# UNCONSTRAINED OPTIMIZATION AND NONLINEAR SYSTEMS OF EQUATIONS REVIEW AND ANALYSIS USING EXCEL 

# A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF <br> NEAR EAST UNIVERSITY 

by
ISA ABDULLAHI BABA

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN MATHEMATICS

NICOSIA
2014

# UNCONSTRAINED OPTIMIZATION AND NONLINEAR SYSTEMS OF EQUATIONS REVIEW AND ANALYSIS USING EXCEL 

# A THESIS SUBMITTED TO THE GRADUATE SCHOOL OF APPLIED SCIENCES OF <br> NEAR EAST UNIVERSITY 

by
ISA ABDULLAHI BABA

## In Partial Fulfilment of the Requirements for the Degree of Master of Science in Mathematics

# Isa Abdullahi Baba: Unconstrained Optimization and Nonlinear Systems of Equations Review and Analysis Using Excel 

Approval of Director of Graduate School of Applied Sciences

Prof. Dr. İlkay SALİHOĞLU

We certify this thesis is satisfactory for the award of the degree of Masters of Science in Mathematics

## Examining Committee in Charge:

Dr. A.M Othman,
Committee Chairman And Supervisor Department of Mathematics, Near East University

Prof. Dr. Adigozal Dosiyev,

Assoc. Prof. Dr. Evren Hınçal,

Department of Mathematics, Eastern
Mediterranean University

Department of Mathematics, Near East University

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Isa Abdullahi Baba
Signature:
Date:

To the memory of my late Father Alhaji Abdullahi Baba may your gentle soul rest in perfect peace Ameen.

## ACKNOWLEDGEMENTS

All praise is for Allah, lord of all that exists. Oh Allah, send prayers and salutations upon our beloved prophet Muhammad, his family, his companions and all those who follow his path until the last day.

First and foremost my profound gratitude goes to my loving, caring, and wonderful parents Alh. Abdullahi Baba and Haj. Rabi Isa for their love, care, understanding, and Prayers. In addition I wish to thank my step mother Haj.Aisha Muhammad for her contributions towards my achievement.

I take this opportunity to express my profound gratitude and deep regards to my supervisor Dr. Abdulrahman Mousa Othman for his exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by him time to time shall carry me a long way in the journey of life on which I am about to embark.

I also take this opportunity to express a deep sense of gratitude to the entire members of Mathematics department. My humble lecturers, Prof. Kaya I. Ozkin, Assoc. Prof. Evren Hinkel, and Asst. Prof. Burak Sekereglu a big thanks for your cordial support, valuable information and guidance, which helped me in completing this task through various stages.

I am obliged to thank my able leader, my mentor, the architect of modern Kano, His Excellency Engr. Dr. Rabiu Musa Kwankwaso. His selfish less government made our dreams a reality. May Allah (S.W.A) reward him abundantly.

Lastly, I thank my siblings Hadiza, Binta, Amina, Safiya, Harun, Aminu, Hassan, Bashir, Sadiya, and Sani for their constant encouragement without which this
assignment would not be possible. To the entire 501 Kano state master's scholarshipStudents I say it was wonderful being one of you. I want at this juncture say a big thanks to my friends, Salim J.D, Salim I.M, Shamsu A, Sagir M, Umar A, Musa S.A, Adamu D, Fa’izu H.A, Samir B, Mustapha N.S, Anas M, Aminu Y.N, Shehu M, Mustapha D and Mahmud A. Thank you all for moral, financial and spiritual support. May Allah bless you all and replenish your purse.


#### Abstract

This thesis gives a review of the methods for solving systems of nonlinear equations and unconstrained minimization of real-valued functions. An analysis of the pertinent mathematical theories andminimization methods are presented and tested using a wellknown setof benchmarkproblems. Methods forsolving systems of nonlinear algebraic equations include; Newton's method, Quasi-Newton's method and Diagonal Broydenlike and Homotopy and Continuation method.

For unconstrainedminimization it covers; the Steepest Descent method, the FletcherReeves and Polak-Ribière Conjugate Gradient methods, the Modified Newton's method and the Quasi-Newton BFGS and DFP method using Analytic line search method to calculate the step length.

Traditionally researchers use one of two computationaltools when seeking approximations to their numerical analysis and optimization problems. They either use readily available software packages or write their own tailor made programs using some high-level programming languages. Both of these are capable of handling fairly complicated and large problems effectively. The disadvantages of both methods are highlighted in the introduction chapter. In this thesis we used the EXCEL spread sheet to carry out the calculations because in our opinion, from teaching view point, it strikes a balance between the other two methods. The capabilities and the limitations of Excel as a computational tool is also studied and presented.


Keywords: Optimization, Convergence, Minimization, Excel, Maximization

## ÖZET

Bu tez, lineerolmayan denklemlerin ve reel değerli fonksiyonların kısıtlamasız minimize sistemlerini çözmek için kullanılan yöntemleri içerir. İlgili matematiksel teoriler ve minimizasyon yöntemlerinin analizileri Benchmark problemleri kullanılarak test edilidi. Lineer olmayan denklem sistemlerinin çözüm yöntemleri şunlardır: Newton, Quasi-Newton, Diagonal Broyden ve Homotopy ve SüreklilikYöntemi.

Serbest - Sinırsız- Kısıtlı (Unconstrained) minimizasyon yöntemi ise şunları kapsamaktadır: Dik İniş Yöntemi, Fletcher-Reeves ve Polak-Ribiere Eşlenik Gradyan Yöntemi, Değiştirilmiş Newton Yöntemi, Quasi-Newton BFGS ve DFP Yöntemi. Bununla birlikte adım sayısını hesaplamak için de analitik hat arama yöntemi kullanılmıştır.

Sayısal analiz ve optimizasyon problemlerine yaklaşımları ararken geleneksel olarak araştırmacılar iki hesaplama araçlarından birini kullanırlar. Araştırmacılar hazır yazılım paketlerini veya bazı üst düzey programlama dillerini kullanarak, yeni programlar yazarlar. Bunların her ikisi de etkili ancak karmaşık olduklarından, söz konusu yöntemlerin kullanılmasıbüyük sorunları da beraberinde getirmektedir. Her iki yöntemin dezavantajları "Giriş Bölümü"nde vurgulanmıştır. Bu tezde, EXCEL kullanılmanın yukarıda belirtilen iki program arasında bir denge oluşturduğu sonucuna varılmıştır.

AnahtarSözcükler: Optimizasyon, Denklem, Minimizasyon, Excel, Maximizasyon

## TABLE OF CONTENTS

ACKNOWLEDGMENTS ..... ii
ABSTRACT ..... iv
ÖZET ..... v
TABLE OF CONTENTS ..... vi
LIST OF TABLES ..... viii
LIST OF ABBREVIATIONS ..... ix
CHAPTER 1: INTRODUCTION
1.1 About Optimization ..... 1
1.2 About this Thesis ..... 1
CHAPTER 2: BACKGROUND AND LITERATURE REVIEW
2.1 Systems of Nonlinear Equations ..... 4
2.1.1 Newton's Method ..... 4
2.1.2 Broyden's Class of Quasi-Newton Methods for Non- Linear System of Equations. ..... 9
2.1.3 Diagonal Broyden-Like Method for Systems of Nonlinear Equations. ..... 12
2.1.4 Homotopy and Continuation Method ..... 14
2.2 Unconstrained Optimization ..... 23
2.2.1 Steepest Descent Method ..... 27
2.2.2 Conjugate Gradient Method ..... 28
2.2.3 Modified Newton's Method ..... 31
2.2.4 Quasi - Newton's Method for Optimization ..... 33
CHAPTER 3: CONVERGENCE ANALYSIS
3.1 Preliminaries ..... 35
3.2 Convergence Analysis of Newton's method ..... 36
3.3 Convergence of Broyden's Method. ..... 42
3.4 Convergence of Diagonal Broyden-Like Method ..... 47
3.5 Convergence of Steepest Descent Method ..... 51
3.6 Convergence of Conjugate Gradient Method ..... 62
3.7 Convergence of Newton's Method for Optimization ..... 65
3.8 Convergence of Quasi-Newton's Method for Optimization ..... 68
CHAPTER 4: METHODOLOGY
4.1 Bench Mark Problems ..... 72
4.2 Step by Step Excel Solution of (1) ..... 73
4.2.1 Solution using Newton's Method ..... 73
4.2.2 Solution using Quasi-Newton's Method. ..... 76
4.2.3 Solution using Diagonal Broyden-Like Method ..... 78
4.2.4 Solution using Homotopy and Continuation Method ..... 80
4.3 Step by Step Excel Solution of (2) ..... 90
4.3.1 Solution using Steepest Descent Method ..... 92
4.3.2 Solution using Conjugate Gradient Method ..... 97
4.3.3 Solution using Modified Newton's Method ..... 99
4.3.4 Solution using Quasi-Newton's DFP (Davidon, Fletcher, Powell) ..... 101
4.3.4 Solution Using Quasi-Newton's BFGS ..... 103
CHAPTER 5: CONCLUSION AND DISCUSSION ..... 106
REFERENCES ..... 108

## LIST OF TABLES

Table 1.1: Excel Showing Descent Direction Calculation Result ..... 19
Table 2.1: Benchmark 1 Solution using Excel (Newton's Method) ..... 70
Table 2.2: Benchmark 1 Solution using Excel Quasi-Newton's Method ..... 72
Table 2.3: Benchmark 1 Solution using Excel (Diagonal Broyden-like Method)... ..... 74
Table 2.4: Benchark 1 Solution using Homotopy and Continuation Method ..... 76
Table 2.5: Steplength Calculation using Excel (Newton - Raphson Method) ..... 86
Table 2.6: Benchmark 2 Solution using Excel (Steepest Descent Method) ..... 88
Table 2.7: Benchmark 2 Solution using Excel (Conjugate Gradient Method) ..... 93
Table 2.8: Benchmark 2 Solution using Excel (Modified Newton's Method) ..... 95
Table 2.9: Benchmark 2 Solution using Excel (DFP Method) ..... 97
Table 2.10: Benchmark 2 Solution using Excel (BFGS Method) ..... 100

## LIST OF ABBREVIATIONS

$\mathbf{C}(\mathbf{X})$ Set of all functions continuous on X
$\mathbf{C}^{\mathbf{n}}(\mathbf{X})$ Set of all functions having n continuous derivatives on X
$\mathbf{C}^{\infty}(\mathbf{X})$ Set of all functions having derivatives of all orders on X
R Set of all real numbers
O (.) Order of convergence
$\mathbf{x} \quad$ Vector or element in $\mathrm{R}^{\mathrm{n}}$
$\mathrm{R}^{\mathrm{n}} \quad$ Set of all ordered $n$-tuples of real numbers
$\rightarrow \quad$ Equation replacement
$\mathbf{A}^{-1} \quad$ Inverse matrix for matrix A
$\mathbf{A}^{\mathbf{T}} \quad$ Transpose of the matrix A
$\operatorname{det} \mathbf{A}$ Determinant of the matrix A
$\|\mathbf{x}\| \quad$ A norm of the vector $\mathbf{x}$
$\|\mathbf{x}\|_{2} \quad$ The $1_{2}$ norm of the vector $\mathbf{x}$
$\|\mathbf{x}\|_{\infty} \quad$ The $1_{\infty}$ norm of the vector $\mathbf{x}$
F Function mapping $\mathrm{R}^{\mathrm{n}}$ into $\mathrm{R}^{\mathrm{n}}$
$\mathrm{A}(\mathbf{x}) \quad$ Matrix whose entries are functions
$\mathbf{J}(\mathbf{x}) \quad$ Jacobian matrix
$\nabla \mathbf{g} \quad$ Gradient of $g$
(x) Parenthesis referring to equation number
[x] Square brackets referring to reference number
$\boldsymbol{x}^{*} \quad$ Scalar
$\boldsymbol{x}^{*} \quad$ Vector
$\boldsymbol{f}^{\prime} \quad$ Derivative of f
$\frac{\partial f}{\partial x} \quad$ Partial derivative of f with respect to x
$\boldsymbol{s}_{\boldsymbol{k}} \quad$ Correction to the previous iteration
$\mathbf{H}_{\mathrm{k}} \quad$ Jacobian approximation
$\mathbf{H}_{1} \quad$ Arbitrary nonsingular matrix
$\operatorname{Tr}$ (.) Trace operator

M Arbitrary constant
$\alpha_{k} \quad$ Steplength
$\mathbf{x}^{(\mathbf{n})} \quad$ Value of x at iteration n

## CHAPTER 1

## INTRODUCTION

### 1.1 About Optimization

In general the numerical optimization is classified in to two branches; constrained and unconstrained optimization. In this thesis we are only concerned the second part namely the unconstrained optimization that involves the minimization of real-valued objective function $f(x)$, that is finding:

$$
\min f(x) \text { or } \max (-f(x))
$$

These types of problems arise practically in almost every branch of science and also in other disciplines. An engineer needs to design a structure that can carry maximum load with possibly minimum cost. Manufacturers aim to design their products to maximize revenue and minimize cost. Scientists and other designers often look for mathematical functions that describe their data with minimum discrepancy. All these are just few examples on how the minimization problems come about and finding the best solutions is one of their top priorities.

These problems can vary from a simple function of single independent variable to functions of n independent variables. Solutions of the problem for $\mathrm{n}=1$ is simply dealt with by differentiating the function and finding the critical points. The complexity of the solution depends on the nonlinearity of function and the size of $n$ that can reach 100 or more, in which case the problems have no exact close form solutions. It is then, where approximate numerical solutions are sought. The numerical methods in general are based on sequences generated by iterations with the hope that the sequence will converge to the exact solution.

The number of iterations required to solve such an optimization problem, depending on its complexity, can reach thousands of iterations demanding vast amount of computer resources in terms of cpu time and storage. Therefore, the research area in dealing with various aspects
of the optimization is immense and it is ongoing. Generally instructors and researchers use one of two computing methods for solving optimization problems.

They either use a readily available software packages such as Maple, Mathematica, Matlab...etc or they write their own programs using Fortran, Basic, Pascal, ...etc.

From the teaching point of view, the software packages work like a box where the user enters information from one end to get the answers from the other end, without him realizing or understanding what has happened between the two ends and how the results were obtained. While writing own program in Fortran ... etc demands a lot of programming skills that the user has to learn and a class room time is never enough for that. In addition often these programs need to be purchased and can be costly. However, in this work we extensively demonstrate in detail how the Excel sheet can be used to solve a variety of problems with some easy to learn procedures and with the advantage that the user is involved in the step by step implementation of the numerical methods, hence striking a balanced alternative between the two options above.

### 1.2 About this thesis

This thesis gives an in-depth review of the classical methods for solving systems of nonlinear equations and unconstrained minimization of real-valued functions. Mathematical theory is presented and Excel spread sheet is used for implementing and testing these methods using some benchmark problems. An extensive set of numerical test results is also provided. It covers a range of methods for Solving Systems of Nonlinear Equations and Numerical Optimization. For Solving Systems of Nonlinear Equations, it includes the methods of Newton's, Quasi-Newton's, Diagonal Broyden-like and Homotopy and Continuation. For Optimization it covers, the Steepest descent method, the Fletcher-Reeves and Polak-Ribière conjugate gradient methods, the Modified Newton's method and the quasi-Newton BFGS and DFP method, using Analytic line search method to calculate the Steplength. In addition, some benchmark problems are used to describe the methodology.

Chapter 2 surveys the theoretical background and the literature review of the methods that are covered by this thesis. An outline of the derivation of each method is given with their
respective algorithms. The characteristic properties of these methods and their connections to practical implementations are discussed. Since different methods are discussed in the thesis, then some methods are to be preferred than the others in many respect such as their speed of convergence, work needed to apply the method and problems that may arise with respect to convergence and singularity etc. Convergence Analysis of the methods and their rate of convergence are discussed in Chapter 3.

The main contribution of this thesis is the implementation of procedure and algorithm using Excel spread sheet. The availability of Microsoft Excel on most personal computers makes optimization so much easier to teach and learn. The programming with excel is very simple and straight forward and error and algorithm failure detection is also straightforward and visible immediately. The effect of changing a value such as the initial guess is instantaneous without the need for the processes of loading, compiling and executing as in some high level programs such as FORTRAN,C++... for example. An overview of the characteristics of Excel spreadsheet is also given in Chapter 3.

Due to the computational nature of solving problems involving systems of nonlinear equation and unconstrained Optimization, testing of algorithms is an essential part of this thesis. Different approaches for evaluating performances of the algorithms are presented in Chapter 4. A comprehensive performance comparison of the reviewed algorithms is given. Also the specific characteristics of each algorithm are analyzed experimentally with illustrations using some benchmark problems. Some of their theoretical results are also experimentally verified. Finally, Chapter 5 summarizes this thesis and gives some recommendations as per as teaching Numerical Analysis and unconstrained Optimization using Excel is concerned.

## CHAPTER 2

## BACKGROUND AND LITERATURE REVIEW

### 2.1 Systems of Nonlinear Equations

Consider the system of nonlinear equations

$$
\begin{align*}
& f_{1}\left(x_{1}, x_{2}, x_{3}, \ldots, x_{n}\right)=0 \\
& f_{2}\left(x_{1}, x_{2}, x_{3}, \ldots, x_{n}\right)=0 \\
& \text {. }  \tag{2.1}\\
& f_{n}\left(x_{1}, x_{2}, x_{3}, \ldots, x_{n}\right)=0
\end{align*}
$$

The above system can be denoted by $\mathbf{F}(\mathbf{x})=\mathbf{0}$, where $\mathbf{x}, \mathbf{0}$ and $\mathbf{F}$ in bold face print are vectors with $\mathbf{F}=\left(f_{1}, f_{2}, \ldots, f_{n}\right): R^{n} \rightarrow R^{n}$ is continuously differentiable in an open neighborhood $\Delta^{*}$ $\subset \Delta$ of a solution $\boldsymbol{x}^{*} \in \Delta$ of the system, where $\mathbf{F}\left(\boldsymbol{x}^{*}\right)=\mathbf{0}$ and the Jacobian matrix of $\mathbf{F}$ at $\boldsymbol{x}^{*}$ is given by $\boldsymbol{J}(\boldsymbol{x})=\boldsymbol{F}^{\prime}\left(\boldsymbol{x}^{*}\right)$ that is a nonsingular matrix. There are many iterative methods for solving (1) which include Newton's method, Quasi-Newton's method, Diagonal Broydenlike method, and Homotopy and Continuation method.
2.1.1 Newton's Method: Around 1669, Isaac Newton (1643-1727) gave a new algorithm for solving a polynomial equation [1], His algorithm was illustrated by the example $y^{3}-2 y-5$ $=0$. He first used a starting value $\mathrm{y}=2$ with an absolute error being 1 . Then he used $\mathrm{y}=2+$ p to get,

$$
p^{3}+6 p^{2}+10 p-1=0
$$

Newton assumed the $p$ value to be very small, hence he neglected $p^{3}+6 p^{2}$ and used $10 p-1$ and the above equation gives $\mathrm{p}=0.1$, therefore a better approximation of the root is $\mathrm{y}=2.1$ with an absolute error being 0.061 a big improvement. It is possible to repeat this process and write
$\mathrm{y}=2.1+\mathrm{q}$, the substitution gives: $\quad q^{3}+6.3 q^{2}+11.23 q+0.061=0$

Again assuming that q is small and ignoring the terms with higher order of q .
gives $\mathrm{q}=\frac{-0.061}{11.23}=-0.0054 \ldots$, and a new approximation is $\mathrm{y}=2.0946$ with an absolute error being 0.000542 , and so on, the process can then be repeated until the required accuracy is attained. Newton used this method only for polynomial equations.

And as it can be seen, he did not use the concept of derivative at all.

Raphson's iteration: - In 1690, a new step was made by Joseph Raphson (1678-1715), He proposed a method [2] which circumvented the substitutions in Newton's approach. His algorithm was on the equation $x^{3}-b x+c=0$, and starting with an approximate solution of the above equation say $\mathrm{g} \approx \mathrm{x}$, a better approximation was given by

$$
x \approx g+\frac{g^{3}-b g+c}{b-3 g^{2}}
$$

Note, that the denominator of the fraction is the negative of the derivative of the function.

This was the historical beginning of Newton-Raphson's algorithm.

Later studies: The method was then studied and generalized by other mathematicians like Simpson (1710-1761), Mourraille (1720-1808), Cauchy (1789-1857), Kantorovich (19121986) ... The aspect of the choice of the starting point was first tackled by Mourraille in 1768 and the difficulty to make this choice is the main dra wback of the algorithm [3].

