any point on the lines we have mentioned can be expressed as \( re^{ic\pi} \), where \( c = 1, 2, 3, \ldots, 2n \). This implies the angle \( \theta = \frac{c\pi}{n} \). Putting this in the iteration function, we get,

\[
\frac{(n-1)r^n e^{in\frac{c\pi}{n}} + 1}{n r^n e^{i\frac{c\pi}{n}}} \times z = \frac{(n-1)r^n e^{ic\pi} + 1}{n r^n e^{i\pi}} \times z.
\]

For odd \( c \), \( e^{ic\pi} = -1 \) and for even \( c \), \( e^{ic\pi} = 1 \). So, for odd \( c \), the iteration function becomes \( \frac{(n-1)r^n(-1)^{n+1} + 1}{n r^n(-1)} \times z \). For even \( c \), we get \( \frac{(n-1)r^n + 1}{n r^n} \times z \). Let us now express both as \( qz \), where \( q = \frac{(n-1)r^n(-1)^{n+1} + 1}{n r^n(-1)} \), which is clearly a real number. This implies Newton’s iteration function generates a sequence of numbers which are some constant multiples of previous approximation \( z \). In words, the iterates do not move away from any of the mentioned lines.
Chapter 5

Conclusions

The most important aspects of a numerical method is the ability to converge and the speed of convergence even if it is under some conditions. We have seen that the Bisection method is a good technique to have guaranteed convergence, when there is a sign change around a zero. We can also get a bound on the number of iterations required for a certain accuracy. Newton’s method, the Secant method and the Chord method are based on a similar idea with a little variation. While Newton’s method needs a calculation of a derivative at each step, the Chord method only needs one evaluation of a derivative and the Secant method does not even need calculations of derivatives. Though computationally we benefit from the Secant and the Chord methods, we get better convergence with Newton’s method. However, these two methods can be useful for problems, where existence of derivatives at some of the iterates (chord method requires existence of derivative only at the initial approximation) are not guaranteed.

We have shown that the convergence of Newton’s method is highly dependent on the initial approximation. In the $y = \tan^{-1}(x)$ problem we saw that for some bad initial approximations the method may even fail to converge, while for a good starting point it is supposed to converge cubically ($f''(x^*) = 0$). Usually for most of the problems, Newton’s method converges for sufficiently close initial approximations. We have shown how to find the region of attraction for simple zeros to have guaranteed convergence.

When Newton’s method converges, it does so quadratically for simple zeros and linearly for zeros with higher multiplicity i.e. when we have derivative being zero at
a solution. From [1] we have seen that for some specific conditions on the iteration function \( g(x) \) and its derivatives, Newton’s method can show higher order convergence. We have shown that similar conditions can be posed on \( f(x) \) and its derivatives as well, for which Newton’s method will converge with better than quadratic order. We have also been able to present a simple formula to calculate the asymptotic error constant analytically, which involves derivatives of \( f \) and not \( g \) (iteration function).

We have explored three techniques to accelerate convergence of Newton’s method. The multistep Newton’s method is applicable to only simple zeros, and can give a boost of at least \( \alpha - 1 \) orders, where \( \alpha \) is the order of convergence for basic Newton’s method (verified analytically until \( \alpha = 8 \)). The composition of Newton’s method to itself gives \( \alpha^2 \) order of convergence, which is also applicable for zeros of multiplicity one and higher. For a zero of higher multiplicity, the asymptotic error constant is \( \lambda^2 \), where \( \lambda \) is the asymptotic error constant of the basic Newton’s method. If the multiplicity \( m \) of a zero of a function \( f(x) = (x - p)^m h(x) \) is known beforehand, the third modification we presented improves a linear convergence to at least quadratic order with an asymptotic error constant \( \frac{h'(p)}{mh(p)} \), where \( f(p) = 0 \) but \( h(p) \neq 0 \).

We have also used Newton’s method to find a local minimizer of a function. Though Newton’s method is not meant to be suitable for minimization [5], we showed that if for some interval \( f''(x) > 0 \) for all \( x \) in that interval except possibly at the minimizer, Newton’s method converges to the minimizer. The higher dimensional analog of \( f''(x) \) is the Hessian matrix of \( f(x_1, x_2, ..., x_n) \). So, we were able to find regions around the minimizer where the Hessian matrix is positive definite using the property of positive definiteness of a matrix.