Newton-Raphson Iteration: Nowadays, Newton-Raphson's method is a generalized process to find an accurate root of a single equation $f(x)=0$. Suppose that $f$ is a $C^{2}$ function on a given interval and $\mathrm{x}^{*}=\mathrm{x}+\mathrm{h}$, then using Taylor's expansion about x :

$$
f(x+h)=f(x)+h f^{\prime}(x)+O\left(h^{2}\right)
$$

Truncating after the second term,

$$
f(x+h)=f\left(x^{*}\right)=0=f(x)+h f^{\prime}(x)
$$

giving,

$$
\begin{aligned}
& h=-\frac{f(x)}{f^{\prime}(x)} \\
& x+h=x-\frac{f(x)}{f^{\prime}(x)}
\end{aligned}
$$

The convergence is quadratic (Convergence analysis will be discussed in Chapter 3)

Newton's method for several variables: Newton's method can also be used to find a root of a system of two equations

$$
f(x, y)=0, \quad g(x, y)=0
$$

Where f and g are $\mathrm{C}^{2}$ functions on a given domain. Using Taylor's expansion of the two functions near ( $\mathrm{x}, \mathrm{y}$ ) assuming $\mathrm{x}^{*}=\mathrm{x}+\mathrm{h}$ and $\mathrm{y}^{*}=\mathrm{y}+\mathrm{h}$ one gets,

$$
\begin{aligned}
& f(x+h, y+k)=f(x, y)+h \frac{\partial f}{\partial x}+k \frac{\partial f}{\partial y}+O\left(h^{2}+k^{2}\right) \\
& g(x+h, y+k)=g(x, y)+h \frac{\partial g}{\partial x}+k \frac{\partial g}{\partial y}+O\left(h^{2}+k^{2}\right)
\end{aligned}
$$

Truncating after the first order terms, means the couple (h,k) are such that,

$$
\begin{aligned}
& f(x+h, y+k)=0 \approx f(x, y)+h \frac{\partial f}{\partial x}+k \frac{\partial f}{\partial y} \\
& g(x+h, y+k)=0 \approx g(x, y)+h \frac{\partial g}{\partial x}+k \frac{\partial g}{\partial y}
\end{aligned}
$$

Hence, it's equivalent to the linear system

$$
\left[\begin{array}{ll}
\frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\
\frac{\partial g}{\partial x} & \frac{\partial g}{\partial y}
\end{array}\right]\left[\begin{array}{l}
h \\
k
\end{array}\right]=-\binom{f(x, y)}{g(x, y)}
$$

The $(2 \times 2)$ matrix is called the Jacobian matrix and it is denoted as,

$$
J=\left[\begin{array}{ll}
\frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\
\frac{\partial g}{\partial x} & \frac{\partial g}{\partial y}
\end{array}\right]
$$

The equation of generating the sequence $x_{n}$ and $y_{n}$ is given by:

$$
\left[\begin{array}{l}
\boldsymbol{x}_{n+1} \\
\boldsymbol{y}_{n+1}
\end{array}\right]=\left[\begin{array}{l}
\boldsymbol{x}_{n} \\
\boldsymbol{y}_{n}
\end{array}\right]-\boldsymbol{J}^{-1}\left(\boldsymbol{x}_{n}, \boldsymbol{y}_{n}\right)\left[\begin{array}{l}
f\left(\boldsymbol{x}_{n}, \boldsymbol{y}_{n}\right) \\
g\left(\boldsymbol{x}_{n}, \boldsymbol{y}_{n}\right)
\end{array}\right]
$$

The procedure above can be extended in a similar manner to 3 or n variables.

The rate of convergence is Quadratic for an initial point $\boldsymbol{x}_{\mathbf{0}}$ in the neighborhood of the solution say $\boldsymbol{x}^{*}$ when the Jacobian matrix is nonsingular (Dennis (1983) as it was referred in [3]) (Convergence analysis will be discussed in Chapter 3).

## Algorithm Newton's Iteration

1. For a single nonlinear equation given $\mathrm{x}_{0}$,

Step 1: Compute $f\left(x_{k}\right) \quad \mathrm{k}=0,1,2, \ldots$

Step 2: Compute $f^{\prime}\left(x_{k}\right)$

Step 3: $x_{k+1}=x_{k}-\frac{f\left(x_{k}\right)}{f^{\prime}\left(x_{k}\right)}$
2. For a system of non-linear equations given $\mathbf{x}_{\mathbf{0}}$,

Step 1: Compute $\boldsymbol{s}_{\boldsymbol{k}}=-\boldsymbol{J}^{-1}\left(\boldsymbol{x}_{\boldsymbol{k}}\right) \mathbf{F}\left(\boldsymbol{x}_{\boldsymbol{k}}\right)$ for $\mathrm{k}=0,1,2 \ldots$

Step 2: Update $\boldsymbol{x}_{\boldsymbol{k}+\boldsymbol{1}}=\boldsymbol{x}_{\boldsymbol{k}}+\boldsymbol{s}_{\boldsymbol{k}}$.

Where $\boldsymbol{J}\left(\boldsymbol{x}_{\boldsymbol{k}}\right)$ is the Jacobian matrix of $\mathbf{F}$ at $\boldsymbol{x}_{\boldsymbol{k}}, \boldsymbol{s}_{\boldsymbol{k}}$ is the correction to the previous iteration.

From a computational point of view Newton's method can be too expensive for large systems due to,

- The computation of the Jacobian elements, which are $\mathrm{n}^{2}$ first derivatives, if performed analytically can be expensive.
- The Jacobian may be singular at $\mathbf{x}_{\mathrm{k}}$.
- The computation of the next step requires for problems with full Jacobian $O\left(n^{3}\right)$ multiplications [5] and [6] which may be costly for large n.


### 2.1.2 Broyden's Class of Quasi-Newton Methods for Non- Linear System of Equations

Broyden was a Physicist working in an electric industry company. He had to solve a problem involving non-linear algebraic equations. Broyden was well aware of the shortcomings of the Newton's method and thought of the way to overcome them. He realized that he doesn't necessarily need to work with the true inverse Jacobian, but with a suitable approximation $\mathrm{H}_{\mathrm{k}}$ to it. Thus one would get an iteration of the form,

$$
\boldsymbol{x}_{k+1}=\boldsymbol{x}_{k}-\boldsymbol{H}_{k} g\left(\boldsymbol{x}_{k}\right) .
$$

He noticed that, from the Taylor expansion if truncated at first term, where $\boldsymbol{g}_{k}=\boldsymbol{g}\left(\boldsymbol{x}_{k}\right)$, one gets the relation

$$
\boldsymbol{g}_{k+1}=\boldsymbol{g}_{k}+\boldsymbol{J}_{k}\left(\boldsymbol{x}_{k+1}-\boldsymbol{x}_{k}\right)
$$

Or alternatively, with

$$
\begin{gathered}
\boldsymbol{s}_{k}=\boldsymbol{x}_{k+1}-\boldsymbol{x}_{k} \quad \boldsymbol{y}_{k}=\boldsymbol{g}_{k+1}-\boldsymbol{g}_{k} \\
\boldsymbol{J}_{k}^{-1} \boldsymbol{y}_{k}=\boldsymbol{s}_{k}
\end{gathered}
$$

Broyden proposed $\mathrm{H}_{\mathrm{k}}$ for the approximation of the inverse Jacobian and that the following equation to be satisfied, which He called the quasi-Newton equation and other mathematician, called it the secant equation.

$$
\boldsymbol{H}_{k+1} \boldsymbol{y}_{k}=\boldsymbol{s}_{k}
$$

The above equation is a system of $n$ linear equations in $n^{2}$ variables, the components of the approximate Jacobian of $g_{k}$. Therefore it is an underdetermined linear system with an infinite number of solutions. The general solution appears in [7] and is further generalized to the case with some fixed elements by Spedicato and Zhao [8]. First consider how Broyden
derived the Broyden class, where $\mathrm{H}_{\mathrm{k}}$ is updated by a simple rank-one correction, a class that contains all quasi-Newton methods for nonlinear algebraic systems in the literature. Then look at some results on optimal conditioning obtained by Spedicato and Greenstadt [9] and at a surprising result on the finite termination of methods of Broyden class.

The solution for the quasi-Newton equation that Broyden [10] considered, is the special one given by correction to $\mathbf{H k}$, where $\mathbf{H}_{1}$ is an arbitrary nonsingular matrix, most of the time the identity matrix. Broyden consideredthe update

$$
\mathbf{H}_{\mathrm{k}+1}=\mathbf{H}_{\mathrm{k}}-\mathbf{u}_{\mathrm{k}} \mathbf{v}_{\mathrm{k}}^{\mathrm{T}},
$$

where $\mathbf{u}_{\mathrm{k}}, \mathbf{v}_{\mathrm{k}}$ are n-dimensional vectors.

From the above formula and also from the quasi-Newton formula he got the following formula, which defines the Broyden class of quasi-Newton methods for nonlinear algebraic equations.

$$
\boldsymbol{H}_{k+1}=\boldsymbol{H}_{k}-\frac{\left(\boldsymbol{H}_{k} \boldsymbol{y}_{k}-\boldsymbol{s}_{k}\right) \boldsymbol{v}_{k}^{T}}{\boldsymbol{y}_{k}^{T} \boldsymbol{v}_{k}}
$$

The above is a class of methods with $\mathbf{v}_{\mathrm{k}}$ a free parameter with the condition that the matrices remain nonsingular. Broyden considered in his 1965 paper [10] only three parameter choices for $\boldsymbol{v}_{k}$, which leads to the following three methods:

- First Update formula with $\boldsymbol{v}_{k}=\boldsymbol{H}_{k}^{T} \boldsymbol{s}_{k}$ gives,

$$
\boldsymbol{H}_{k+1}=\boldsymbol{H}_{k}-\frac{\left(\boldsymbol{s}_{k}-\boldsymbol{H}_{k} \boldsymbol{y}_{k}\right) \boldsymbol{s}_{k}^{T} H_{k}}{\boldsymbol{s}_{k}^{T} H_{k} \boldsymbol{y}_{k}}
$$

- Second Update formula with $\boldsymbol{v}_{k}=\boldsymbol{y}_{k}$ gives,

$$
\boldsymbol{H}_{k+1}=\boldsymbol{H}_{k}-\frac{\left(\boldsymbol{s}_{k}-\boldsymbol{H}_{k} \boldsymbol{y}_{k}\right) \boldsymbol{y}_{k}^{T}}{\boldsymbol{y}_{k}^{T} \boldsymbol{y}_{k}}
$$

- Symmetric Update formula with $\boldsymbol{v}_{k}=\left(H_{k} \boldsymbol{y}_{k}-\boldsymbol{s}_{k}\right)$ gives,

$$
\boldsymbol{H}_{k+1}=\boldsymbol{H}_{k}-\frac{\left(\boldsymbol{s}_{k}-H_{k} \boldsymbol{y}_{k}\right)\left(\boldsymbol{s}_{k}-H_{k} \boldsymbol{y}_{k}\right)^{T}}{\left(\boldsymbol{s}_{k}-H_{k} \boldsymbol{y}_{k}\right)^{T} \boldsymbol{y}_{k}}
$$

Broyden's first update formula which he defines as the good method is the most used for this class. Broyden defined the second method as the bad method, because its performance was bad. Other numerical analyst found the performance not very worse than that of the first update formula. The third method, known as SR1 method, was initially considered unsuitable because it can lead to a division by zero. Such a method however, occurs also in Broyden's rank-two class of quasi-Newton methods, being therefore an intersection of the two classes, and lot of work has been done to make use of some of its special properties [11] and [12].

Broyden thought that methods in his class had no finite termination on a linear system. Until when first Gay [13], then O'Leary [14] and Ping [15] proved that such methods under mild conditions find the solution of a general linear system in no more than 2 n steps. The result was fruit of a rather complex analysis. In [16], Broyden's method was shown to be a special case of the finitely terminating class of ABS methods [17]. This result follows by proving that two steps of the Broyden class can be identified with one step of a certain ABS method, though the formula for the ABS parameter is not explicitly available.

The convergence analysis for Broyden's class is available in his definitive paper [18]. It is shown that the methods converge from a starting point sufficiently close to the solution, with a $q$-superlinear rate of convergence. The rate worsens with the increase of dimension. Results on the convergence of the sequence $\left\{\mathbf{H}_{k}\right\}$ are still a subject for investigation. Detail of this convergence will be discussed in Chapter 3.

The Shortcoming of Broyden's method is that, the quadratic convergence in Newton's method is lost been replaced by Superlinear convergence.

### 2.1.3 Diagonal Broyden-Like Method for Systems of Nonlinear Equations

The most critical part of Quasi-Newton's method is on the formation and storage of a full matrix approximation to the Jacobian matrix at every iteration. Some alternative methods are proposed to take care of the short-comings of Newton's method. These weaknesses, together with some other weaknesses of Newton's like methods especially when handling large-scale systems of non-linear equations, leads to the innovation of this method by Waziri et al [19]. It is important to note that, the diagonal updating strategy has been applied in unconstrained optimization problems [20], [21], [22], [23], and [24].

This method attempts to provide a different approximation to the Newton's step via diagonal updating by means of variational techniques. It is worth mentioning that the new updating scheme has been applied to solve (1) without the cost of computation or storage of the Jacobian matrix. This may reduce: computational cost, matrix storage requirement, CPU time and eliminates the needs of solving $n$ linear equations at each iteration. The diagonal updating method works very efficient and the results are very reliable. In addition, this method can also solve some problems, which cannot be solved by methods involving Jacobian matrix computation [19].

## Algorithm DBLM (Diagonal Broyden-Like method)

Given $\boldsymbol{x}_{0}$, and $\boldsymbol{Q}_{0}$, set $k=0$

Step 1: Compute $\boldsymbol{F}\left(\boldsymbol{x}_{k}\right)$ and $\boldsymbol{x}_{k+1}=\boldsymbol{x}_{k}-\boldsymbol{Q}_{k} F\left(\boldsymbol{x}_{k}\right)$

Step 2: If $\left\|\boldsymbol{s}_{k}\right\|_{\infty}+\left\|F\left(\boldsymbol{x}_{k}\right)\right\|_{\infty} \leq \epsilon_{1}$ stop. Else go to Step 3.

Step 3: If $\left\|\boldsymbol{y}_{k}\right\|_{\infty} \geq \epsilon_{1}$, compute $\boldsymbol{Q}_{k+1}$, else, $\boldsymbol{Q}_{k+1}=\boldsymbol{Q}_{k}$. Set $\mathrm{k}=\mathrm{k}+1$ and go to step 1.

The update $\left(Q_{k+1}\right)$ formula is given by,

$$
\begin{equation*}
Q_{k+1}=Q_{k}+\frac{\left(y_{k}^{T} s_{k}-y_{k}^{T} Q_{k} y_{k}\right)}{\operatorname{Tr}\left(A_{k}^{2}\right)} A_{k} \tag{4}
\end{equation*}
$$

Where $\boldsymbol{A}_{k}=\operatorname{diag}\left(\boldsymbol{y}_{k}^{(1)^{2}}, \boldsymbol{y}_{k}^{(2)^{2}}, \ldots, \boldsymbol{y}_{k}^{(n)^{2}}\right), \operatorname{Tr}\left(A_{k}^{2}\right)=\sum_{i=1}^{n} \boldsymbol{y}_{k}^{(i)^{4}}$ and $\operatorname{Tr}($.$) is the trace operator.$

To safeguard the possibly very small $\left\|\boldsymbol{y}_{k}\right\|$ and $\operatorname{Tr}\left(A^{2}\right)$ it is required that $\left\|\boldsymbol{y}_{k}\right\|_{\infty} \geq \epsilon_{1}$ for some chosen small $\epsilon_{1}>0$. Else set $Q_{k+1}=Q_{k}$, hence,
$\boldsymbol{Q}_{k+1}$ is given as: $\boldsymbol{Q}_{k+1}=\left\{\begin{array}{c}\boldsymbol{Q}_{\mathbf{k}}+\frac{\left(\mathbf{y}_{\mathbf{k}}^{\mathrm{T}} \boldsymbol{s}_{\mathbf{k}}-\mathbf{y}_{\mathbf{k}}^{\mathrm{T}} \mathbf{Q}_{\mathbf{k}} \mathbf{y}_{\mathbf{k}}\right)}{\operatorname{Tr}\left(\mathbf{A}_{\mathbf{k}}^{2}\right)} \mathbf{A}_{\mathbf{k}} ;\left\|\boldsymbol{y}_{k}\right\|_{\infty} \geq \epsilon_{1} \\ \boldsymbol{Q}_{\mathbf{k}} \quad ; \quad \text { otherwise }\end{array}\right.$

The Convergence analysis will be discussed in Chapter 3.

### 2.1.4 Homotopy and Continuation Method

A Homotopy is a continuous deformation; a function that takes a real interval continuously into a set of functions.

Homotopy or continuation, methods for nonlinear systems embed the problem to be solved within collection of problems. Specifically, to solve a problem of the form

$$
\boldsymbol{F}(\boldsymbol{x})=0
$$

Which has the unknown solution $\boldsymbol{x}^{*}$, we consider a family of problems described using parameter a $\lambda$ that assumes value in $[0,1]$. A problem with a known solution $\mathbf{x}$ (0) corresponds to the situation where $\lambda=0$, and the problem with the unknown solution $\boldsymbol{x}(1) \equiv \boldsymbol{x}^{*}$ corresponds to $\lambda=1$.

For example, suppose $\mathbf{x}(0)$ is an initial approximation to the solution of $F\left(\boldsymbol{x}^{*}\right)=0$. Define

$$
\mathrm{G}:[0,1] \times R^{n} \rightarrow R^{n}
$$

By

$$
G(\lambda, \mathbf{x})=\lambda \mathbf{F}(\mathbf{x})+(1-\lambda)[\mathbf{F}(\mathbf{x})-\mathbf{F}(\mathbf{x}(0))]=\mathbf{F}(\mathbf{x})+(\lambda-1) \mathbf{F}(\mathbf{x}(0)) .
$$

Then for various values of $\lambda$ the solution to $G(\lambda, x)=0$, can be found.

When $\lambda=0$ this equation assumes the form

$$
0=\mathbf{G}(0, \mathbf{x})=\mathbf{F}(\mathbf{x})-\mathbf{F}(\mathrm{x}(0)),
$$

And $\mathbf{x}(0)$ is a solution. when $\lambda=1$, the equation assumes the form

$$
0=\mathbf{G}(1, \mathrm{x})=\mathbf{F}(\mathrm{x})
$$

And $\mathbf{x}(1)=\boldsymbol{x}^{*}$ is a solution.
The function G, with parameter $\lambda$, provides us with a family of functions that can lead from the known value $\mathrm{x}(0)$ to the solution $\mathbf{x}(1)=\boldsymbol{x}^{*}$. The function G is called a homotopy between the function $G(0, \mathbf{x})=\mathbf{F}(\mathbf{x})-\mathbf{F}(\mathbf{x}(0))$ and the function $G(1, \mathbf{x})=\mathbf{F}(\mathbf{x})$.
The Continuation problem is to determine a way to proceed from the known solution $\mathbf{x}(0)$ of $\mathrm{G}(0, \mathbf{x})=0$ to the unknown solution $\mathbf{x}(1)=\boldsymbol{x}^{*}$ of $\mathrm{G}(1, \mathbf{x})=0$, that is the solution to $\mathbf{F}(\mathbf{x})=0$.

We first assume that $\mathbf{x}(\lambda)$ is the unique solution to the equation

$$
\mathrm{G}(\lambda, \mathbf{x})=0,
$$

For each $\lambda \in[0,1]$. The set $\{\mathbf{x}(\lambda) \mid 0 \leq \lambda \leq 1\}$ can be viewed as a curve in $R^{n}$ from $\mathrm{x}(0)$ to
$\mathbf{x}(1)=\boldsymbol{x}^{*}$ parameterized by $\lambda$. A continuation method finds a sequence of steps along this curve corresponding to $\left\{\boldsymbol{x}\left(\lambda_{k}\right)\right\}_{k=0}^{m}$ where $\lambda_{0}=0<\lambda_{1}<\cdots<\lambda_{m}=1$.

If the function $\lambda \rightarrow \boldsymbol{x}(\lambda)$ and $G$ are differentiable then differentiating $\mathrm{G}(\lambda, \mathbf{x})=0$, with respect to $\lambda$ gives

$$
0=\frac{\partial G(\lambda, x(\lambda))}{\partial \lambda}+\frac{\partial G(\lambda, x(\lambda))}{\partial x} x^{\prime}(\lambda)
$$

And solving for $\boldsymbol{x}^{\prime}$ gives

$$
x^{\prime}(\lambda)=-\left[\frac{\partial G(\lambda, x(\lambda))}{\partial x}\right]^{-1} \frac{\partial G(\lambda, x(\lambda))}{\partial \lambda}
$$

This is a system of differential equations with the initial condition $\mathbf{x}(0)$.

Since,

$$
\mathrm{G}(\lambda, \mathbf{x})=\mathrm{F}(\mathbf{x})+(\lambda-1) \mathbf{F}(\mathbf{x}(0)) .
$$

We can determine both

$$
\frac{\partial G \mathrm{G}(\lambda, \mathbf{x})}{\partial x}=J(x(\lambda))
$$

The Jacobian matrix, and

$$
\frac{\partial G(\lambda, x(\lambda))}{\partial \lambda}=\boldsymbol{F}(\boldsymbol{x}(0))
$$

Therefore, the system of differential equation becomes

$$
x^{\prime}(\lambda)=-\left[J(x(\lambda))^{-1} F(x(0)), \quad \text { for } 0 \leq \lambda \leq 1,\right.
$$

with the initial condition $\mathbf{x}(0)$. The following theorem gives conditions under which the continuation method is feasible.

Theorem 1: Let $\mathbf{F}(\mathbf{x})$ be continuously differentiable for $\mathbf{x} \in R^{n}$. Suppose that the Jacobian matrix $\mathbf{J}(\mathbf{x})$ is nonsingular for all $\mathbf{x} \in R^{n}$ and that a constant M exists with $\left\|\mathbf{J}(\boldsymbol{x})^{-1}\right\| \leq \mathrm{M}$, for all $\mathbf{x} \in R^{n}$. Then for any $\mathbf{x}(0)$ in $R^{n}$, there exists a unique function $\mathbf{x}(\lambda)$, such that

$$
G(\lambda, \mathbf{x})=0
$$

For all $\lambda \in[0,1]$. Moreover, $\mathbf{x}(\lambda)$ is continuously differentiable and

$$
\boldsymbol{x}^{\prime}(\lambda)=-\left[J(\boldsymbol{x}(\lambda))^{-1} \boldsymbol{F}(\boldsymbol{x}(0))\right], \quad \text { for each } \lambda \in[0,1] .
$$

The Continuation method can be used as a stand-alone method, and does not require a particular good choice of $\mathbf{x}(0)$. However the method can be used to give an initial approximation for Newton's or Broyden's method [25].

### 2.2 Unconstrained Optimization

Optimization can be defined in a classical sense, as the art of obtaining best policies to satisfy certain objectives, sometimes satisfying some fixed requirements. Optimization can be categorized into constrained and unconstrained optimization. In this thesis we are only concerned with unconstrained optimization.

Unconstrained Optimization: Unconstrained Optimization is the problem of finding a vector $\boldsymbol{x}$ that is a local minimum or maximum to a scalar function $f(x)$ :

$$
\min _{x} f(x) \text { or } \max _{x}(-f(x))
$$

The term unconstrained means that no restriction is placed on the range of $\mathbf{x}$.

Basics for Unconstrained Optimization: Although many methods exist for unconstrained optimization, the methods can be broadly categorized in terms of the derivative information that is used. Search methods that use only function evaluations are most suitable for problems that are not smooth or have a number of discontinuities. Gradient methods are more efficient when the function to be minimized / maximized is continuous in its first derivative. Higher order methods (such as Newton's method) are only suitable when the second-order derivative can easily be calculated, this is because the calculation of secondorder derivative, using numerical differentiation, is computationally expensive.

To minimize $f(\mathbf{x})$,

The basic Iteration for all the methods here can be written as follows:

$$
x^{k+1}=x^{k}+\alpha_{k} d^{k} \quad k=0,1, \ldots
$$

Where $\boldsymbol{d}^{k}$ is known as the descent direction, and $\alpha_{k}$ is a scalar known as the steplength. The starting point $\boldsymbol{x}^{0}$ is chosen arbitrarily. At each iteration $\alpha_{k}$ and $\boldsymbol{d}^{k}$ are chosen such that $f\left(x^{k+1}\right)<f\left(x^{k}\right)$.

The iteration is terminated when the given convergence criteria is attained. Since the necessary condition for the minimum of unconstrained problem is that, its gradient is 0 at the optimum, the convergence criterion is given as:

$$
\left\|\nabla f\left(x^{k+1}\right)\right\|<t o l
$$

Where the tolerance (tol) is a small number (e.g. $10^{-5}$ ).

Descent Direction: For a given direction to be a direction of descent, the following condition must be satisfied,

$$
f\left(\boldsymbol{x}^{k+1}\right)<f\left(\boldsymbol{x}^{k}\right)
$$

Or

$$
f\left(\boldsymbol{x}^{k}+\alpha_{k} d^{k}\right)<f\left(\boldsymbol{x}^{k}\right)
$$

Using Taylor series expansion,

$$
f\left(\boldsymbol{x}^{k}\right)+\alpha_{k} \nabla f\left(\boldsymbol{x}^{k}\right)^{T} d^{k}<f\left(\boldsymbol{x}^{k}\right)
$$

Or $\quad \alpha_{k} \nabla f\left(x^{k}\right)^{T} d^{k}<0$

If the steplength $\alpha_{k}$ is restricted to positive values, then the following is the criteria for $\boldsymbol{d}^{k}$ at a descent direction when given point $\boldsymbol{x}^{k}$ :

$$
\nabla f\left(\boldsymbol{x}^{k}\right)^{T} \boldsymbol{d}^{k}<0
$$

Furthermore, the numerical value of the product $\nabla f\left(\boldsymbol{x}^{k}\right)^{T} \boldsymbol{d}^{k}$ indicates how fast the function is decreasing along this direction.

Example: Use Excel to check for the following function, if the given directions $\mathbf{d}_{1}, \mathbf{d}_{2}$, are directions of descent at the given point $\boldsymbol{x}^{k}$ :

$$
\begin{aligned}
& f(\boldsymbol{x})=\left(x_{1}^{2}+x_{2}-11\right)^{2}+\left(x_{1}+x_{2}^{2}-7\right) \\
& \boldsymbol{d}^{1}=(1,1) ; \boldsymbol{d}^{2}=(-1,1) \text { and } \boldsymbol{x}^{k}=(1,2)
\end{aligned}
$$

Solution:

$$
\nabla f=\left[\begin{array}{c}
1-44 x_{1}+4 x_{1}^{3}+4 x_{1} x_{2} \\
-22+2 x_{1}^{2}+4 x_{2}
\end{array}\right]
$$

Table 1.1: Excel showing descent direction calculation result


From the Excel result, $d^{1}<0$ is a descent direction, while $d^{2}>0$ is not.

Numerical Optimization Method: At each iteration of a numerical Optimization method, there is need to determine two things 1) Descent direction $\left(d_{k}\right)$, and 2) Step length $\left(\alpha_{k}\right)$ :

$$
\boldsymbol{x}^{k+1}=\boldsymbol{x}^{k}+\alpha_{k} d_{k}
$$

There are various methods for Step-length calculation, such as Analytic line search, Equal interval search, Section search, Golden section search, Quadratic interpolation method, and approximate line search. However in this thesis, Analytical line search method is considered and the solution is obtained using Newton-Raphson method.

Analytic Line Search: If an explicit expression for $\phi(\alpha)$ is known, the optimum step length can easily be calculated using the necessary and sufficient conditions for the minimum of a function of 1 variable. The necessary condition is $\frac{d \phi}{d \alpha}=0$ and the sufficient condition is

$$
\frac{d^{2} \phi}{d \alpha^{2}}>0
$$

## Algorithm Analytic Line Search (solution using Newton-Raphson method)

Given $\boldsymbol{x}^{0}$, calculate $\mathrm{f}\left(\boldsymbol{x}^{0}\right)$, and $\boldsymbol{d}_{0}$, to find $\alpha$
Step 1: Calculate $\phi(\alpha)=\mathrm{f}\left(\boldsymbol{x}^{0}+\alpha \boldsymbol{d}_{0}\right)$

Step 2: Evaluate $\frac{d \phi(\alpha)}{d \alpha}=g(\alpha)$ and $\frac{d^{2} \phi(\alpha)}{d \alpha^{2}}=g^{\prime}(\alpha)$
Step 3: Apply Newton Raphson method on $g(\alpha)$, to find the optimum value of $\alpha$.
Descent direction: As in the case of Step length calculation, there are also several methods of Descent direction calculation, which includes Steepest Descent method, Conjugate Gradient Method, Modified Newton's Method, and Quasi-Newton Methods.

### 2.2.1 Steepest Descent Method

The steepest descent method, can be traced back to Cauchy (1847), is the simplest gradient method for unconstrained optimization:

$$
\min _{x \in R^{n}} f(x)
$$

Where $\mathrm{f}(\mathrm{x})$ is a continuous and differential function in $R^{n}$. The method has the form:

$$
\boldsymbol{x}_{k+1}=\boldsymbol{x}_{k}+\alpha_{k}\left(-\boldsymbol{g}_{k}\right)
$$

Where $\boldsymbol{g}_{k}=\boldsymbol{g}\left(\boldsymbol{x}_{k}\right)=\nabla \boldsymbol{F}\left(\boldsymbol{x}_{k}\right)$ is the gradient vector of $\mathbf{F}(\boldsymbol{x})$ at the current iterate point $\boldsymbol{x}_{k}$ and $\alpha_{k}>0$ is the stepsize. Because the search direction in the method is the opposite of the gradient direction, it is the steepest descent direction locally, which gives the name of the method. Locally the steepest descent direction is the best direction in the sense that it reduces the objective function as much as possible.

The method is very valuable apart from being used as a starting method for solving systems of nonlinear systems [25].

The algorithm for the method of steepest descent for finding a local minimum for an arbitrary function g from $R^{n}$ into R can be described as follows:

## Algorithm of Steepest Descent Method

Given $\boldsymbol{x}^{0}$

Step 1: Calculate $\mathbf{F}\left(\boldsymbol{x}^{0}\right)$ and $\nabla \mathbf{F}\left(\boldsymbol{x}^{0}\right)$.

Step 2: Calculate the Step length
Step 3: Update the next value $\boldsymbol{x}^{k+1}=\boldsymbol{x}^{k}+\alpha_{k} \boldsymbol{d}_{k}$.

Definition 1 (Gradient of a function): For $\mathrm{g}: R^{n} \rightarrow R$, the gradient of g at
$\boldsymbol{x}=\left(x_{1}, x_{2}, \ldots, x_{n}\right)^{T}$ is denoted $\nabla g(\boldsymbol{x})$ and is defined as

$$
\nabla g(\boldsymbol{x})=\left(\frac{\partial g}{\partial x_{1}}(\boldsymbol{x}), \frac{\partial g}{\partial x_{2}}(\boldsymbol{x}), \ldots, \frac{\partial g}{\partial x_{n}}(\boldsymbol{x})\right)^{T}
$$

The gradient for a multivariable function is similar to the derivative of a single variable function in the sense that a differentiable multivariable function can have a relative minimum at $\mathbf{x}$ only when the gradient at $\mathbf{x}$ is the zero vector.

Though the convergence of the method is linear, but it converges even for poor initial approximations [25]. The analysis of the convergence will be discussed in chapter 3.

### 2.2.2 Conjugate Gradient Method

In the steepest descent method for solving nonlinear optimization the steps are along directions that nullify some of the progress of the others. The basic idea of the conjugate gradient method is to move in the non-disturbance direction. Suppose a line minimization along the direction $u$ is done. Then the gradient $\nabla F$ at that point is perpendicular to $u$, because otherwise one can be able to move further along $u$. Next, one should move along some other direction $v$. In steepest descent $v=-\nabla \mathbf{F}$. In the conjugate gradient method some direction is added to $-\nabla \mathbf{F}$ to become v . v is chosen in such a way that it does not undo the minimization along $u$. In order to be perpendicular to $u$ before and after the movement along v. At least
locally the change in $\nabla \mathbf{F}$ is needed to be perpendicular to $u$. Now observe that a small change $\delta \mathbf{x}$ in $\mathbf{x}$ will produce a small change in $\nabla \mathbf{F}$ given by

$$
\delta(\nabla f)=H f . \delta x
$$

The idea of moving along non-interfering direction leads to the condition

$$
u^{T} \delta(\nabla \boldsymbol{F})=0
$$

And the next move should be along the direction v such that

$$
u^{T} H \boldsymbol{F} v=0
$$

Even though $v$ is not perpendicular to $u$, it is $H \boldsymbol{F}$-orthogonal to $u$.
The connection between $\delta x$ and $\delta(\nabla \mathbf{F})$ in terms of the Hessian HFis a differential relationship. Here it is used for finite motions to the extent that Taylor's approximation of order 2 is valid. Suppose f is expanded around a point y keeping x constant,

$$
f(x+y)=f(y)+\nabla f(y)^{T} x+\frac{1}{2} x^{T} H f x
$$

Thus f looks like quadratic equation. If f is assumed to be quadratic, then the Hessian $H f$ does not vary along directions u and v . Thus the condition above makes sense. With this reasoning as background, one develops the conjugate gradient method for quadratic functions formed from symmetric positive definite matrices. For such quadratic functions, by moving along successive non-interfering directions the conjugate gradient method converges to the global minimum in at most n steps.
For general functions, the conjugate gradient method once near a local minimum, the algorithm converges quadratically to the solution.

Thus, the descent direction $d_{k}$ in the steepest descent method is been corrected as follows;

$$
\boldsymbol{d}_{k}=-\nabla f\left(\boldsymbol{x}^{k}\right)+\beta \nabla \boldsymbol{d}_{k-1}
$$

Where mostly in practice $\beta$ is calculated by one of the following formulae,

Fletcher-Reeve's formula: $\quad \beta=\frac{\left[\nabla f\left(x^{k}\right)\right]^{T} \nabla f\left(x^{k}\right)}{\left[\nabla f\left(x^{k-1}\right)\right]^{T} \nabla f\left(x^{k-1}\right)}$

Polak-Ribiere formula: $\quad \beta=\frac{\left[\nabla \boldsymbol{F}\left(x^{k-1}\right)-\nabla \boldsymbol{F}\left(x^{k}\right)\right]^{T} \nabla \boldsymbol{F}\left(x^{k}\right)}{\left[\nabla \boldsymbol{F}\left(x^{k-1}\right)\right]^{T} \nabla \boldsymbol{F}\left(x^{k-1}\right)}$

The numerator in the Fletcher-Reeve's formula is the square of the norm of the gradient of $f$ at the current point. The case is slightly different in Polak-Ribiere formula because the numerator is slightly modified. The denominator is the same in both cases. Polak-Ribiere formula usually gives better results than Fletcher-Reeve's formula [27].

## Algorithm Conjugate Gradient Method:

## Given $X^{0}$

Step 1: Compute $\mathbf{F}(\mathbf{x}), \nabla \mathbf{F}(\mathbf{x})$ and set $\beta, d_{k-1}$ as zeros for the first iteration.

Step 2: Compute $d_{k}=-\nabla \boldsymbol{F}\left(\mathbf{x}^{k}\right)+\beta \nabla d_{k-1}$ and also calculate the step length $\alpha$.
Step 3: Update the next value $\mathbf{x}^{k+1}=\mathbf{x}^{k}+\alpha_{k} d_{k}$.

This method gives better result in practice than Steepest descent method, and the convergence is also faster.

### 2.2.3 Modified Newton's Method

Newton's method for solving systems of nonlinear equation was discussed in chapter 2.1.1, the difference is not much with Newton's method for Optimization being the Computation of Hessian instead of the Jacobian matrix. As in 2.1.1 the Newton method was derived by considering quadratic approximation of the function using Taylor series:

$$
f\left(\mathbf{x}^{k+1}\right) \approx f\left(\mathbf{x}^{k}\right)+\nabla f\left(\mathbf{x}^{k}\right)^{T} \boldsymbol{d}^{k}+\frac{1}{2}\left(\boldsymbol{d}^{k}\right)^{T} \boldsymbol{H}\left(\mathbf{x}^{k}\right) \boldsymbol{d}^{k}
$$

Where $H\left(x^{k}\right)$ is the Hessian matrix at $x^{k}$. Differentiating with respect to $\boldsymbol{d}^{k}$, one gets

$$
\nabla f\left(\mathbf{x}^{k}\right)+H\left(\mathbf{x}^{k}\right) \boldsymbol{d}^{k}=0
$$

The direction can then be obtained by solving the system, i.e.

$$
d^{k}=-\left[H\left(\mathbf{x}^{k}\right)\right]^{-1} \nabla f\left(\mathbf{x}^{k}\right)
$$

In its original form, the method was used without steplength calculations. Thus, the iterative scheme before, was as follows:

$$
\mathbf{x}^{k+1}=\mathbf{x}^{k}-\left[H\left(\mathbf{x}^{k}\right)\right]^{-1} \nabla f\left(\mathbf{x}^{k}\right)
$$

However, in this form the method may not converge when started from a point that is far away from the optimum. The Modified Newton method uses the direction given by the Newton method and then computes an appropriate steplength along the direction. This is what makes the method very stable. Thus the iterations are as follows:

$$
\mathbf{x}^{k+1}=\mathbf{x}^{k}+\alpha_{k} d^{k} \quad \mathrm{k}=0,1, \ldots
$$

With $d^{k}=-\left[H\left(\mathbf{x}^{k}\right)\right]^{-1} \nabla f\left(\mathbf{x}^{k}\right)$ and $\alpha_{k}$ obtained from minimizing $f\left(\mathbf{x}^{k}+\alpha_{k} d^{k}\right)$.

## Algorithm Modified Newton Method

Choosing a starting value $\mathbf{x}^{0}$,

Step 1: Compute $\mathbf{F}(\mathbf{x})$ and $\nabla \boldsymbol{F}(\boldsymbol{x})$.

Step 2: Compute $\mathrm{H}(\mathbf{x})$ and $[H(\boldsymbol{x})]^{-1}$
Step 3: Calculate $d_{k}=-\left[H\left(\mathbf{x}^{k}\right)\right]^{-1} \nabla f\left(\mathbf{x}^{k}\right)$ and $\alpha_{k}$.

Step 4: Update the next value of X by $\mathbf{x}^{k+1}=\mathbf{x}^{k}+\alpha_{k} d_{k}$.

The convergence of the method is quadratic. Each iteration, however, requires more computations because of the need to evaluate the Hessian matrix and then to solve the system of equations to get the direction [27].

### 2.2.4 Quasi - Newton's Method For Optimization

Consider the Quasi-Newton methods with line searches for finding a local minimum of a function $f(\boldsymbol{x})$, where $\boldsymbol{x} \in R^{n}$. It is assumed that $f \in c^{2}$ with positive definite Hessian $H^{*}$ at a solution $\mathbf{x}^{*}$, although only the gradient $g\left(\mathbf{x}^{k}\right)=\nabla f\left(\mathbf{x}^{k}\right)$ (denoted by $g_{k}$ ) is used in practice, where $\mathbf{x}^{k}$ estimates $\mathbf{x}^{*}$ at iteration k . The Hessian approximation at $\mathbf{x}^{k}$ is denoted by $B_{K}$ and its inverse by $H_{k}$.

At each iteration of quasi-newton's methods, a positive definite Hessian approximation $B_{k}$ is updated to a new approximation $B_{k+1}$ using $\boldsymbol{q}^{k}$ and $\boldsymbol{s}^{k}$ defined by $\boldsymbol{q}^{k}=\nabla f\left(\mathbf{x}^{k+1}\right)-$ $\nabla f\left(\mathbf{x}^{k}\right)$ and $\boldsymbol{s}^{k}=\mathbf{x}^{k+1}-\mathbf{x}^{k}$ respectively, for which the quasi-Newton condition is satisfied. To define the update matrix, several formulae have been proposed. The first formula has been suggested by Davidon in 1959, in a technical report [28], subsequently investigated by Fletcher and Powell, published in 1963, [29], it became known as DFP and published with further detail in 1991 [30]. These authors referred to the corresponding DFP method as a variable metric method and it is also known as the first quasi-Newton method. Other popular quasi-Newton formula is BFGS which was obtained independently by Broyden, Fletcher, Goldfarb and Shanno, in 1970 [31]. These formulae belong to the

Broyden family of updates which has certain useful properties when the line search structure is used. The BFGS method is the most effective, while the DFP method may converge slowly in certain cases [32].

Several overviews on quasi-Newton methods have been published [33], [34], [35], [36], [37], [38] and [39].

## Algorithm Quasi-Newton's method

Given the starting point $\boldsymbol{x}^{0}$,
Step 1: Compute $\mathbf{F}(\mathbf{x})$ and $\nabla \boldsymbol{F}(\boldsymbol{x})$.
Step 2: Determine $d_{k}=-Q^{k} \nabla \boldsymbol{F}\left(\mathbf{x}^{k}\right)$ and $\alpha_{k}$. Where Q is a Jacobian inverse matrix
Step 3: Update the next value of X by $\mathbf{x}^{k+1}=\mathbf{x}^{k}+\alpha_{k} d_{k}$.
Step 4: Update $Q^{k}$ for the subsequent iterations.
Now to update $Q^{k}$ there are different Quasi-newton's methods which differ in the way $Q^{k}$ matrix is updated which includes DFP, BFGS, and SR1 methods. Here only DFP and BFGS methods are considered.