We have established certain analogies between Newton’s method in the single variable and the multivariable case. We derived Newton’s method for single variables from the first order Taylor polynomial. We have verified symbolically that the approximation we are supposed to get using the first order Taylor polynomial is
exactly the approximation we get from Newton’s method for two variables. The higher dimensional analog of a first derivative being zero is the singular Jacobian. We have seen that the order of convergence for a solution, where the Jacobian is singular, is linear. For nonsingular Jacobian we get quadratic convergence.

In the issue of finding a region of attraction, we employed the analogous technique as the single variable one. Further, we have seen that Ostrowski’s theorem give us a sufficient condition for a solution to be a point of attraction using the spectral radius of the Jacobian of the Newton’s iteration function. We have also verified numerically that systems of equations in two and three variables show the analogous phenomena. Acceleration techniques seemed to behave similarly in one and higher dimensional cases.

Finally, we analyzed Newton’s method for complex valued functions. We have shown analytically that solving \( f(z) = 0 \) for complex \( z \) is exactly the same of solving the system \( \vec{F}(x, y) = \vec{0} \), where the real and imaginary part of \( f(z) \) form a system of two equations \( f_1(x, y) = 0 \) and \( f_2(x, y) = 0 \) in real variables \( x \) and \( y \). Solving \( f(z) = 0 \) is more desirable and computationally more efficient. We have analyzed where in the complex plane we can have bad initial approximations for the famous \( n \)-th root of unity problems. We have seen that these points are distributed in the complex plane in a fractal manner. The graphs were amazingly beautiful, because of the symmetry involved. However, we assume any function which has a complex rational iteration function should have fractal bad zones even though they might not be symmetric. Based on the fractal bad zone, we expected the good zones for convergence to show fractal behavior as well. We have seen numerically that is exactly what we get. One final interesting behavior we observed is that if we choose an initial approximation on any of the axes of preimages of the origin or a line that contains a root of the function \( z^n - 1 = 0 \), the iterations do not move away from the line. We have shown this analytically for any \( n \).
Bibliography


Appendix A

Mathematica Codes

**Code Mathematica A.1: Basic Single Variable Newton's Method**

```mathematica
f[x_] := ArcTan[x];
actualSoln = 0; currentX = 1.39174520027; tol = 10^{-6}; i = 1; n = 500;
iList = {"I", 0}; approximationList = {"Approximation", currentX};
LinList = {"Lin", " "}; QuadList = {"Quad", " "}; CubeList = {"Cube", " "};

While [i < n,
fprimeCurrentX = f'[currentX];
If[Abs[fprimeCurrentX] > 0, newX = N[currentX - f[currentX]/fprimeCurrentX], Print["Method Failed"]];
d = Abs[newX - currentX];
If[Abs[currentX - actualSoln] > 0,
Lin = Abs[newX - actualSoln]/(Abs[currentX - actualSoln]);
Quad = Abs[newX - actualSoln]/(Abs[currentX - actualSoln])^2;
Cube = Abs[newX - actualSoln]/(Abs[currentX - actualSoln])^3;
AppendTo[iList, i];
AppendTo[approximationList, newX];
AppendTo[LinList, Lin];
AppendTo[QuadList, Quad];
AppendTo[CubeList, Cube];
];
AppendTo[iList, i]; AppendTo[approximationList, newX];
AppendTo[LinList, " "]; AppendTo[QuadList, " "]; AppendTo[CubeList, " "]; If[d < tol, Break[]];
currentX = newX; i++;
result = Transpose[{iList, approximationList, LinList, QuadList, CubeList}]; TableForm[%]
```

(* Comments:
For acceleration of Single Variable Newton’s Method we only need to change the definition of newX in line 7
of the above codeblock as followings:
1. newX = N[currentX - (f[currentX] + f[currentX - f[currentX]/fprimeCurrentX])/fprimeCurrentX]
2. newX = N[currentX - f[currentX]/fprimeCurrentX - f[currentX - f[currentX]/fprimeCurrentX]/fprimeCurrentX]
3. newX = N[currentX - m f[currentX]/fprimeCurrentX], where m is the multiplicity of zero *)