## DFP (Davidon, Fletcher, Powell) Update

The $Q^{k}$ update, that is $Q^{k+1}$ of this method is given by,

$$
Q^{k+1}=Q^{k}+\frac{s^{k}\left(s^{k}\right)^{T}}{\left(q^{k}\right)^{T} s^{k}}-\frac{\left(Q^{k} q^{k}\right)\left(Q^{k} q^{k}\right)^{T}}{\left(q^{k}\right)^{T} Q^{k} q^{k}}
$$

Where

$$
q^{k}=\nabla f\left(\mathbf{x}^{k+1}\right)-\nabla f\left(\mathbf{x}^{k}\right) \text { and } \boldsymbol{s}^{k}=\mathbf{x}^{k+1}-\mathbf{x}^{k}
$$

## BFGS (Broyden, Fletcher, Goldfarb, and Shanon) Update

The $Q^{k}$ update, that is $Q^{k+1}$ of this method is given by,
$Q^{k+1}=Q^{k}+\left(1+\frac{\left(q^{k}\right)^{T} Q^{k} q^{k}}{\left(q^{k}\right)^{T} s^{k}}\right) \frac{s^{k}\left(s^{k}\right)^{T}}{\left(q^{k}\right)^{T} s^{k}}$

$$
-\frac{1}{\left(q^{k}\right)^{T} \boldsymbol{s}^{k}}\left[\left(\boldsymbol{s}^{k}\left(q^{k}\right)^{T} Q^{k}\right)^{T}+\boldsymbol{s}^{k}\left(q^{k}\right)^{T} Q^{k}\right]
$$

Numerical result shows the efficiency of BFGS formula over DFP (M.A. Bhati, 2000).Though the effort of computing Hessian matrix in each iteration by Modified Newton's method is been taken care of by Quasi-Newton's method, but its convergence is super linear, hence the rate of convergence is slower than that of Modified Newton's method.

Detailed of the convergence analysis will be seen in Chapter 3.

## CHAPTER 3

## CONVERGENCE ANALYSIS

3.1.1 Preliminaries : Performance of two or more Algorithms is usually compared by their rate of convergence. That is if

$$
\boldsymbol{x}^{k} \rightarrow \boldsymbol{x}^{*}
$$

the interest is usually on how fast it does happen.

Definition 1: Let $\left\{\boldsymbol{x}^{k}\right\} \in R^{n}$ and $\boldsymbol{x}^{*} \in R^{n}$ be such that $\boldsymbol{x}^{k} \rightarrow \boldsymbol{x}^{*}$, if $\exists r \in[0,1]$ and $k_{0} \in N$ s.t

$$
\frac{\left\|x^{k+1}-x^{*}\right\|}{\left\|x^{k}-x^{*}\right\|} \leq r \quad \forall k>k_{0}
$$

then $\boldsymbol{x}^{k} \rightarrow \boldsymbol{x}^{*}$ linearly

Definition 2: Let $\left\{\boldsymbol{x}^{k}\right\} \in R^{n}$ and $\boldsymbol{x}^{*} \in R^{n}$ be such that $\boldsymbol{x}^{k} \rightarrow \boldsymbol{x}^{*}$, if

$$
\lim _{k \rightarrow \infty} \frac{\left\|x^{k+1}-x^{*}\right\|}{\left\|x^{k}-x^{*}\right\|}=0
$$

then $\boldsymbol{x}^{k} \rightarrow \boldsymbol{x}^{*}$ Super Linearly

Definition 3: Let $\left\{\boldsymbol{x}^{k}\right\} \in R^{n}$ and $\boldsymbol{x}^{*} \in R^{n}$ be such that $\boldsymbol{x}^{k} \rightarrow \boldsymbol{x}^{*}$, then if $\exists M>0$ and $k_{0} \in N$ s.t

$$
\frac{\left\|x^{k+1}-x^{*}\right\|}{\left\|x^{k}-x^{*}\right\|^{2}} \leq M \quad \forall k>k_{0}
$$

then $\boldsymbol{x}^{k} \rightarrow \boldsymbol{x}^{*}$ Quadratically

Quadratic convergence is faster than superlinear convergence, while superlinear convergence is faster than linear convergence.

Estimating Rate of Convergence: Let the error after $n$ steps of an iterative algorithm be $e_{n}=\boldsymbol{x}_{n}-r$, then $e_{n+1}=\boldsymbol{x}_{n+1}-r$. As $n \rightarrow \infty$ from the above definitions,

$$
\begin{gathered}
\lim _{n} \frac{\left|x_{n+1}-r\right|}{\left|x_{n}-r\right|^{\alpha}}=\lambda \\
\Rightarrow \quad\left|e_{n+1}\right| \approx \lambda\left|e_{n}\right|^{\alpha} \text { and }\left|e_{n}\right| \approx \lambda\left|e_{n-1}\right|^{\alpha}
\end{gathered}
$$

Forming the ratio of the above gives,

$$
\frac{\left|e_{n+1}\right|}{\left|e_{n}\right|} \approx \frac{\lambda\left|e_{n}\right|^{\alpha}}{\lambda\left|e_{n-1}\right|^{\alpha}} \approx\left|\frac{\left|e_{n}\right|}{\left|e_{n-1}\right|}\right|^{\alpha}
$$

Solving for $\alpha$, gives

$$
\alpha=\frac{\log \left|\frac{e_{n+1}}{e_{n}}\right|}{\log \left|\frac{e_{n}}{e_{n-1}}\right|}
$$

Using the above, one can approximate the convergence rate $\alpha$ given any two consecutive error ratios.
Theorem 1 (Taylor's Theorem): Let $f:[a, b] \rightarrow R$, also let $f, f^{\prime}, f^{\prime \prime}, \ldots, f^{(n-1)}$ be continuous on $(\mathrm{a}, \mathrm{b})$. Then there exist $c \in(a, b)$ such that,

$$
f(b)=f(a)+f^{\prime}(a)(b-a)+\frac{f^{\prime \prime}(a)}{2!}(b-a)^{2}+\cdots+\frac{f^{(n)}(c)}{n!}(b-a)^{n}
$$

Lemma 1 (Banach Lemma) :Let $c \in R^{n \times n}$ with $|c|<1$ then,

$$
\begin{aligned}
& I+C \text { is invertible and } \\
& \frac{1}{1+|C|} \leq\left|(I+C)^{-1}\right| \leq \frac{1}{1-|C|}
\end{aligned}
$$

## Lipschitz Condition:

$f(x)$ Satisfies Lipschitz condition on an interval I if $\exists m>0$ s.t.

$$
f\left(x_{1}\right)-f\left(x_{2}\right) \leq M\left|x_{1}-x_{2}\right| \quad \forall x_{1}, x_{2} \in I
$$

### 3.1.2 Convergence Analysis for Newton's method of solving equation of one variable

 Theorem 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 $g^{\prime}$ exists on $(a, b)$ and that a constant $0<\mathrm{k}<1$ exists with$$
\left|g^{\prime}(x)\right| \leq k, \quad \text { for all } x \in(a, b)
$$

Then for any $p_{0}$ in $[a, b]$, the sequence defined by

$$
p_{n}=g\left(p_{n-1}\right), \quad n \geq 1
$$

Converges uniquely to a fixed point in $[\mathrm{a}, \mathrm{b}]$

## Newton's convergence of solving equation of one variable

Theorem 3: Let $f \in C^{2}[a, b]$. If $p \in(a, b)$ is such that $f(p)=0$ and $f^{\prime}(p) \neq 0$, then there exists a $\delta>0$ such that Newton's method generates a sequence $\left\{p_{n}\right\}_{n=1}^{\infty}$ converging to p for any initial approximation $p_{0} \in[p-\delta, p+\delta]$.

Proof: The proof is based on analyzing Newton's method as the functional iteration scheme $p_{n}=g\left(p_{n-1}\right), \quad n \geq 1$ with

$$
g(x)=x-\frac{f(x)}{f^{\prime}(x)}
$$

Let $k \in(0,1)$. First find an interval $[p-\delta, p+\delta]$ that g maps into itself and for which $\left|g^{\prime}(x)\right| \leq k$, for all $x \in(p-\delta, p+\delta)$.

Since $f^{\prime}$ is continuous and $f^{\prime}(p) \neq 0$, it then implies there exists a $\delta_{1}>0$, such that $f^{\prime}(x) \neq 0$ for $x \in\left[p-\delta_{1}, p+\delta_{1}\right] \subseteq$. $\left.\mathrm{a}, \mathrm{b}\right]$. Thus g is defined and continuous on $[p-$ $\left.\delta_{1}, p+\delta_{1}\right]$. Also

$$
g^{\prime}(x)=1-\frac{f^{\prime}(x) f^{\prime}(x)-f(x) f^{\prime \prime}(x)}{\left[f^{\prime}(x)\right]^{2}}=\frac{f(x) f^{\prime \prime}(x)}{\left[f^{\prime}(x)\right]^{2}}
$$

For $x \in\left[p-\delta_{1}, p+\delta_{1}\right]$, and, since $f \in C^{2}[a, b]$, then $g^{\prime} \in C^{1}\left[p-\delta_{1}, p+\delta_{1}\right]$.

By assumption, $f(p)=0$, so

$$
g^{\prime}(p)=\frac{f(p) f^{\prime \prime}(p)}{\left[f^{\prime}(p)\right]^{2}}=0
$$

Since $g^{\prime}$ is continuous on $k \in(0,1)$, then there exists a $\delta$, such that $0<\delta<\delta_{1}$, and

$$
\left|g^{\prime}(x)\right| \leq k, \quad \text { for all } x \in[p-\delta, p+\delta]
$$

It remains to show that g maps $[p-\delta, p+\delta]$ into $[p-\delta, p+\delta]$. If $x \in[p-\delta, p+\delta]$, the Mean Value Theorem implies that for some number $\xi$ between x and $\mathrm{p},|g(x)-g(p)|=$ $\left|g^{\prime}(\xi)\right||x-p|$. So

$$
|g(x)-p|=|g(x)-g(p)|=\left|g^{\prime}(\xi)\right||x-p| \leq k|x-p|<|x-p|
$$

Now since, $x \in[p-\delta, p+\delta]$, it follows that $|x-p|<\delta$, and that $|g(x)-p|<\delta$. Hence $g$ maps $[p-\delta, p+\delta]$ into $[p-\delta, p+\delta]$.

All the hypothesis of the fixed point Theorem 2 are now satisfied, so the sequence $\left\{p_{n}\right\}_{n=1}^{\infty}$, defined by

$$
p_{n}=g\left(p_{n-1}\right)=p_{n-1}-\frac{f\left(p_{n-1}\right)}{f^{\prime}\left(p_{n-1}\right)}, \text { for } n \geq 1
$$

Converges to p for any $p_{0} \in[p-\delta, p+\delta]$.

Theorem 4: Let r be a fixed point of the iteration $x_{n+1}=g\left(x_{n}\right)$ and suppose that $g^{\prime}(r)=0$, but $g^{\prime \prime}(r) \neq 0$. Then the iteration will have quadratic rate of convergence.

## Proof

Using Taylor series expansion about fixed point $r$

$$
g(x)=g(r)+g^{\prime}(r)(x-r)+\frac{g^{\prime \prime}(r)}{2}(x-r)^{2}+\frac{g^{\prime \prime \prime}(\xi)}{6}(x-r)^{3}
$$

Substitute $x_{n}$ for x and $x_{n+1}=g\left(x_{n}\right), g(r)=r$, and $g^{\prime}(r)=0$
$\Rightarrow x_{n+1}=r+\frac{g^{\prime \prime}(r)}{2}\left(x_{n}-r\right)^{2}+\frac{g^{\prime \prime \prime}(\xi)}{6}\left(x_{n}-r\right)^{3}$

Subtract r from both sides and divide through by $\left(x_{n}-r\right)^{2}$

$$
\frac{x_{n+1}-r}{\left(x_{n}-r\right)^{2}}=\frac{g^{\prime \prime}(r)}{2}+\frac{g^{\prime \prime \prime}(\xi)}{6}\left(x_{n}-r\right)
$$

As $\rightarrow \infty$, then

$$
\lim _{n \rightarrow \infty} \frac{\left|x_{n+1}-r\right|}{\left|x_{n}-r\right|^{2}}=\frac{\left|g^{\prime \prime}(r)\right|}{2}
$$

Since $\alpha=2$, this implies the iteration will converge quadratically.

The fixed-point iteration function for Newton's method is given by

$$
\begin{gathered}
g(x)=x-\frac{f(x)}{f^{\prime}(x)} \\
\Rightarrow g^{\prime}(x)=1-\frac{f^{\prime}(x) f^{\prime}(x)-f(x) f^{\prime \prime}(x)}{\left[f^{\prime}(x)\right]^{2}}=\frac{f(x) f^{\prime \prime}(x)}{\left[f^{\prime}(x)\right]^{2}}
\end{gathered}
$$

When evaluated at $\mathrm{r}, g^{\prime}(r)=0$ since $f(r)=0\left(\right.$ as long as $\left.f^{\prime}(r) \neq 0\right)$
$\Rightarrow$ Newton's method will converge quadratically.

## Convergence of Newton's Method of solving systems of Nonlinear Equations

Lemma 2: Let $f: R^{n} \rightarrow R^{n}$ be a continuously differentiable in an open convex set $D \subset R^{n}$. Suppose a constant $\gamma$ exists such that $\left\|f^{\prime}(x)-f^{\prime}(y)\right\| \leq \gamma| | x-y \|$ for all $x, y \in D$. Then $\left\|f(x)-f(y)-f^{\prime}(y)(x-y)\right\| \leq \frac{\gamma}{2}| | x-y \|^{2}$.

## Proof

By the line integration,

$$
f(x)-f(y)=\int_{0}^{1} f^{\prime}(y+t(x-y))(x-y) d t .
$$

So

$$
f(x)-f(y)-f^{\prime}(y)(x-y)=\int_{0}^{1}\left[f^{\prime}(y+t(x-y))-f^{\prime}(y)\right](x-y) d t .
$$

It follows that

$$
\begin{aligned}
\left\|f(x)-f(y)-f^{\prime}(y)(x-y)\right\| & \leq \int_{0}^{1}\left\|f^{\prime}(y+t(x-y))-f^{\prime}(y)\right\|\|x-y\| d t \\
& \leq \int_{0}^{1} \gamma t\|x-y\|^{2} d t=\frac{\gamma}{2}\|x-y\|^{2}
\end{aligned}
$$

Theorem 5: Let $f: R^{n} \rightarrow R^{n}$ be a continuously differentiable in an open convex set $D \subset R^{n}$. Assume that $\exists \xi \in D$ and $\beta, \gamma>0$ s.t
i) $\quad f(\xi)=\mathbf{0}$
ii) $\quad f^{\prime}(\xi)^{-1}$ exists
iii) $\quad\left|\left|f^{\prime}(\xi)^{-1}\right|\right| \leq \beta$ and
iv) $\quad\left|\left|f^{\prime}(x)-f^{\prime}(y)\right|\right| \leq \gamma| | x-y| |$ for $x, y$ in a neighbourhood of $\xi$.

Then $\exists \in>0$ s.t.for all $x_{0} \in N(\xi, \in)$, the sequence $\left\{x_{k}\right\}$ defined by,

$$
x_{k+1}=x_{k}+s_{k} \text { and } f^{\prime}\left(x_{k}\right) x_{k}=-f\left(x_{k}\right)
$$

Is well defined, converges to $\xi$ and satisfies,

$$
\left|\mid x_{k+1}-\xi\|\leq \beta \gamma\| x_{k}-\xi \|^{2} .\right.
$$

## Proof

By continuity of $f^{\prime}$, choose $\in \leq \min \left\{\gamma, \frac{1}{2 \beta \gamma}\right\}$ so that $f^{\prime}(x)$ is nonsingular for all $x \in N(\xi, \epsilon)$.

For $\mathrm{k}=0,\left|\left|x_{0}-\xi\right|\right|<\epsilon$. So,

$$
\begin{gathered}
\left\|f^{\prime}(\xi)^{-1}\left(f^{\prime}\left(x_{0}\right)-f^{\prime}(\xi)\right)| | \leq\right\| f^{\prime}(\xi)^{-1}\left|\left\|\mid f^{\prime}\left(x_{0}\right)-f^{\prime}(\xi)\right\|\right. \\
\leq \beta \gamma \| x_{0}-\xi| | \leq \frac{1}{2}
\end{gathered}
$$

By the Banach Lemma,

Now,

$$
\begin{aligned}
& x_{1}-\xi=x_{0}-\xi-f^{\prime}\left(x_{0}\right)^{-1} f\left(x_{0}\right)=x_{0}-\xi-f^{\prime}\left(x_{0}\right)^{-1}\left(f\left(x_{0}\right)-f(\xi)\right) \\
&=f^{\prime}\left(x_{0}\right)^{-1}\left[f(\xi)-f\left(x_{0}\right)-f^{\prime}\left(x_{0}\right)\left(\xi-x_{0}\right)\right]
\end{aligned}
$$

so,

$$
\begin{aligned}
\left\|x_{1}-\xi\right\| & \leq\left\|f^{\prime}\left(x_{0}\right)^{-1}\right\|\left\|f(\xi)-f\left(x_{0}\right)-f^{\prime}\left(x_{0}\right)\left(\xi-x_{0}\right)\right\| \\
& \leq 2 \beta \frac{\gamma}{2}\left\|\xi-x_{0}\right\|^{2}=\beta \gamma\left\|\xi-x_{0}\right\|^{2}(\text { by Lemma } 1)
\end{aligned}
$$

$$
\leq \beta \gamma \epsilon\left\|x_{0}-\xi\right\| \leq \frac{1}{2}\left\|x_{0}-\xi\right\| \leq \epsilon
$$

The proof is completed by induction.

Note: The above theorem shows that the Newton's method converges quadratically if $f^{\prime}(\xi)$ is nonsingular and if the starting point is very close to $\boldsymbol{\xi}$.

### 3.1.3 Convergence of Broyden's Method

Lemma 3: Let $f: R^{n} \rightarrow R^{n}$ be a continuously differentiable in an open convex set $\subset R^{n}$. Suppose $\exists \gamma$ constant s.t. $\left\|f^{\prime}(x)-f^{\prime}(y)\right\| \leq \gamma| | x-y \|$ for all $x, y \in D$. Then it holds that, for all $\mathrm{x}, \mathrm{y}, \boldsymbol{\xi} \in D,\left\|f(x)-f(y)-f^{\prime}(\xi)(\boldsymbol{u}-\boldsymbol{v})| | \leq \frac{\gamma}{2}(| | \boldsymbol{x}-\boldsymbol{\xi}\|+\| \boldsymbol{y}-\xi| |)\right\| \boldsymbol{x}-\boldsymbol{y} \|$

## Proof

By the line integral,

$$
\begin{aligned}
\| f(x)-f(y)- & f^{\prime}(\xi)(\boldsymbol{u}-\boldsymbol{v})\|=\| \int_{0}^{1}\left[f^{\prime}\left(y+t(x-y)-f^{\prime}(\xi)\right](x-y) d t \|\right. \\
& \leq \gamma\|x-y\| \int_{0}^{1}\|y+t(x-y)-\xi\| d t \\
& \leq \gamma\|x-y\| \int_{0}^{1}\{t| | x-\xi \|+(1-t)| | y-\xi| |\} d t
\end{aligned}
$$

Lemma 4: Let $f: R^{n} \rightarrow R^{n}$ be a continuously differentiable in an open convex set $D \subset R^{n}$. Suppose $\exists \gamma$ constant s.t. $\left\|f^{\prime}(x)-f^{\prime}(y)\right\| \leq \gamma| | x-y \|$ for all $x, y \in D$. Then for $x_{k+1}, x_{k} \in D$, holds that

$$
\left|\mid B_{k+1}-f^{\prime}(\xi)\|\leq\| B_{k}-f^{\prime}(\xi) \|+\frac{\gamma}{2}\left(| | x_{k+1}-\xi| |+\left|\left|x_{k}-\xi\right|\right|\right) .\right.
$$

## Proof

By definition

$$
\begin{gathered}
B_{k+1}-f^{\prime}(\xi)=B_{k}-f^{\prime}(\xi)+\frac{\left(\nabla f_{k}-B_{k} s_{k}\right) s_{k}^{T}}{s_{k}^{T} s_{k}} \\
=B_{k}\left(I-\frac{s_{k} s_{k}^{T}}{s_{k}^{T} s_{k}}\right)-f^{\prime}(\xi)\left(I-\frac{s_{k} s_{k}^{T}}{s_{k}^{T} s_{k}}\right)+\frac{\left(\nabla f_{k}-f^{\prime}(\xi)\right) s_{k} s_{k}^{T}}{s_{k}^{T} s_{k}}
\end{gathered}
$$

Taking norm

$$
\left\|B_{k+1}-f^{\prime}(\xi)\right\| \leq\left\|B_{k}-f^{\prime}(\xi)\right\|\left\|I-\frac{s_{k} s_{k}^{T}}{s_{k}^{T} s_{k}}| |+| | \frac{\left(\nabla f_{k}-f^{\prime}(\xi)\right) s_{k} s_{k}^{T}}{s_{k}^{T} s_{k}}\right\|
$$

But $\left|\left|I-\frac{s_{k} s_{k}^{T}}{s_{k}^{T} s_{k}}\right|\right| \leq 1$

Therefore, the $3^{\text {rd }}$ term is estimated by,
(by the above lemma)

Theorem 6: Let $f: R^{n} \rightarrow R^{n}$ be a continuously differentiable in an open convex set $D \subset R^{n}$. Suppose $\exists \xi \in R^{n}, \beta, \gamma>0$ s.t.
i) $\quad f(\xi)=\mathbf{0}$
ii) $\quad f^{\prime}(\xi)^{-1}$ exists
iii) $\left|\left|f^{\prime}(\xi)^{-1}\right|\right| \leq \beta$ and
iv) $\quad\left|\left|f^{\prime}(x)-f^{\prime}(y) \| \leq \gamma\right|\right| x-y| |$ for $x, y$ in a neighbourhood of $\xi$.

Then $\exists \delta_{1}, \delta_{2}>0$ such that, if $\left|\left|x_{0}-\xi\right|\right|<\delta_{1}$ and $\left|\mid B_{0}-f^{\prime}(\xi) \| \leq \delta_{2}\right.$, then the Broyden’s method is well defined, converges to $\xi$, and satisfies

$$
\left|\left|x_{k+1}-\xi\right|\right| \leq c_{k}| | x_{k}-\xi| |
$$

With $\lim _{k \rightarrow \infty} c_{k}=0$ (superlinear convergence).

## Proof

Choose $\delta_{2} \leq \frac{1}{6 \beta}$ and $\delta_{1} \leq \frac{2 \delta}{5 \gamma}$. Then

$$
\left|\left|f^{\prime}(\xi)^{-1} \beta_{0}-I\right|\right| \leq \beta \delta_{2} \leq \frac{1}{6}
$$

By Banach lemma $B_{0}^{-1}$ exists. So $x_{1}$ can be defined furthermore,

$$
\begin{aligned}
\left\|B_{0}^{-1}\right\| & =\left\|f^{\prime}(\xi)+\left(B_{0}-f^{\prime}(\xi)\right)^{-1}\right\| \\
& \leq \frac{\left\|f^{\prime}(\xi)^{-1}\right\|}{1-\left\|f^{\prime}(\xi)^{-1} \mid\right\| B_{0}-f^{\prime}(\xi) \|} \leq \frac{\beta}{1-\beta \delta_{2}}
\end{aligned}
$$

Thus,

$$
\begin{aligned}
& \quad\left|\mid \epsilon_{1}\|=\| x_{1}-\xi\|=\| x_{0}-B_{0}^{-1}\left(f\left(x_{0}\right)-f(\xi)\right)-\xi \|\right. \\
& =\left\|-B_{0}^{-1}\left[f\left(x_{0}\right)-f(\xi)-B_{0}\left(x_{0}-\xi\right)\right]\right\| \\
& =\| B_{0}^{-1}\left[f\left(x_{0}\right)-f(\xi)-f^{\prime}(\xi)\left(x_{0}-\xi\right)+\left(f^{\prime}(\xi)-B_{0}\right)\left(x_{0}-\xi\right)\right] \mid \\
& \leq \frac{\beta}{1-\beta \delta_{2}}\left[\frac{\gamma}{2}\left\|\epsilon_{0}\right\|^{2}+\delta_{2}\left\|\epsilon_{0}\right\|\right] \leq \frac{\beta}{1-\beta \delta_{2}}\left[\gamma, v o e r 2 \delta_{1}+\delta_{2}\right]| | \epsilon_{0} \| \\
& \leq \frac{\beta}{1-\beta \delta_{2}} \frac{6 \delta_{2}}{5}\left\|\epsilon_{0}\right\| \leq \frac{\frac{1}{6}}{1-\frac{1}{6}} \frac{6}{5}\left\|\epsilon_{0}\right\| \leq \frac{1}{2}| | \epsilon_{0} \|
\end{aligned}
$$

From lemma 3

$$
\begin{aligned}
& \left|\left|B_{1}-f^{\prime}(\xi)\left\|\leq| | B_{0}-f^{\prime}(\xi)\right\|+\frac{\gamma}{2}\left(\left\|x_{1}-\xi\right\|+\| x_{0}-\xi| |\right)\right.\right. \\
\leq & \delta_{2}+\frac{\gamma}{2}\left(\frac{3}{2}| | \epsilon_{0}| |\right) \leq \delta_{2}\left(1+\frac{\gamma}{2} \frac{3}{2} \frac{2}{5 \gamma}\right) \\
= & \left(1+\frac{3}{10}\right) \delta_{2} \leq \frac{3}{2} \delta_{2} .
\end{aligned}
$$

Thus

$$
\left\|f^{\prime}(\xi)^{-1} B_{1}-I\right\| \leq 2 \beta \delta_{2} \leq \frac{1}{3} .
$$

## By Banach Lemma

$B_{1}^{-1}$ exists
$\left\|B_{1}^{-1}\right\| \leq \frac{\left|\left|f^{\prime}(\xi)^{-1}\right|\right|}{1-\| f^{\prime}(\xi)^{-1}| || | B_{1}-f^{\prime}(\xi)| |} \leq \frac{\beta}{1-2 \beta \delta_{2}} \leq \frac{3}{2} \beta$

The following estimation can now be made

Continuing

$$
\begin{aligned}
\left\|B_{2}-f^{\prime}(\xi)\right\| & \leq\left\|B_{1}-f^{\prime}(\xi)\right\|+\frac{\gamma}{2}\left(\left\|\epsilon_{1}\right\|+\left\|\epsilon_{2}\right\|\right) \\
& \leq \frac{13 \delta_{2}}{10}+\frac{\gamma}{2}\left(\frac{3}{2}| | \epsilon_{1} \|\right) \leq \delta_{2}\left(1+\frac{3}{10}+\frac{\gamma}{2} \frac{3}{2} \frac{1}{2} \frac{2}{5 \gamma}\right) \\
& =\delta_{2}\left(1+\frac{3}{10}+\frac{1}{2} \frac{3}{10}\right) \leq \delta_{2}\left(2-\left(\frac{1}{2}\right)^{2}\right) \leq 2 \delta_{2}
\end{aligned}
$$

The proof is complete by mathematical induction.

### 3.1.4 Convergence of Diagonal Broyden-like Method

Theorem 7: Let the following assumptions hold
i) $\quad \mathrm{F}$ is differentiable in an open convex set $\Delta$ in $R^{n}$
ii) $\quad F^{\prime}(x)$ is continuous for all x and there exists $x^{*}$ in $\Delta$ such that $F\left(x^{*}\right)=0$,
iii) There exists constants $t_{1} \leq t_{2}$ such that

$$
t_{1}| | \Phi\left\|^{2} \leq \Phi^{T} F^{\prime}(x) \Phi \leq t_{2}\right\| \Phi \|^{2} \text { for all } x \in \Delta \text { and } \Phi \in R^{n}
$$

iv) $\exists v>0$ such that the Jacobian matrix satisfies the Lipchitz condition, that is

$$
\left\|F^{\prime}(x)-F^{\prime}(y)\right\| \leq v\|x-y\| \quad \text { for all } x, y \in \Delta
$$

Then there exists $K_{B}>0, \delta>0$ and $\delta_{1}>0$ such that if $x_{0} \in B(\delta)$ and the matrix valued function $B(x)$ satisfies $\left\|I-B(x) F^{\prime}\left(x^{*}\right)\right\|=p(x)<\delta_{1}$ for all $x \in B(\delta)$ then the iteration

$$
x_{k+1}=x_{k}-B\left(x_{k}\right) F\left(x_{k}\right)
$$

Converges linearly.

Proof (see Kelly 1995).

Theorem 8: Let $\left\{\boldsymbol{x}_{k}\right\}$ be a sequence generated by $\boldsymbol{x}_{k+1}=\boldsymbol{x}_{k}-Q_{k} F\left(\boldsymbol{x}_{k}\right)$, where $Q_{k}$ is defined by

$$
\begin{equation*}
Q_{k+1}=Q_{k}+\frac{\left(y_{k}^{T} s_{k}-y_{k}^{T} Q_{k} y_{k}\right)}{\operatorname{Tr}\left(A_{k}^{2}\right)} A_{k} \tag{3.1}
\end{equation*}
$$

Where $A_{k}=\operatorname{diag}\left(y_{k}^{(1)^{2}}, y_{k}^{(2)^{2}}, \ldots, y_{k}^{(n)^{2}}\right), \operatorname{Tr}\left(A_{k}^{2}\right)=\sum_{i=1}^{n} y_{k}^{(i)^{4}}$ and $\operatorname{Tr}($.$) is the trace$ operator.

Also let
i) $\quad \mathbf{F}$ is differentiable in an open convex set $\Delta$ in $R^{n}$
ii) $\quad \boldsymbol{F}^{\prime}(\boldsymbol{x})$ is continuous for all $\mathbf{x}$ and there exists $\boldsymbol{x}^{*}$ in $\Delta$ such that $\boldsymbol{F}\left(\boldsymbol{x}^{*}\right)=0$,
iii) There exists constants $t_{1} \leq t_{2}$ such that

$$
t_{1}| | \Phi\left\|^{2} \leq \Phi^{T} \boldsymbol{F}^{\prime}(x) \Phi \leq t_{2}| | \Phi\right\|^{2} \text { for all } \boldsymbol{x} \in \Delta \text { and } \Phi \in R^{n}
$$

iv) $\exists v>0$ such that the Jacobian matrix satisfies the Lipchitz condition, that is

$$
\left|\left|\boldsymbol{F}^{\prime}(x)-\boldsymbol{F}^{\prime}(y)\|\leq v| | x-y \mid\| \quad \text { for all } x, y \in \Delta .\right.\right.
$$

And there exists constants $\theta>0, \delta>0$, and $\lambda>0$, such that if $x_{0} \in \Delta$ and $Q_{0}$ satisfies $\left|\left|I-Q_{0} \boldsymbol{F}^{\prime}\left(x^{*}\right)\right|_{\boldsymbol{F}}<\delta\right.$ for all $x_{0} \in \Delta$ the sequence $\left\{\boldsymbol{x}_{k}\right\}$ converges linearly to $x^{*}$.

## Proof

It is enough to show that the updating formula $Q_{k}$ satisfies $\left|I I-Q_{k} \boldsymbol{F}^{\prime}\left(\boldsymbol{x}^{*}\right)\right|_{\boldsymbol{F}}<\delta_{k}$, for some constant $\delta_{k}>0$ and for all k. Since

$$
\begin{align*}
& B_{k}=\frac{\left(\mathrm{y}_{\mathrm{k}}^{\mathrm{T}} s_{\mathrm{k}}-\mathrm{y}_{\mathrm{k}}^{\mathrm{T}} \mathrm{Q}_{\mathrm{k}} \mathrm{y}_{\mathrm{k}}\right)}{\operatorname{Tr}\left(\mathrm{A}_{\mathrm{k}}^{2}\right)} \mathbf{A}_{\mathrm{k}}, \text { then } \\
& \left\|\left|Q_{k+1}\left\|_{F} \leq| | Q_{k}\right\|_{F}\right.\right. \tag{3.2}
\end{align*}
$$

Without the lost of generality, by assuming $Q_{0}=I$ and for $K=0$, then

$$
\left|\left|Q _ { 1 } \left\|_{F} \leq\left|\left|Q_{0}\left\|_{F}+| | B_{0}\right\|_{F} .\right.\right.\right.\right.\right.
$$

Since $Q_{0}$ is an identity matrix, hence $\left|\mid Q_{0} \|_{F}=\sqrt{n}\right.$. From the equation

$$
\begin{aligned}
B_{k}= & \frac{\left(y_{k}^{\mathrm{T}} \mathrm{~s}_{\mathbf{k}}-\mathrm{y}_{\mathrm{k}}^{\mathrm{T}} \mathrm{Q}_{\mathrm{k}} \mathrm{y}_{\mathrm{k}}\right)}{\operatorname{Tr}\left(\mathrm{A}_{\mathrm{k}}^{2}\right)} \mathbf{A}_{\mathbf{k}} \quad \text { when } \mathrm{k}=0 \\
\left|B_{0}^{(i)}\right| & =\left\lvert\, \frac{\left|\frac{\mathrm{y}_{0}^{\mathrm{T}} \mathrm{~s}_{0}-\mathrm{y}_{0}^{\mathrm{T}} \mathrm{Q}_{0} \mathrm{y}_{0}}{\operatorname{Tr}\left(\mathrm{~A}_{0}^{2}\right)} \mathrm{A}_{0}\right|}{}\right. \\
& \leq \frac{\left|\mathrm{y}_{0}^{\mathrm{T}} \mathrm{~s}_{0}-\mathrm{y}_{0}^{\mathrm{T}} \mathrm{Q}_{0} \mathrm{y}_{0}\right|}{y_{0}^{(\max )^{2}} \sum_{i=1}^{n} y^{(i)^{4}}} y_{0}^{(\max )^{4}} .
\end{aligned}
$$

But $\frac{y_{0}^{(\max )^{4}}}{\sum_{i=1}^{n} y^{(i)^{4}}} \leq 1$, then

$$
\left|B_{0}^{(i)}\right| \leq \frac{\left|\mathrm{y}_{0}^{\mathrm{T}} F^{\prime}\left(\widehat{x_{0}}\right) y_{0}-\mathrm{y}_{0}^{\mathrm{T}} \mathrm{Q}_{0} \mathrm{y}_{0}\right|}{y_{0}^{\text {maxa })^{2}}}
$$

By letting $t=\max \left\{\left|t_{1}\right|,\left|t_{2}\right|\right\}$ then

$$
\left|B_{0}^{(i)}\right| \leq \frac{|\mathrm{t}-1|\left(\mathrm{y}_{0}^{\mathrm{T}} \mathrm{y}_{0}\right)}{y_{0}^{(\max )^{2}}} .
$$

For $i=1,2, \ldots, n$ and $y_{0}^{(i)^{2}} \leq y_{0}^{(\max )^{2}}$, it follows that

$$
\left|B_{0}^{(i)}\right| \leq \frac{|\mathrm{t}-1| n y_{0}^{(\max )^{2}}}{y_{0}^{(\max )^{2}}},
$$

And thus

$$
\left|\left|B_{0}\right|\right|_{F} \leq n^{\frac{3}{2}}|\mathrm{t}-1| .
$$

Letting $\lambda=n^{\frac{3}{2}}|\mathrm{t}-1|$, then

$$
\left|\mid B_{0} \|_{F} \leq \lambda\right.
$$

Substituting in (*) and letting $\theta=\sqrt{n}+\lambda$, it follows that

$$
\left\|Q_{1}\right\|_{F} \leq \theta
$$

Since $Q_{1}=Q_{0}+B_{0}$ and it is assumed that at $k=0,| | I-Q_{0} \boldsymbol{F}^{\prime}\left(x^{*}\right) \|_{\boldsymbol{F}}<\delta$, then

Hence $\left|\left|I-Q_{1} \boldsymbol{F}^{\prime}\left(x^{*}\right)\right|_{\boldsymbol{F}}<\delta+\lambda \Omega=\delta_{1}\right.$, where $\Omega=| | \boldsymbol{F}^{\prime}\left(x^{*}\right) \|_{\boldsymbol{F}}$. Therefore, by induction, $\left|\left|I-Q_{1} \boldsymbol{F}^{\prime}\left(x^{*}\right)\right|_{\boldsymbol{F}}<\delta_{k}\right.$ for all $k$. Hence, from theorem 7, the sequence $\left\{\boldsymbol{x}_{k}\right\}$ generated by Diagonal Broyden-like method converges linearly to $\boldsymbol{x}^{*}$.

### 3.1.5 Convergence of Steepest Descent Method

Algorithm: Let $f: R^{n} \rightarrow R$ be convex and continuously differentiable function. For the convex analysis of Algorithm A assume that f satisfies Lipschitz condition with constant L , i.e. $\exists L>0$ s.t.
$\left|f\left(x_{1}\right)-f\left(x_{2}\right)\right| \leq L| | x_{1}-x_{2}| |$
$\forall x_{1}, x_{2} \in R^{n}$. Assume the set $T^{*} \neq \phi\left(T^{*}\right.$ set of minimizers of f$), f^{*}$ denote the minimum value of $f$ on $R^{n}$.

The general form of the algorithms is given as:
$x^{0} \in R^{n}$
$\boldsymbol{x}^{k+1}=\boldsymbol{x}^{k}-\alpha_{k} \nabla \boldsymbol{F}\left(\boldsymbol{x}^{k}\right)$
Where $\alpha_{k}>0$ is chosen according to one of the following Algorithms,

## Algorithm A: given L,

Let $\delta_{1}, \delta_{2}$ be positive $\exists$

$$
\begin{equation*}
\frac{L}{2} \delta_{1}+\delta_{2}<1 \tag{3.6}
\end{equation*}
$$

Take $\alpha_{k}$ satisfying,

$$
\begin{equation*}
\delta_{1} \leq \alpha_{k} \leq \frac{2}{L}\left(1-\delta_{2}\right) \tag{3.7}
\end{equation*}
$$

Algorithm B: Let $\Psi: R_{+} \rightarrow R\left(R_{+}\right.$nongetive real line) $\exists$ :

B1) $\Psi$ is convex and continuously differentiable.
B2) $\Psi(0)=0$ and $\Psi(0)<1$
B3) $\lim _{u \rightarrow 0^{+}} \inf \frac{\Psi(u)}{u^{2}}>0$.

Note: $\Psi$ is non decreasing.

For $\delta_{1}>0$ and $\delta_{2}>0$, defining $t_{j}($ for $\mathrm{j}=0,1,2, \ldots)$ as

$$
\begin{equation*}
\delta_{1}<t_{0}<\delta_{2} \tag{3.8}
\end{equation*}
$$

And if $f\left(\boldsymbol{x}^{k}-t_{j} \nabla f\left(x^{k}\right)\right) \leq f\left(\boldsymbol{x}^{k}\right)-\Psi\left(t_{j}\right)| | \nabla f\left(\boldsymbol{x}^{k}\right)| |^{2}$
Then,

$$
\begin{aligned}
& \alpha_{k}=t_{j} \text { and the iterations terminates. Otherwise, } \\
& t_{j+1}=\frac{t_{j}}{2}
\end{aligned}
$$

Definition 4: A sequence $\left\{\boldsymbol{y}^{k}\right\}$ is quasi Fejer convergent to a set $U \subseteq R^{n}$ if $\forall u \in$ $U \exists\left\{\epsilon_{k}\right\} \subseteq R$ s.t. $\epsilon_{k} \geq 0, \sum \epsilon_{k}<\infty$ and $\left\|y^{k+1}-u\right\|^{2} \leq\left|\left|y^{k}-u\right|\right|^{2}+\epsilon_{k} \quad \forall k$.

Theorem 9: If $\left\{\boldsymbol{y}^{k}\right\}$ is quasi Fejer convergent to a set $U \subseteq R^{n}(u \neq \phi)$,then $\left\{y^{k}\right\}$ is bounded. If furthermore a limit point y of $\left\{\boldsymbol{y}^{k}\right\}$ is in $U$, then $\lim _{k \rightarrow \infty} \boldsymbol{y}^{k}=\boldsymbol{y}$.

## Proof

Let $u \in U$. Apply definition 1 .

$$
\begin{aligned}
& \quad\left\|\boldsymbol{y}^{k}-u\right\|^{2} \leq\left\|\boldsymbol{y}^{0}-u\right\|^{2}+\sum_{j=0}^{k-1} \epsilon_{j} \leq\left\|\boldsymbol{y}^{0}-u\right\|^{2}+\sum_{j=0}^{\infty} \epsilon_{j} . \\
& \Rightarrow\left\{\boldsymbol{y}^{k}\right\} \text { is bounded. }
\end{aligned}
$$

Let $y \in U$ be a limit point of $\left\{\boldsymbol{y}^{k}\right\}$ and $\delta>0$. Let $\left\{\boldsymbol{y}^{l k}\right\}$ be a subsequence of $\left\{\boldsymbol{y}^{k}\right\}$ which converges to y .