73
### Code Mathematica A.2: Basic Newton’s Method for multivariables

```mathematica
Clear[f1, f2, x1, x2, J, newx1, newx2, x1List, x2List, iList, LinList, QuadList, CubeList, result]
f1[x1_, x2_] := x_1^2 + x_2^2 - 9;
f2[x1_, x2_] := -x_1 + x_2^2 - 3;
jacobianOfF = Outer[D, {f1[x1, x2], f2[x1, x2]}, {x1, x2}];
inverseJacobian = Inverse[jacobianOfF];
X = {x1, x2};
F = {f1[x1, x2], f2[x1, x2]};
changeInVectorX = LinearSolve[jacobianOfF, {-F[[1]], -F[[2]]}];
x1 = 4; x2 = 3; actualSolnX1 = 2; actualSolnX2 = Sqrt[5];
n = 100; Tol = 10^-6; i = 1;
actualSolnX = {actualSolnX1, actualSolnX2};
oldDistance = Norm[actualSolnX - X], 2];
x1List = {"X_1", x1}; x2List = {"X_2", x2}; iList = {"I", 0};
LinList = {"\lambda Lin", " "};
QuadList = {"\lambda Quad", " "};
CubeList = {"\lambda Cube", " "};
While[i < n,
If[oldDistance > 0,
newx1 = x1 + N[changeInVectorX[[1]]];
newx2 = x2 + N[changeInVectorX[[2]]];
newX = {newx1, newx2};
AppendTo[iList, i];
AppendTo[x1List, newx1];
AppendTo[x2List, newx2];
newDistance = Norm[(actualSolnX - newX), 2];
\lambda Lin = newDistance/oldDistance;
\lambda Quad = newDistance/oldDistance^2;
\lambda Cube = newDistance/oldDistance^3;
AppendTo[LinList, \lambda Lin];
AppendTo[QuadList, \lambda Quad];
AppendTo[CubeList, \lambda Cube];
] x1 = newx1; x2 = newx2; oldDistance = newDistance,
AppendTo[LinList, " "]; AppendTo[QuadList, " "]; AppendTo[CubeList, " "]; Break[];
If[Norm[changeInVectorX, 2] < Tol, Break[]];
i++;
];
result = Transpose[{iList, x1List, x2List, LinList, QuadList, CubeList}] ;
TableForm[%, TableHeadings -> {None, {"I", "X_1", "X_2", \lambda Lin, \lambda Quad, \lambda Cube"}}];
```

74
### Code Mathematica A.3: Multistep Newton's Method

```mathematica
Clear[f1, f2, x1, x2, J, newx1, newx2, x1List, x2List, iList, LinλList, QuadλList, CubeλList, result]
f1[x1_, x2_] := x1^2 + x2^2 - 9;
f2[x1_, x2_] := -x1 + x2^2 - 3;

jacobianOff = Outer[D, {f1[x1, x2], f2[x1, x2]}, {x1, x2}];
inverseJacobian = Inverse[jacobianOff];

X = {x1, x2}; F = {f1[x1, x2], f2[x1, x2]};

intermediateX = X - inverseJacobian.F;

intermediateX1 = intermediateX[[1]]; intermediateX2 = intermediateX[[2]]; 
intermediateF = {f1[intermediateX1, intermediateX2], f2[intermediateX1, intermediateX2]};

finalF = Simplify[F + intermediateF];

changeInVectorX = LinearSolve[jacobianOff, {-finalF[[1]], -finalF[[2]]}];

x1 = 4; x2 = 3; actualSolnX1 = 2; actualSolnX2 = Sqrt[5];
n = 100; Tol = 10^(-6); i = 1;

actualSolnX = {actualSolnX1, actualSolnX2};

oldDistance = Norm[(actualSolnX - X), 2];
x1List = {"X1"}, x1; x2List = {"X2"}, x2; iList = {"I", 0};
LinλList = {"λLin", ""}; QuadλList = {"λQuad", ""};
CubeλList = {"λCube", ""}; QuartλList = {"λQuart", ""};

While[i < n, 
If[oldDistance > 0, 
newx1 = x1 + N[changeInVectorX[[1]]];
newx2 = x2 + N[changeInVectorX[[2]]];
newX = {newx1, newx2};
Appending[iList, i]; AppendTo[x1List, newx1]; AppendTo[x2List, newx2];
newDistance = Norm[(actualSolnX - newX), 2];
λLin = newDistance/oldDistance;
λQuad = newDistance/oldDistance^2;
λCube = newDistance/oldDistance^3;
λQuart = newDistance/oldDistance^4;
Appending[LinλList, λLin]; AppendTo[QuadλList, λQuad];
Appending[CubeλList, λCube]; AppendTo[QuartλList, λQuart];
]
]
If[Norm[changeInVectorX, 2] < Tol, Break[]];
i++; ];

result = Transpose[{iList, x1List, x2List, LinλList, QuadλList, CubeλList, QuartλList}];
TableForm[%]
```