Using definition $4, \exists k_{0}$ s.t. $\sum_{k_{0}}^{\infty} \epsilon_{j}<\frac{\delta}{2}$, and $\exists k_{1} \geq k_{0}$ s.t.

$$
\left|\left|\boldsymbol{y}^{l k}-\boldsymbol{y}\right|\right|^{2}<\frac{\delta}{2} \quad \forall k_{1} \geq k_{0}
$$

$\Rightarrow \forall k_{1} \geq k_{0}$

$$
\left\|\boldsymbol{y}^{k}-\boldsymbol{y}\right\|^{2} \leq\left|\left|\boldsymbol{y}^{l k}-\boldsymbol{y}\right|\right|^{2}+\sum_{j=l k}^{\infty} \epsilon_{j}<\frac{\delta}{2}+\frac{\delta}{2}
$$

$\Rightarrow \lim _{k \rightarrow \infty} \boldsymbol{y}^{k}=\boldsymbol{y}$

Theorem 10: Let $\boldsymbol{F}: R^{n} \times R \rightarrow R$ s.t.
i) $\quad \exists\left(\boldsymbol{x}_{0}, u_{0}\right) \in R^{n} \times R$ s.t. $\boldsymbol{F}\left(\boldsymbol{x}_{0}, u_{0}\right)=0$
ii) $\quad \mathbf{F}$ is continuous in the neighborhood of $\left(\boldsymbol{x}_{0}, u_{0}\right)$.
iii) $\quad \mathbf{F}$ is differentiable with respect to the variable u in $\left(\boldsymbol{x}_{0}, u_{0}\right)$ and $\frac{\partial \boldsymbol{F}}{\partial u}\left(\boldsymbol{x}_{0}, u_{0}\right) \neq 0$

This implies, there exist a neighborhood $V\left(x_{0}\right)$ and atleast 1 function $u: V\left(x_{0}\right) \rightarrow R$
s.t. $u\left(x_{0}\right)=u_{0}$ and $\boldsymbol{F}(x, u(x))=0 \quad \forall x \in V\left(x_{0}\right)$

If furthermore,
iv) $\frac{\partial F}{\partial u}$ is continuous at $\left(\boldsymbol{x}_{0}, u_{0}\right)$.

Then only the function u satisfies (8) and is continuous at $\boldsymbol{x}_{0}$.

Let $G=\left\{\boldsymbol{x} \in R^{n} / \nabla \boldsymbol{F}(\boldsymbol{x}) \neq 0\right\}$. By continuous differentiability of $\mathbf{F}, G$ is open.

Proposition 1: Let $\Psi$ satisfy B1, B2, and B3. Then
i) $\forall x \in G \exists!u(x)>0$ s.t.

$$
\begin{equation*}
f(x-u(x) \nabla f(x))=f(x)-\Psi(\mathrm{u}(\mathrm{x})) \mid\|\nabla f(x)\|^{2} \tag{3.11}
\end{equation*}
$$

And

$$
\begin{equation*}
f(x-u \nabla f(x)) \leq f(x)-\Psi(\mathrm{u})| | \nabla f(x) \|^{2} \text { iff } 0 \leq u \leq u(x) \tag{3.12}
\end{equation*}
$$

ii) u: $G \rightarrow R_{+} \quad$ is continuous in $G$

Proof
i) For any fixed $x \in G, u \in R_{+} \quad$ by defining

$$
\begin{equation*}
\boldsymbol{F}(\boldsymbol{x}, u)=f(x-u \nabla f(x))-f(x)+\Psi(u)\|\nabla f(x)\|^{2} \tag{3.13}
\end{equation*}
$$

By B1, and B2 F(x,.) is convex and continuously differentiable, also

$$
\begin{align*}
& \boldsymbol{F}(\boldsymbol{x}, 0)=0  \tag{3.14}\\
& \frac{\partial F(x, 0)}{\partial u}=\|\nabla f(x)\|^{2}\left(\Psi^{\prime}(0)-1\right)<0 \tag{3.15}
\end{align*}
$$

And

$$
\begin{equation*}
\boldsymbol{F}(\boldsymbol{x}, 0) \geq f^{*}-f(x)+\Psi(\mathrm{u}) \mid\|\nabla f(x)\|^{2} \tag{3.16}
\end{equation*}
$$

From 12) and 13)
$\mathrm{F}(\mathbf{x},$.$) is negative in some interval to the right of 0$, also from (14), B1 and B2.

$$
\lim _{u \rightarrow \infty} \boldsymbol{F}(\boldsymbol{x}, u)=+\infty
$$

It follows, that $\exists u(x)>0$ s.t.

$$
\boldsymbol{F}(\boldsymbol{x}-u(\boldsymbol{x}))=0
$$

and (9) holds from the uniqueness of $\mathrm{u}(\mathrm{x})$.

Above was from the complexity of $\mathbf{F}(\mathrm{x},$.$) , and the fact that a convex function of real variable$ can take a given value different from its minimum, at most at 2 different points, while

$$
\boldsymbol{F}(\boldsymbol{x}, 0)=\boldsymbol{F}(\boldsymbol{x}-u(\boldsymbol{x}))=0
$$

And 0 is not the minimum value of $\mathbf{F}(\mathbf{x},$.$) by 12$, and 13 .

Hence the proof of 1 .
ii) Let $u_{0}=u\left(\boldsymbol{x}_{0}\right)$ given by (i), given $\boldsymbol{x}_{0}$ in G. Then,

$$
\boldsymbol{F}\left(\boldsymbol{x}_{0}, u_{0}\right)=0
$$

$\boldsymbol{F}(.,$.$) is continuous in a neighborhood of \left(\boldsymbol{x}_{0}, u_{0}\right)$ and also,

$$
\begin{equation*}
\frac{\partial \boldsymbol{F}\left(x_{0}, u_{0}\right)}{\partial u}=-\nabla f\left(x_{0}, u_{0} \nabla f\left(x_{0}\right)\right)^{t} \nabla f\left(x_{0}\right)+\Psi^{\prime}\left(u_{0}\right)\|\nabla f(x)\|^{2} \tag{3.17}
\end{equation*}
$$

As $\mathbf{F}\left(\boldsymbol{x}_{0},.\right)$ is strictly increasing at $u_{0}$, then

$$
\frac{\partial \boldsymbol{F}\left(x_{0}, u_{0}\right)}{\partial u}>0 .
$$

From (15)
$\frac{\partial \boldsymbol{F}}{\partial u}(\ldots)$ is continuous at $\left(\boldsymbol{x}_{0}, u_{0}\right)$ and all the hypothesis of theorem (10) holds, $\Rightarrow \mathrm{u}$ is continuous at $x_{0}$.

Proposition 2: Let $T=\left\{z \in R^{n} / f(z) \leq \lim _{k \rightarrow \infty} \inf f\left(x^{k}\right)\right\}$ then for any $z \in T\left|\left|x^{k+1}-z\right|\right|^{2} \leq\left|\left|x^{k}-z\right|\right|^{2}+\| x^{k+1}-\left.x\right|^{2}$, Where $\left\{x^{k}\right\}$ is generated by (2) and (3) with any $\alpha_{k}>0$.

## Proof

Let $z \in T$. Then

$$
\begin{aligned}
& \left|\left|x^{k+1}-z\right|\right|^{2}-\left|\left|x^{k}-z\right|\right|^{2}-\left|\left|x^{k+1}-x\right|\right|^{2} \\
& =-2\left(z-x^{k}\right)^{t}\left(x^{k+1}-x^{k}\right)=2 \alpha_{k}\left(z-x^{k}\right)^{t} \nabla f\left(x^{k}\right) \\
& \leq 2 \alpha_{k}\left(f(z)-f\left(x^{k}\right) \leq 0 .\right.
\end{aligned}
$$

Using (3) in the second equality, the gradient inequality in the first inequality, and definition of T. The proof follows.

## Analysis of Backtracking procedure

Proposition 3: The backtracking procedure of Algorithm B defined by (6)-(7) stops after a finite number of iterations.

$$
\begin{equation*}
\min \left\{\delta_{1}, \frac{u\left(x^{k}\right)}{2}\right\} \leq \alpha_{k} \leq \min \left\{\delta_{2}, u\left(\boldsymbol{x}^{k}\right)\right\} \tag{3.18}
\end{equation*}
$$

## Proof

Consider 2 cases of $t_{0}$

1) $t_{0} \in\left(0, u\left(x^{k}\right)\right)$
2) $t_{0} \geq u\left(x^{k}\right)$.

Case 1) By Proposition 1 (i), $\alpha_{k}=t_{0}$ from (6) and (7), and iterations stops at $\mathrm{j}=0$.
(16) is then established since $t_{0}<\delta_{2}$ and $t_{0}<u\left(\boldsymbol{x}^{k}\right)$.
$\Rightarrow \alpha_{k}=t_{0}<\min \left\{\delta_{2}, u\left(\boldsymbol{x}^{k}\right)\right\}$ and $t_{0}>\delta_{1}$

Therefore,

$$
t_{0}=\alpha_{k} \geq \min \left\{\delta_{1}, \frac{u\left(x^{k}\right)}{2}\right\}
$$

Case (2) $\exists s \in N, s \geq 1$ s.t.

$$
\begin{equation*}
2^{s-1} u\left(\boldsymbol{x}^{k}\right)<t_{0} \leq 2^{s} u\left(\boldsymbol{x}^{k}\right) \tag{3.19}
\end{equation*}
$$

Then,

$$
\begin{equation*}
\frac{u\left(x^{k}\right)}{2}<\frac{t_{0}}{2^{s}} \leq u\left(\boldsymbol{x}^{k}\right) \tag{3.20}
\end{equation*}
$$

By (6) and (7)

$$
\begin{align*}
& t_{j}=\frac{t_{0}}{2^{j}}, \text { therefore, from (18), } \\
& \frac{u\left(x^{k}\right)}{2}<t_{s} \leq u\left(\boldsymbol{x}^{k}\right) \tag{3.21}
\end{align*}
$$

Claim: $\alpha_{k}=t_{s}$.

From (17) and (18) $t_{s-1}>u\left(\boldsymbol{x}^{k}\right)$ and $t_{s} \leq u\left(\boldsymbol{x}^{k}\right)$,

Using Proposition (1), (7) is satisfied by $\alpha_{k}=t_{s}$.

Therefore (16) follows from (19) and the fact that $t_{s} \leq t_{0}<\delta_{2}$.

Proposition 4: From Algorithm A and B, it holds that:
i) $\exists \delta>0$ s.t.
$f\left(x^{k+1}\right) \leq f\left(x^{k}\right)-\gamma| | x^{k+1}-x^{k}| |^{2} \quad \forall k$
ii) $\quad\left\{f\left(\boldsymbol{x}^{k+1}\right)\right\}$ is decreasing and convergent.
iii) $\quad \sum_{k=0}^{\infty}| | x^{k+1}-x^{k}| |^{2}<\infty$

## Proof

For Algorithm A, using Newton-Leibniz formula:

$$
\begin{aligned}
f\left(\boldsymbol{x}^{k+1}\right) & =f\left(\boldsymbol{x}^{k}\right)-\alpha_{k}| | \nabla f\left(\boldsymbol{x}^{k}\right)| |^{2}+\alpha_{k} \int_{0}^{1}\left(\nabla f\left(\boldsymbol{x}^{k}-u \alpha_{k} \nabla f\left(\boldsymbol{x}^{k}\right)-\nabla f\left(\boldsymbol{x}^{k}\right)\right)^{t} \nabla f\left(\boldsymbol{x}^{k}\right) d u\right. \\
& \leq f\left(x^{k}\right)-\alpha_{k}| | \nabla f\left(\boldsymbol{x}^{k}\right)| |^{2}+L \alpha_{k}^{2}| | \nabla f\left(\boldsymbol{x}^{k}\right)| |^{2} \int_{0}^{1} u d u \\
& \leq f\left(\boldsymbol{x}^{k}\right)-\alpha_{k}\left(1-\frac{L \alpha_{k}}{2}\right)| | \nabla f\left(\boldsymbol{x}^{k}\right)| |^{2}=f\left(\boldsymbol{x}^{k}\right)-\frac{1}{\alpha_{k}}\left(1-\frac{L \alpha_{k}}{2}\right)| | \boldsymbol{x}^{k+1}-\boldsymbol{x}^{k}| |^{2}
\end{aligned}
$$

using

$$
\begin{aligned}
& \delta_{1} \leq \alpha_{k} \leq \frac{2}{L}\left(1-\delta_{2}\right) \\
& \alpha_{k}^{-1}\left(1-\frac{L \alpha_{k}}{2}\right) \geq \frac{\delta_{2} L}{2\left(1-\delta_{2}\right)}
\end{aligned}
$$

$\Rightarrow(20)$ is established for $\gamma=\frac{\delta_{2} L}{2\left(1-\delta_{2}\right)}$.

For Algorithm B,

$$
f\left(\boldsymbol{x}^{k+1}\right) \leq f\left(x^{k}\right)-\Psi\left(\alpha_{k}\right)| | \nabla f\left(\boldsymbol{x}^{k}\right)| |^{2}
$$

Then,

$$
\begin{equation*}
\frac{\Psi\left(\alpha_{k}\right)}{\alpha_{k}^{2}}\left|\left|\boldsymbol{x}^{k+1}-\boldsymbol{x}^{k}\right|\right|^{2}=\frac{\Psi\left(\alpha_{k}\right)}{\alpha_{k}^{2}} \alpha_{k}^{2}| | \nabla f\left(\boldsymbol{x}^{k}\right)| |^{2} \leq f\left(\boldsymbol{x}^{k+1}\right)-f\left(\boldsymbol{x}^{k}\right) \tag{3.23}
\end{equation*}
$$

Taking

$$
0<\xi<\lim _{u \rightarrow 0^{+}} \inf \frac{\Psi(u)}{u^{2}}, \text { and using B3. }
$$

By definition of $\xi, \exists \Theta>0$ s.t.if $\alpha \in(0,1)$ then

$$
\begin{equation*}
\frac{\Psi(u)}{\alpha^{2}}>\xi \tag{3.24}
\end{equation*}
$$

For each k , there are 2 possibilities
a) $\alpha_{k} \in(0, \theta)$, so $\Psi\left(\alpha_{k}\right) \alpha_{k}^{2}>\xi$ by (22)
b) $\alpha_{k} \geq \theta$, by proposition (3), $\alpha_{k} \leq \min \left\{\delta_{2}, u\left(\boldsymbol{x}^{k}\right)\right\} \leq \delta_{2}$,

Therefore, from B1 and B2 $\Psi$ is increasing

$$
\begin{aligned}
& \Rightarrow \Psi\left(\alpha_{k}\right) \geq \Psi(\theta) \\
& \Rightarrow \frac{\Psi\left(\alpha_{k}\right)}{\alpha_{k}^{2}} \geq \frac{\Psi(\theta)}{\delta_{2}^{2}}
\end{aligned}
$$

Taking $\gamma=\min \left\{\xi, \frac{\psi(\theta)}{\delta_{2}^{2}}\right\}$ and using (21) to establish (20) for algorithms B.
It can be seen that (ii) follows from (i), using $\gamma>0$.

To prove iii), by (i) $\exists \gamma>0$ s.t.

$$
\sum_{k=0}^{r}\left\|x^{k+1}-s^{k}\right\|^{2} \leq \frac{1}{\gamma}\left(f\left(x^{0}\right)-f\left(x^{r}\right) \leq \frac{1}{\delta}\left(f\left(x_{0}\right)-f^{*}\right)\right.
$$

Letting r $\rightarrow \infty$

$$
\sum_{k=0}^{\infty}\left\|x^{k+1}-x^{k}\right\|^{2}<\infty
$$

Proposition 5: The sequence $\left\{\boldsymbol{x}^{k}\right\}$ generated by (2) and (3) is convergent to a point $\boldsymbol{x}^{*} \in T$.

## Proof

By Proposition 2 and 4 iii) $\left\{x^{k}\right\}$ is quasi-Fejer convergent to T, with

$$
\epsilon_{k}=\left\|x^{k+1}-x^{k}\right\|^{2}
$$

It remains now to show that, there is a limit of $\left\{\boldsymbol{x}^{k}\right\}$ in T. Using Theorem 9 It can be seen that $\left\{\boldsymbol{x}^{k}\right\}$ is bounded, so it has limit. And by using proposition 4 ii) any cluster point is in T .

Theorem 11: The sequence $\left\{\boldsymbol{x}^{k}\right\}$ generated by (2) and (3) converges to a minimizer of f . Proof: By proposition (5),

$$
\lim _{k \rightarrow \infty} \boldsymbol{x}^{k}=\boldsymbol{x}^{*} \in T
$$

Therefore, it is enough to prove $\boldsymbol{x}^{*} \in T^{*}\left(T^{*}\right.$ set of minimizers of f$)$

For Algorithm A,

$$
\left\|x^{k+1}-x^{k}\right\|^{2}=\alpha_{k}^{2}| | \nabla f\left(x^{k}\right)| |^{2} \geq \delta_{1}^{2}| | \nabla f\left(x^{k}\right)| |^{2} \text { by (5) }
$$

Then
$\nabla f\left(\boldsymbol{x}^{*}\right)=0$ by proposition 4iii) and continuous by $\nabla f($.$) , so \boldsymbol{x}^{*}$ is a minimizer of f by convexity.

For Algorithm B,

Suppose $\boldsymbol{x}^{*} \in T^{*}$, then by convexity of $\mathrm{f}, \boldsymbol{x}^{*} \in G$ and $\left\|\nabla f\left(\boldsymbol{x}^{*}\right)\right\|>0$.

By proposition $1, \mathrm{u}\left(\boldsymbol{x}^{*}\right)>0$ and $\mathrm{u}\left(\boldsymbol{x}^{k}\right)$ converges to $\mathrm{u}\left(\boldsymbol{x}^{*}\right)$. this implies $\exists k_{0}$ s.t. $\quad \forall k \geq k_{0}$

$$
\begin{equation*}
\mathrm{u}\left(x^{k}\right) \geq \frac{\mathrm{u}\left(x^{*}\right)}{2} \quad \text { and }\left|\left|\nabla f\left(x^{k}\right)\right|\right|^{2} \geq \frac{1}{2}| | \nabla f\left(x^{*}\right) \|^{2} \tag{3.25}
\end{equation*}
$$

let $\omega=\left(\min \left\{\delta_{1}, \frac{\mathrm{u}\left(x^{*}\right)}{2}\right\}\right)^{2} \frac{\left.| | \nabla f\left(x^{*}\right)\right|^{2}}{2}$. Then $\forall k \geq k_{0}$

$$
\begin{align*}
\left\|x^{k+1}-x^{k}\right\|^{2} & =\alpha_{k}^{2}| | \nabla f\left(x^{k}\right)| |^{2} \geq\left(\min \left\{\delta_{1}, \frac{\mathrm{u}\left(x^{*}\right)}{2}\right\}\right)| | \nabla f\left(x^{k}\right)| |^{2} \\
& \geq\left(\min \left\{\delta_{1}, \frac{\mathrm{u}\left(x^{*}\right)}{2}\right\}\right)^{2} \frac{| | \nabla f\left(x^{*}\right) \|^{2}}{2}=\omega>0 \tag{3.26}
\end{align*}
$$

Using (3) in the first equality, proposition 3 in the first inequality and (23) in the $2^{\text {nd }}$ one.
Since (24) contradict proposition 4iii), then by contradiction $\boldsymbol{x}^{*} \in T^{*}$ ( $T^{*}$ set of minimizers of f).

### 3.1.6 Convergence of Conjugate Gradient Method

Theorem 12: Let $\left\{x_{k}\right\}$ be the sequence generated by a line search algorithm under the exact line search, or any in exact line search, that

$$
f\left(\boldsymbol{x}_{k}\right)-f\left(\boldsymbol{x}_{k+1}\right) \geq c_{1} c_{2} \frac{\left(-d_{k}^{T} g_{k}\right)^{2}}{\left\|d_{k}\right\|^{2}} \quad c_{1} c_{2}>0 \text { holds }
$$

If

$$
\sum_{k=1}^{\infty} \cos ^{2} \theta_{k}=\infty
$$

Then the sequence is convergent in the sense that

$$
\begin{equation*}
\lim _{k \rightarrow \infty} \inf | | g_{k}| |=0 \tag{3.27}
\end{equation*}
$$

Furthermore,

$$
\text { if } \exists \eta>0 \text { s.t. } \cos ^{2} \Theta_{k} \geq \eta \quad \forall k,
$$

then the sequence is strongly convergent in the sense that $\lim \left|\left|g_{k}\right|\right|=0$.