75
Clear[ f1, f2, F, x1, x2, X, newx1, newx2, newX, result]

f1[x1_, x2_] := x1^2 + x2^2 - 9;
f2[x1_, x2_] := -x1 + x2^2 - 3;
F = {f1[x1, x2], f2[x1, x2]}; X = {x1, x2};
jacobianOfF = Outer[D, {f1[x1, x2], f2[x1, x2]}, X];
jacobianFunctionForm[x1_, x2_, ] := jacobianOfF /. Thread[X -> {x1, x2}];
inverseJacobian = Inverse[jacobianOfF];
intermediateX = Simplify[X - inverseJacobian.F];
intermediateX1 = intermediateX[[1]]; intermediateX2 = intermediateX[[2]];
intermediateF = {f1[intermediateX1, intermediateX2], f2[intermediateX1, intermediateX2]};
intermediateJacobianOfF = jacobianFunctionForm[intermediateX1, intermediateX2];
changeInVectorX = Simplify[intermediateJacobianOfF.intermediateF];
newX = newX + N[changeInVectorX];
actualSolnX = Sqrt[5]; n = 100; Tol = 10^-6; i = 1;
actualSolnX = {actualSolnX1, actualSolnX2};
oldDistance = Norm[(actualSolnX - X), 2]; newDistance = Norm[(actualSolnX - newX), 2];
λLin = newDistance/oldDistance; λQuad = newDistance/oldDistance^2;
λCube = newDistance/oldDistance^3; λQuart = newDistance/oldDistance^4;
λQuint = newDistance/oldDistance^5;
newX = newX + N[changeInVectorX];
While[ i < n, If[Norm[(X - actualSolnX), 2] > 0, 
newx1 = x1 + N[changeInVectorX[[1]]];
newx2 = x2 + N[changeInVectorX[[2]]];
newX = {newx1, newx2};
AppendToList, i]; AppendToList[x1List, newX1]; AppendToList[x2List, newX2];
oldDistance = Norm[(actualSolnX - X), 2]; newDistance = Norm[(actualSolnX - newX), 2];
λLin = newDistance/oldDistance; λQuad = newDistance/oldDistance^2;
λCube = newDistance/oldDistance^3; λQuart = newDistance/oldDistance^4;
λQuint = newDistance/oldDistance^5;
AppendToList[λLinList, λLin]; AppendToList[λQuadList, λQuad];
AppendToList[λCubeList, λCube]; AppendToList[λQuartList, λQuart];
AppendToList[λQuintList, λQuart];
x1 = newX1;x2 = newX2;
AppendToList[x1List, x2List, LinList, QuadList, CubeList, QuartList, QuintList];
Break[]];
If[Norm[changeInVectorX, 2] < Tol, Break[]];
i++;];
result = Transpose[{iList, x1List, x2List, LinList, QuadList, CubeList, QuartList, QuintList}];
TableForm[%]
Code Mathematica A.5: Preimages of the zeros of Newton’s iteration function

\begin{verbatim}
f[z_] := z^3 - 1;
g[z_] := Simplify[z - f[z]/f'[z]];
RHS = {0.};
i = 1;
levels = 6;
n = (3^levels)/2;
Do[
solns = z /. N[Solve[g[z] == RHS[[i]], z]];
RHS = DeleteDuplicates[Join[RHS, solns]];
i++;
, {n}]
ListPlot[ReIm[RHS], PlotStyle -> {Black}]
\end{verbatim}