Theorem 13 (convergence of Conjugate gradient): Let $\left\{\boldsymbol{x}_{k}\right\}$ be the sequence generated by

$$
d_{k+1}=-g_{k+1}+B_{k} d_{k} \quad(\text { conjugate gradient method })
$$

Such that

1) $f\left(\boldsymbol{x}_{k}\right)-f\left(\boldsymbol{x}_{k+1}\right) \geq c_{1} c_{2} \frac{\left(-d_{k}^{T} g_{k}\right)^{2}}{\left\|\mid d_{k}\right\|^{2}} \quad c_{1} c_{2}>0$ holds
2) $\left|d_{k}^{T} \nabla f\left(\boldsymbol{x}_{k}+\alpha_{k} d_{k}\right)\right| \leq-c_{2} d_{k}^{T} \nabla f\left(\boldsymbol{x}_{k}\right)$

If
i) $\quad\left\{f\left(\boldsymbol{x}_{k}\right)\right\}$ is bounded below
ii) $\quad\left\{B_{k}\right\}$ is bounded
iii) $\quad \sum_{k=1}^{\infty} \frac{1}{\left\|d_{k}\right\|^{2}}=\infty$ holds

The method converges in the sense that (1) holds.

Proof

Suppose that (1) is not true, $\exists c_{7}>0 \quad$ s.t.

$$
\left|\left|g_{k}\right|\right| \geq c_{7} \quad \forall k
$$

It follows from (2) that,

$$
\frac{d_{k+1}^{T} g_{k+1}}{\left\|g_{k+1}\right\|^{2}}=-1+\beta_{k+1} \frac{d_{k}^{T} g_{k}}{\left\|g_{k+1}\right\|^{2}}
$$

which gives that,

$$
\begin{aligned}
& 1 \leq \frac{-d_{k+1}^{T} g_{k+1}}{\left\|g_{k+1}\right\|^{2}}+\left|\beta_{k+1}\right| \frac{\left|d_{k}^{T} g_{k}\right|}{\left\|g_{k+1}\right\|^{2}} \\
\leq & \frac{-d_{k+1}^{T} g_{k+1}}{\left\|g_{k+1}\right\|^{2}}+c_{2}\left|\beta_{k+1}\right| \frac{| | g_{k} \|^{2}}{\left\|g_{k+1}\right\|^{2}} \frac{\left|d_{k g_{k}}^{T}\right|}{\left\|g_{k}\right\|^{2}} \\
\leq & \sqrt{1+c_{2}^{2}\left|\beta_{k+1}\right|^{2}| | g_{k}\left\|^{2}| | g_{k+1}\right\|^{-2}} \sqrt{\frac{\left(d_{k+1}^{T} g_{k+1}\right)^{2}}{\left\|g_{k+1}\right\|^{4}}+\frac{\left(d_{k}^{T} g_{k}\right)^{2}}{\left\|g_{k}\right\|^{4}}}
\end{aligned}
$$

From the above inequality and from the assumption made,
$\Rightarrow c_{5}>0$ s.t.

$$
\begin{equation*}
\frac{\left(d_{k}^{T} g_{k}\right)^{2}}{\left\|g_{k}\right\|^{4}}+\frac{\left(d_{k+1}^{T} g_{k+1}\right)^{2}}{\left\|g_{k+1}\right\|^{4}} \geq c_{5} \quad \forall k \text { holds } \tag{3.29}
\end{equation*}
$$

It then follows from theorem (9) and equation (3) that,

$$
\begin{equation*}
\sum_{k=1}^{\infty} \min \left[\frac{1}{| | d_{2 k-1} \|^{2}} \cdot \frac{1}{\left\|d_{k}\right\|^{2}}\right]<\infty, \tag{3.30}
\end{equation*}
$$

which shows that,

$$
\max \left[\left|\left|d_{2 k-1}\right|\right|, \| d_{k}| |\right] \rightarrow \infty
$$

Thus, the definition and the boundedness of $\left\|g_{k}\right\|$ shows that,

$$
\begin{aligned}
& \left\|d_{2 k}| | \leq \frac{1}{2} \max \left[| | d_{2 k-1}| |,\left|\left|d_{k}\right|\right|\right]+\left|\beta_{2 k-1} \|\left|d_{2 k-1}\right|\right| .\right. \\
& \left.\|\left|d_{2 k}\right|\left|\leq\left(2\left|\beta_{2 k-1}\right|+\frac{1}{2}\right)\right|\left|d_{2 k-1}\right| \right\rvert\, .
\end{aligned}
$$

It follows from (4) and the boundedness of $\beta_{k}$ that,

$$
\sum_{k=1}^{\infty}\left[\frac{1}{\left\|d_{2 k}\right\|^{2}}\right]<\infty
$$

Repeating the above analysis with the indices $2 \mathrm{k}-1$ and 2 k replaced by 2 k and $2 \mathrm{k}+1$ respectively. It can be proved that,

$$
\sum_{k=1}^{\infty}\left[\frac{1}{| | d_{2 k+1} \|^{2}}\right]<\infty
$$

Therefore, it follows that,

$$
\sum_{k=1}^{\infty}\left[\frac{1}{| | d_{k}| |^{2}}\right]<\infty
$$

which contradicts the assumption. Hence the theorem is true by contradiction.

### 3.1.7 Convergence of Newton's Method for Optimization

Proposition 6: Suppose that M is a symmetric Matrix. Then the following are equivalent,

1) $h>0$ satisfies $\left\|M^{-1}\right\| \leq \frac{1}{h}$
2) $h>0$ satisfies $\left|\left|M_{v}\right|\right| \geq h . h .||v| \| \quad$ for any vector $v$.

Proposition 1: Suppose that $f(x)$ is twice differentiable. Then,

$$
\nabla f(z)-\nabla f(x)=\int_{0}^{1}[H(x+t(z-x))](z-x) d t
$$

Theorem 14 (Convergence of Newton's method for Optimization): Suppose $f(x)$ is twice differentiable and $x^{*}$ is a point for which $\nabla f\left(\boldsymbol{x}^{*}\right)=0$. Suppose $H(\boldsymbol{x})$ satisfies the following conditions:

1) $\exists h>0$ a scalar for which $\left\|\left[H\left(x^{*}\right)\right]\right\| \leq \frac{1}{h}$
2) $\exists \beta>0$ a scalar and $L>0$ for which $||H(\boldsymbol{x})-H(\boldsymbol{y}) \| \leq L|| \boldsymbol{x}-\boldsymbol{y}| |$

$$
\forall x, y \text { satisfying }\left|\mid \boldsymbol{x}-\boldsymbol{x}^{*} \| \leq \beta \text { and }\right| \mid \boldsymbol{y}-\boldsymbol{x}^{*} \| \leq \beta
$$

Let x satisfy $\left|\left|x-x^{*}\right|\right| \leq \delta \gamma$, where $0<\delta<1$ and $\gamma=\min \left\{\beta, \frac{2 h}{3 L}\right\}$ and let $\boldsymbol{x}_{N}=\boldsymbol{x}-H(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x})$, then,
i) $\quad\left|\left|x_{N}-x^{*}\right|\right| \leq\left|\left|x-x^{*}\right|^{2}\left(\frac{L}{2\left(h-L| | x-x^{*}| |\right)}\right)\right.$
ii) $\quad\left|\left|x_{N}-x^{*}\right|\right| \leq \delta| | x-x^{*}| |$, and hence the iterates converges to $\boldsymbol{x}^{*}$.
iii) $\left\|\boldsymbol{x}_{N}-\boldsymbol{x}^{*}\right\| \leq\left\|\boldsymbol{x}-\boldsymbol{x}^{*}\right\|^{2}\left(\frac{3 L}{2 h}\right)$.

## Proof

$$
\begin{aligned}
\boldsymbol{x}_{N}-\boldsymbol{x}^{*} & =\boldsymbol{x}-H(\boldsymbol{x})^{-1} \nabla f(\boldsymbol{x})-\boldsymbol{x}^{*} \\
& =\boldsymbol{x}-\boldsymbol{x}^{*}+H(\boldsymbol{x})^{-1}\left(\nabla f\left(\boldsymbol{x}^{*}\right)-\nabla f(\boldsymbol{x})\right) \\
& =\boldsymbol{x}-\boldsymbol{x}^{*}+H(\boldsymbol{x})^{-1} \int_{0}^{1}\left[H\left(\boldsymbol{x}+t\left(\boldsymbol{x}-x^{*}\right)\right)\right]\left(x-\boldsymbol{x}^{*}\right) d t \text { from proposition (2) }
\end{aligned}
$$

$$
=H(\boldsymbol{x})^{-1} \int_{0}^{1}\left[H\left(\boldsymbol{x}+t\left(\boldsymbol{x}-\boldsymbol{x}^{*}\right)-H(\boldsymbol{x})\right]\left(\boldsymbol{x}-\boldsymbol{x}^{*}\right) d t\right.
$$

Therefore,

$$
\begin{align*}
\left\|\boldsymbol{x}_{N}-\boldsymbol{x}^{*}\right\| & \leq\left\|H(\boldsymbol{x})^{-1}\right\| \int_{0}^{1} \|\left[H\left(\boldsymbol{x}+t\left(\boldsymbol{x}-\boldsymbol{x}^{*}\right)-H(\boldsymbol{x})\right]\| \| \boldsymbol{x}-\boldsymbol{x}^{*} \| d t\right. \\
& \leq\left\|\boldsymbol{x}-\boldsymbol{x}^{*}\left|\left\|\left|H(\boldsymbol{x})^{-1}\right| \mid \int_{0}^{1} L t\right\| \boldsymbol{x}-\boldsymbol{x}^{*} \| d t\right.\right. \\
& \leq\left\|\boldsymbol{x}-\boldsymbol{x}^{*}\right\|^{2} \| H(\boldsymbol{x})^{-1}| | L \int_{0}^{1} t d t \\
& =\frac{\left\|\boldsymbol{x}-\boldsymbol{x}^{*}\right\|^{2} \| H(\boldsymbol{x})^{-1}| | L}{2} \tag{3.31}
\end{align*}
$$

To bound $\left\|H(x)^{-1}\right\|$. Let v be any vector. Then,

$$
\|H(\boldsymbol{x}) v\|=\left\|H\left(\boldsymbol{x}^{*}\right) v+\left(H(\boldsymbol{x})-H\left(\boldsymbol{x}^{*}\right) v\right)\right\|
$$

$$
\geq \mid\left\|H\left(\boldsymbol{x}^{*}\right) v\right\|-\left\|\left(H(\boldsymbol{x})-H\left(\boldsymbol{x}^{*}\right) v\right)\right\|
$$

$$
\geq h .\left||v|\|-\| H(\boldsymbol{x})-H\left(\boldsymbol{x}^{*}\right)\right| \||v| \mid
$$

$$
\geq h .||v||-L\left|\left\|x-x^{*}| ||v|\right\|\right.
$$

$$
=\left(h-L| | \boldsymbol{x}-\boldsymbol{x}^{*}| |\right)| | v| |
$$

Using proposition (1) again,

$$
\Rightarrow \quad\left\|H(\boldsymbol{x})^{-1}\right\| \leq \frac{1}{h-L| | x-x^{*}| |}
$$

Substituting in $\left(^{*}\right)$ gives,

$$
\begin{equation*}
\left\|x_{N}-x^{*}\right\| \leq\left\|x-x^{*}\right\|^{2} \frac{L}{2\left(h-L| | x-x^{*} \mid\right)} \tag{3.32}
\end{equation*}
$$

Since $L\left|\left|x-x^{*}\right|\right| \leq \delta \cdot \frac{2 h}{3}<\frac{2 h}{3} \quad$ then,

Finally,

And hence (i), (ii), and (iii) proves the theorem.

### 3.1.8 Convergence of Quasi-Newton's Method for Optimization

Wolfe's Condition: This is a popular inexact line search condition which demands that $\alpha_{k}$ should give sufficient decrease in the objective function $f$, as measured by,

$$
\begin{equation*}
f\left(\boldsymbol{x}_{k}+\alpha d_{k}\right) \leq f\left(\boldsymbol{x}_{k}\right)+c_{1} \alpha \nabla f_{k}^{T} d_{k} \tag{3.35}
\end{equation*}
$$

Where $c_{1}$ is some fixed constant and $c_{1} \in(0,1)$.
Equation (i) requires that for any value of $\alpha$, the graph of ,

$$
F(\alpha)=f\left(\boldsymbol{x}_{k}+\alpha d_{k}\right) \text { lies below } f\left(\boldsymbol{x}_{k}\right)+c_{1} \alpha \nabla f_{k}^{T} d_{k}
$$

Using Taylor's theorem,

$$
f\left(\boldsymbol{x}_{k}+\alpha d_{k}\right)=f\left(\boldsymbol{x}_{k}\right)+\alpha \nabla f_{k}^{T} d_{k}+O\left(\alpha^{2}\right)
$$

And since $d_{k}$ is a descent direction i.e. $\nabla f_{k}^{T} d_{k}<0$, such $\alpha$ exists.

Condition (i) is only true for very small $\alpha$. For all $\alpha$, there is need for condition (ii), called Curvature condition

$$
\begin{equation*}
\nabla f\left(\boldsymbol{x}_{k}+\alpha d_{k}\right)^{T} d_{k} \geq c_{2} \nabla f_{k}^{T} d_{k} \tag{3.36}
\end{equation*}
$$

Where $c_{2}$ is some fixed constant and $c_{2} \in\left(c_{1}, 1\right)$.
Condition (ii) implies that, very large $\alpha$ is chosen such that slope of $F(\alpha)$ is larger than $c_{2} F(0)$.

Conditions (i) and (ii) are the Wolfe's Conditions.

Theorem 15 (Dennis and More): Suppose that $F: R^{n} \rightarrow R$ is three times continuously differentiable. Consider the iteration,

$$
\boldsymbol{x}_{k+1}=\boldsymbol{x}_{k}+\alpha_{k} d_{k}
$$

Where $d_{k}$ is a descent direction, and $\alpha_{k}$ satisfies the Wolfe's condition with $c_{1} \geq \frac{1}{2}$. If $\left\{\boldsymbol{x}_{k}\right\}$ converges to $\boldsymbol{x}^{*}$, such that $\nabla f\left(\boldsymbol{x}^{*}\right)=0$ and $\nabla^{2} f\left(\boldsymbol{x}^{*}\right)$ is positive definite, and if $d_{k}$ satisfies

$$
\begin{equation*}
\lim _{k \rightarrow \infty} \frac{\left\|\nabla f_{k}+\nabla^{2} f_{k} d_{k}\right\|}{\left\|d_{k}\right\|}=0 \tag{3.37}
\end{equation*}
$$

Then,
i) The steplength $\alpha_{k}=1$ is admissible $\forall k>k_{0}$, and
ii) If $\alpha_{k}=1 \forall k>k_{0}$, then $\boldsymbol{x}_{k} \rightarrow \boldsymbol{x}^{*}$ superlinearly.

Remark: If $B_{k} d_{k}=-\nabla f_{k}$, then equation (3) is equivalent to,

$$
\begin{equation*}
\lim _{k \rightarrow \infty} \frac{\left\|\left(B_{k}+\nabla^{2} f_{k}\right) d_{k}\right\|}{\left\|d_{k}\right\|}=0 \tag{3.38}
\end{equation*}
$$

This implies, it suffices that $B_{k}$ become increasingly accurate approximation of $\nabla^{2} f\left(x^{*}\right)$ along the search direction $d_{k}$. Therefore, condition (4) is the necessary and sufficient condition for superlinear convergence of quasi-Newton methods.

Theorem 16: Suppose that $F: R^{n} \rightarrow R$ is three times continuously differentiable. Consider the iteration,

$$
\boldsymbol{x}_{k+1}=\boldsymbol{x}_{k}+\alpha_{k} d_{k}, \quad d_{k}=-B_{k} \nabla f_{k}
$$

Assume,
i) $\quad\left\{\boldsymbol{x}_{k}\right\} \rightarrow \boldsymbol{x}^{*}$
ii) $\quad \nabla f\left(x^{*}\right)=0$
iii) $\quad \nabla^{2} f\left(x^{*}\right)$ is positive definite.

Then $\left\{\boldsymbol{x}_{k}\right\}$ converges superlinearly iff equation (4) holds.

## Proof

Equation (4) is equivalent to

$$
\begin{aligned}
d_{k}-d_{k}^{N} & =\left(\nabla^{2} f_{k}\right)^{-1}\left(\nabla^{2} f_{k} d_{k}+\nabla f_{k}\right) \\
& =\left(\nabla^{2} f_{k}\right)^{-1}\left(\nabla^{2} f_{k}-B_{k}\right) d_{k} \\
& \left.=O\left(| | \nabla^{2} f_{k}--B_{k}\right) d_{k}| |\right) \\
& =o\left(| | d_{k}| |\right) .
\end{aligned}
$$

The fact that $\left\|\left(\nabla^{2} f_{k}\right)^{-1}\right\|$ is bounded sufficiently close to $x^{*}$ is used. While proving the quadratic convergence of the Newton's method, it is shown that

$$
\left|\left|\boldsymbol{x}_{k}+d_{k}^{N}-\boldsymbol{x}^{*}\right|\right|=O\left(| | \boldsymbol{x}_{k}-\boldsymbol{x}^{*}| |\right)
$$

Hence,

Since,

$$
\begin{aligned}
\left|\left|\nabla f_{k}\right|\right|=\| \nabla f_{k}-\nabla f\left(\boldsymbol{x}^{*}\right)| | & =\| \int_{0}^{1} \nabla^{2} f\left(\boldsymbol{x}_{k}+t\left(\boldsymbol{x}^{*}-\boldsymbol{x}\right)\right)\left(\boldsymbol{x}^{*}-\boldsymbol{x}_{k}\right) d t| | \\
& =O\left(\left\|\boldsymbol{x}^{*}-\boldsymbol{x}\right\|\right)
\end{aligned}
$$

And since $\left\|\left(\nabla^{2} f_{k}\right)^{-1}\right\|$ is bounded sufficiently close to $x^{*}$ then,

$$
\left\|d_{k}\right\|=O\left(\left\|x_{k}-x^{*}\right\|\right)
$$

Hence,

$$
\left|\mid x_{k}+d_{k}-x^{*} \| \leq O\left(\left\|x_{k}-x^{*}\right\|\right)\right.
$$

Giving the superlinear convergence.

## CHAPTER 4

## METHODOLOGY

In order to evaluate the performance of excel in handling optimization problems, the step by step algorithm of using excel is applied to two popular benchmark problems. Also the various methods used both for systems of nonlinear equations and optimizations are compared. The comparison of the methods is based on the number of iterations required to reach an acceptable solution and also the amount of storage required.

### 4.1 Bench mark problems

1. For systems of nonlinear equations (R.L. Burden and J.D. Faires 1996)

$$
\begin{align*}
& f_{1}\left(x_{1}, x_{2}, x_{3}\right)=3 x_{1}-\cos \left(x_{2} x_{3}\right)-\frac{1}{2} \\
& f_{2}\left(x_{1}, x_{2}, x_{3}\right)=x_{1}^{2}-81\left(x_{2}-0.1\right)^{2}+\sin x_{3}+1.06 \\
& f_{3}\left(x_{1}, x_{2}, x_{3}\right)=e^{-x_{1} x_{2}}+20 x_{3}+\frac{10 \pi-3}{3} \\
& \quad x^{(0)}=\left(x_{1}, x_{2}, x_{3}\right)^{T}=(0.1,0.1,-0.1)^{T} \tag{4.1}
\end{align*}
$$

2. For Optimization (Rosenbrock's Function)

$$
\begin{gather*}
f\left(x_{1}, x_{2}\right)=100\left(x_{2}-x_{1}^{2}\right)^{2}+\left(1-x_{1}\right)^{2} \\
x^{(0)}=\left(x_{1}, x_{2}\right)^{T}=(0.5,0.5)^{T} \tag{4.2}
\end{gather*}
$$

### 4.2 Step by Step excel solution of (4.1)

The methods used for solving (1) are, Newton's method, Quasi-Newton' method, Diagonal Broyden-like method and Homotopy and Continuation method.

### 4.2.1 Solution using Newton's method

Step 1: Given $\boldsymbol{x}^{(0)}$ compute $\mathbf{F}\left(\boldsymbol{x}^{(0)}\right)=\left[\begin{array}{c}3 x_{1}-\cos \left(x_{2} x_{3}\right)-\frac{1}{2} \\ x_{1}^{2}-81\left(x_{2}-0.1\right)^{2}+\sin x_{3}+1.06 \\ e^{-x_{1} x_{2}}+20 x_{3}+\frac{10 \pi-3}{3}\end{array}\right]$
By substituting the values of $\boldsymbol{x}^{(0)}=(0.1,0.1,-0.1)^{T}$
Step 2: Compute the Jacobian and the Jacobian inverse see table 1,
$\boldsymbol{J}(\boldsymbol{x})=\left[\begin{array}{lll}\frac{\partial f_{1}}{\partial x_{1}} & \frac{\partial f_{1}}{\partial x_{2}} & \frac{\partial f_{1}}{\partial x_{3}} \\ \frac{\partial f_{2}}{\partial x_{1}} & \frac{\partial f_{2}}{\partial x_{2}} & \frac{\partial f_{2}}{\partial x_{3}} \\ \frac{\partial f_{3}}{\partial x_{1}} & \frac{\partial f_{3}}{\partial x_{2}} & \frac{\partial f_{3}}{\partial x_{3}}\end{array}\right]=\left[\begin{array}{ccc}3 & x_{3} \sin \left(x_{2} x_{3}\right) & x_{2} \sin \left(x_{2} x_{3}\right) \\ 2 x_{1} & -162\left(x_{2}+0.1\right) & \cos x_{3} \\ -x_{2} e^{-x_{1} x_{2}} & -x_{1} e^{-x_{1} x_{2}} & 20\end{array}\right]$

For the Jacobian inverse,

1) Highlight the appropriate number of rows and columns (here the matrix is 3 by 3 ), and then
2) Put the following command (= minverse ())
3) Inside the parenthesis highlight the matrix that is to be inverted (here the Jacobian matrix).
4) Press $f_{2}$ button and then press ctrl, shift, and enter (buttons) together.

Step 3: Calculate $\boldsymbol{S}_{\boldsymbol{k}}=-\boldsymbol{J}\left(\boldsymbol{x}^{(0)}\right)^{-1} \boldsymbol{F}\left(\boldsymbol{x}^{(0)}\right)$ as follows,

1) Highlight the required number of rows and columns ( 3 by 3 matrix multiplied by 3 by 1) 3 by 1 .
2) Put the following command (= mmult ())
3) Inside the parenthesis highlight the first matrix (the Jacobian inverse matrix), put a comma and then highlight the second matrix $\left(\boldsymbol{F}\left(\boldsymbol{x}^{(0)}\right)\right.$ matrix) and close the bracket.
4) Press $f_{2}$ button and then press ctrl, shift, and enter (buttons) together.

Step 4: Calculate $\boldsymbol{x}^{(1)}=\boldsymbol{x}^{(0)}+\boldsymbol{S}_{k}$ as follows,

1) Click on a new cell, put equal sign (=)
2) Click on the first cell (of $\boldsymbol{x}^{(0)}$ )
3) Put addition sign (+) and click on the second cell (of $\boldsymbol{S}_{k}$ ) and press enter.
4) Drag the cursor down to obtain the rest of the values.

Step 5: Copy and paste until the stopping criterion is attained.

The stopping criterion here is based on $L_{\infty} \leq 10^{-5}$ and can be obtained using the following command, $\left(=\max \left(\operatorname{abs}\left(x_{1}^{(0)}-x_{1}^{(1)}\right), \operatorname{abs}\left(x_{2}^{(0)}-x_{2}^{(1)}\right), \operatorname{abs}\left(x_{3}^{(0)}-x_{3}^{(1)}\right)\right)\right.$.

Table 2.1: Benchmark 1 solution using Excel (Newton's Method)


The solution at iteration number 4

### 4.2.2 Solution Using Quasi-Newton's Method

Step 1: Given $\boldsymbol{x}^{(0)}$ compute $\mathbf{F}\left(\boldsymbol{x}^{(0)}\right)$ as in the case of Newton's method.

Step 2: Set $\mathbf{y}$ (3 by 1) matrix and $\mathbf{s}$ ( 3 by 1 ) matrix as zeros.
Step 3: Compute $\mathbf{J}\left(\boldsymbol{x}^{(0)}\right), \mathbf{J}\left(\boldsymbol{x}^{(0)}\right)^{-1}$ and $\nabla_{k}=\mathbf{J}\left(\boldsymbol{x}^{(0)}\right)^{-1} \boldsymbol{F}$.
Step 4: For the second iteration compute $\boldsymbol{x}^{(1)}=\boldsymbol{x}^{(0)}+\nabla_{k}$, compute $\mathbf{F}\left(\boldsymbol{x}^{(1)}\right)$ by copy and paste.

Step 5: Compute y by the command $\left(=\mathrm{F}\left(\boldsymbol{x}_{1}^{(1)}\right)\right.$ cell $-\mathrm{F}\left(\boldsymbol{x}_{1}^{(0)}\right)$ cell $)$ and drag down, and compute $\mathbf{s}$ by the command $\left(=\boldsymbol{x}_{1}^{(1)}\right.$ cell $-\boldsymbol{x}_{1}^{(0)}$ cell $)$ and drag down.

Step 6: Set $\mathbf{J}\left(\boldsymbol{x}^{(1)}\right)=0$ s and compute $\mathbf{J}\left(\boldsymbol{x}^{(1)}\right)^{-1} \approx B^{k+1}=B^{k}+\frac{\left(s_{k}-B^{k} y_{k}\right) s_{k}^{T} B^{k}}{s_{k}^{T} B^{k} y_{k}}$ by,

1) Highlight 3 by 3 matrix
2) Enter the following command
(=
$B^{k}+\left(s_{k}-\operatorname{Mmult}\left(B^{k}, y_{k}\right)\right) *$
(Mmult(transpose $\left.\left.\left(s_{k}\right), B^{k}\right)\right) /$ mmult(transpose $\left.\left(s_{k}\right), \operatorname{mmult}\left(B^{k}, y_{k}\right)\right)$ )
3) Press $f_{2}$ button and then press ctrl, shift, and enter (buttons) together.

Step 7: compute $\nabla_{k}$ by copy and paste.

Step 8: Copy and paste until the stopping criterion is attained.

Table 2.2: Benchmark 1 solution using Excel Quasi-Newton's method (Broyden's method)


The Solution at iteration number 6.

### 4.2.3 Solution using Diagonal Broyden-like method

Step 1: Given $\boldsymbol{x}^{(0)}$ compute $\mathbf{F}\left(\boldsymbol{x}^{(0)}\right)$.

Step 2: Set y (3 by 1) matrix, s (3 by 1) matrix and A ( 3 by 3 ) matrix as zeros.
Step 3: Compute $\mathbf{J}\left(\boldsymbol{x}^{(0)}\right), \mathbf{J}\left(\boldsymbol{x}^{(0)}\right)^{-1}$ and $\nabla_{k}=\mathbf{J}\left(\boldsymbol{x}^{(0)}\right)^{-1} \boldsymbol{F}$.
Step 4: For the second iteration compute $\boldsymbol{x}^{(1)} \boldsymbol{x}^{(0)}+\nabla_{k}$, compute $\mathrm{F}\left(\boldsymbol{x}^{(1)}\right)$ by copy and paste.

Step 5: Compute y by the command $\left(=\mathrm{F}\left(\boldsymbol{x}_{1}^{(1)}\right)\right.$ cell $-\mathrm{F}\left(\boldsymbol{x}_{1}^{(0)}\right)$ cell $)$ and drag down, and compute $s$ by the command $\left(=x_{1}^{(1)}\right.$ cell $-\boldsymbol{x}_{1}^{(0)}$ cell $)$ and drag down.

Step 6: A is a diagonal matrix:
$A_{11}=y_{1}^{1} * y_{1}^{1}, A_{22}=y_{2}^{1} * y_{2}^{1}, A_{33}=y_{3}^{1} * y_{3}^{1}$.
Step 7: Set $\mathrm{J}\left(\boldsymbol{x}^{(1)}\right)=0$ s and compute $\mathrm{J}\left(\boldsymbol{x}^{(1)}\right)^{-1}=Q_{k+1}=Q_{k}+\frac{\left(y_{k}^{T} s_{k}-y_{k}^{T} Q_{k} y_{k}\right) A_{k}}{\operatorname{Tr}\left(A^{2}\right)}$ by,

1) Highlight 3 by 3 matrix
2) Enter the following command
$\left(=Q_{k}+\left(\left(\right.\right.\right.$ mmult $\left(\right.$ transpose $\left.\left(y_{k}\right), s_{k}\right)-\operatorname{mmult}\left(\operatorname{transpose}\left(y_{k}\right), \operatorname{mmult}\left(Q_{k}, y_{k}\right)\right) /$ $\left.\left.\left.\left(\operatorname{sum}\left(A_{11}^{2}, A_{22}^{2}, A_{33}^{2}\right)\right)\right)\right) * A_{k}\right)$
3) Press $f_{2}$ button and then press ctrl, shift, and enter (buttons) together.

Step 8: compute $\nabla_{k}$ by copy and paste

Step 9: Copy and paste until the stopping criterion is attained.

Table 2.3: Benchmark 1 solution using Excel (Diagonal Broyden-like Method)


The Solution at iteration number 6

### 4.2.4 Solution using Homotopy and Continuation method

Step 1: Given $\boldsymbol{x}^{(0)}$ compute $\mathbf{F}\left(\boldsymbol{x}^{(0)}\right)$ as in the case of Newton's method and also set $\mathrm{h}=0.25$.

Step 2: Compute $\mathbf{J}\left(\boldsymbol{x}^{(0)}\right)$, and $\mathbf{J}\left(\boldsymbol{x}^{(0)}\right)^{-1}$ as in the Newton's method and then use it to calculate $\left(\boldsymbol{K}_{\mathbf{1}}=\boldsymbol{h}\left[-\boldsymbol{J}\left(\boldsymbol{x}^{(0)}\right)\right]^{-\mathbf{1}} \boldsymbol{F}\left(\boldsymbol{x}^{(0)}\right)\right)$ by:

1) Highlight 1 by 3 matrix.
2) Insert the following command ( $=\mathrm{h}^{*} \operatorname{Mmult}\left(-\mathbf{J}\left(\boldsymbol{x}^{(0)}\right)^{-1}, \mathbf{F}\left(\boldsymbol{x}^{(0)}\right)\right)$
3) Press $f_{2}$ button and then press ctrl, shift, and enter (buttons) together.

Step 3: Compute $\boldsymbol{x}^{(0)}=\boldsymbol{x}^{(0)}+\frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{K}_{\mathbf{1}}$ and then use it to calculate $\left[\boldsymbol{J}\left(\boldsymbol{x}^{(0)}+\frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{K}_{\mathbf{1}}\right)\right]$ and $\left[\boldsymbol{J}\left(\boldsymbol{x}^{(0)}+\frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{K}_{1}\right)\right]^{-\mathbf{1}}$ as in the Newton's method.

Step 4: Compute $\mathrm{K}_{2}$ the same way as $\mathrm{K}_{1}$.

Step 5: Compute $\boldsymbol{x}^{(0)}=\boldsymbol{x}^{(0)}+\frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{K}_{2}$ and then use it to calculate $\left[\boldsymbol{J}\left(\boldsymbol{x}^{(0)}+\frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{K}_{2}\right)\right]$ and $\left[\boldsymbol{J}\left(\boldsymbol{x}^{(0)}+\frac{\mathbf{1}}{\mathbf{2}} \boldsymbol{K}_{\mathbf{2}}\right)\right]^{\mathbf{- 1}}$ as in the Newton's method.

Step 6: Compute K 3 the same way as $\mathrm{K}_{2}$.
Step 7: Compute $\boldsymbol{x}^{(0)}=\boldsymbol{x}^{(0)}+\boldsymbol{K}_{\mathbf{3}}$ and then use it to calculate $\left[\boldsymbol{J}\left(\boldsymbol{x}^{(0)}+\boldsymbol{K}_{3}\right)\right]$ and $\left[\boldsymbol{J}\left(\boldsymbol{x}^{(0)}+\boldsymbol{K}_{3}\right)\right]^{-\mathbf{1}}$ as in the Newton's method.

Step 8: Compute $\mathrm{K}_{4}$ the same way as $\mathrm{K}_{3}$.
Step 9: Calculate $\left(\boldsymbol{x}^{(1)}=\boldsymbol{x}^{(0)}+\frac{1}{6}\left(K_{1}+2 K_{2}+2 K_{3}+K_{4}\right)\right)$
Step 10: Copy and paste until the stopping criterion is attained.

Table 2.4: Benchmark 1 solution using Excel (Homotopy and Continuation Method)








| \% Mi |  | Excel | TCH |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | X |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| A | C | E | G | I | J | K | M | N | 0 | Q | 5 | T U | W | X | Y | AA | AC | AD AE | AG | AH Al | AK | AM | AN AO | AQ | AR AS | AU | AW ${ }^{\text {a }}$ |
| 262 N | x | h | F |  | J(wi) |  |  | J (wi) ${ }^{-1}$ |  | K1 |  | (wj+0.5k1) |  | ilwi $+0.5 \mathrm{k} 1{ }^{\text {d }}$ |  | k2 |  | j [ $w j+0.5 \mathrm{k} 2$ ) |  | i $(\mathrm{w}+0.0 .5 \mathrm{k} 2)^{-1}$ | k3 |  | j $(\mathrm{wj}+\mathrm{k} 3$ ) |  | $(\text { (wi } 1+3)^{-1}$ | k4 | $L_{0}$ |
| 26364 | 0.5 | 0.3 | 0 | 3 | 0 | -0 | 0.33 | 4E-06 | 0 | 9E-09 | 3 | $0 \quad 0$ | 0.33 | -3E-06 | 1E-07 | -8E-09 | 3 | $0 \quad$-3E-07 | 0.333 | 4E-06 -2E-07 | 1E-08 | 3 | 0 -1E-07 | 0.33 | 3E-06 -0 | 6E-09 | 6E-05 |
| 264 | 0 | 0.3 | -0.01 | 1 | -16.3 | 0.9 | 0.02 | -0.061 | 0 | -0 |  | 160.87 | -0.02 | 20.061 | -0 | 2E-04 | 1 | -16.3 0.866 | 0.02 | -0.0614 0.0027 | 0 | 1 | -16.3 0.866 | 0.02 | -0.062 0.003 | -2E-04 |  |
| 265 | -0.5 | 0.3 | 0 | - 0 | -0.5 | 20 | 0 | -0.002 | 0.05 | -0 | - 0 | -0.5 20 | 0 | 0.002 | 0.05 | 4E-06 | 0 | -0.5 20 |  | -0.0015 0.00501 | 0 | 0 | -0.5 20 | 0 | -0.002 0.05 | -4E-06 |  |
| 266 N | , | - | F |  | J(wi) |  |  | J (wi) ${ }^{-1}$ |  | K1 |  | wj+0.5k1) |  | i(wi +0.5 k 1 ] |  | k2 |  | j ${ }^{\text {(wi }}$ +0.5k2) |  | i(wi $+0.5 \mathrm{k} 2)^{-1}$ | k3 |  | j(wj+k3) |  | $(1 / 2+3)^{-1}$ | k4 | , |
| 26765 | 0.5 | 0.3 | 0 | 3 | 0 | -0 | 0.33 | 3E-06 | 0 | 7E-09 |  | $0-0$ | 0.33 | -3E-06 | 1E-07 | -7E-09 | 3 | 0 -2E-07 | 0.333 | 4E-06 -2E-07 | $8 \mathrm{E}-09$ | 3 | 0 -1E-07 | 0.33 | 2E-06 | 5E-09 | 5E-05 |
| 268 | 0 | 0.3 | -0.01 | 1 | -16.3 | 0.9 | 0.02 | -0.061 | 0 | -0 |  | 160.87 | -0.02 | 0.061 | 0 | 1E-04 | 1 | -16.3 0.866 | 0.02 | -0.0614 0.0027 | -0 | 1 | 16.30 .866 | 0.02 | -0.062 0.003 | -1E-04 |  |
| 269 | -0.5 | 0.3 | 0 | -0 | -0.5 | 20 | 0 | -0.002 | 0.05 | -0 |  | -0.5 20 | -0 | 0.002 | 0.05 | 4E-06 | 0 | -0.5 20 | 5E-04 | $-0.00150 .0501$ | 0 | - 0 | -0.5 20 | 0 | -0.002 0.05 | -4E-06 |  |
| 0 N | x | h | F |  | I wi ] |  |  | 1 wid $^{-1}$ |  | K1 |  | wi+0.5k1) |  | ilwi $+0.5 \mathrm{k} 1 \mathrm{l}^{1}$ |  | k2 |  | ilwi +0.5 k 2 l |  | ilwi $+0.5 \mathrm{k} 2)^{-1}$ | k3 |  | ilwi+k3) |  | (widk 3$)^{-1}$ | k4 | L |
| 27166 | 0.5 | 0.3 | 0 | 3 | 0 | -0 | 0.33 | 3E-06 | 0 | 6E-09 |  | $0-0$ | 0.33 | -3E-06 | 1E-07 | -6E-09 | 3 | -2E-07 | 0.333 | 3E-06 $-18-07$ | 7E-09 | 3 | -8E-08 | 0.33 | 2E-06 -0 | 4E-09 | 5E-05 |
| 272 | 0 | 0.3 | -0.01 | 1 | -16.3 | 0.9 | 0.02 | -0.061 | 0 | -0 |  | 160.87 | -0.02 | 20.061 | -0 | 1E-04 | 1 | -16.3 0.866 | 0.02 | -0.0614 0.0027 | -0 | 1 | -16.3 0.866 | 0.02 | $-0.0620 .003$ | -1E-04 |  |
| 273 | -0.5 | 0.3 | 0 | -0 | -0.5 | 20 | 0 | -0.002 | 0.05 | -0 | -0 | -0.5 20 | 0 | 0.002 | 0.05 | 3E-06 | - | -0.5 20 | 5E-04 | $-0.00150 .0501$ | 0 | - | -0.5 20 | 0 | -0.002 0.05 | -3E-06 |  |
| 274 N | x | h | F |  | J(wi) |  |  | J (wi) ${ }^{-1}$ |  | K1 |  | (wj+0.5k1) |  | i(wi +0.5 k 1 ] |  | k2 |  | j (wijo.5k2) |  | i $(\mathrm{w}+1+0.5 \mathrm{k} 2)^{-1}$ | k3 |  | j $\mathbf{W} j+1+3$ ) |  | $(\mathrm{w} i+\mathrm{k})^{-1}$ | k4 | $L_{0}$ |
| 27567 | 0.5 | 0.3 | 0 | 3 | 0 | -0 | 0.33 | 3E-06 | 0 | 5E-09 |  | $0-0$ | 0.33 | -2E-06 | 1E-07 | -5E-09 | 3 | -2E-07 | 0.333 | 3E-06 -1E-07 | 6E-09 | 3 | -7E-08 | 0.33 | 2E-06 | 4E-09 | 4E-05 |
| 276 | 0 | 0.3 | -0.01 | 1 | -16.3 | 0.9 | 0.02 | -0.062 | 0 | -0 |  | 160.87 | -0.02 | 20.061 | - 0 | 1E-04 | 1 | -16.3 0.866 | 0.02 | $-0.06150 .0027$ | -0 | 1 | -16.3 0.866 | 0.02 | -0.062 0.003 | -1E-04 |  |
| 277 | -0.5 | 0.3 | 0 | - 0 | -0.5 | 20 | 0 | -0.002 | 0.05 | -0 |  | -0.5 20 | -0 | 0.002 | 0.05 | 3E-06 | 0 | -0.5 20 | 5E-04 | $-0.00150 .0501$ | 0 | - | -0.5 20 | 0 | -0.002 0.05 | -3E-06 |  |
| 278 N | x | h | F |  | J(wi) |  |  | $\mathrm{J}(\mathrm{wi})^{-1}$ |  | K1 |  | wj+0.5k1) |  | i[wi +0.5 k 1 ] |  | k2 |  | j $\left.{ }^{\text {j }} \mathrm{j}+0.5 \mathrm{k} 2\right)$ |  | i $(\mathrm{w}+0.50 .52)^{-1}$ | k3 |  | j $(\mathrm{wj}+\mathrm{k} 3)$ |  | $(\text { (wi+k } 3)^{-1}$ | k4 | $L_{0}$ |
| 27968 | 0.5 | 0.3 | 0 | 3 | 0 | - 0 | 0.33 | 3E-06 | -0 | 4E-09 |  | $0-0$ | 0.33 | -2E-06 | 1E-07 | -4E-09 | 3 | -1E-07 | 0.333 | 3E-06 -1E-07 | 5E-09 | 3 | -6E-08 | 0.33 | 2E-06 | 3E-09 | 4E-05 |
| 280 | 0 | 0.3 | -0.01 | 1 | -16.3 | 0.9 | 0.02 | -0.062 | 0 | -0 |  | 160.87 | -0.02 | 20.061 | -0 | 1E-04 | 1 | -16.3 0.866 | 0.021 | $-0.06150 .0027$ | -0 | 1 | -16.3 0.866 | 0.02 | -0.062 0.003 | -1E-04 |  |
| 281 | -0.5 | 0.3 | 0 | - 0 | -0.5 | 20 | 0 | -0.002 | 0.05 | -0 |  | -0.5 20 | -0 | 0.002 | 0.05 | 3E-06 | - 0 | -0.5 20 | 5E-04 | $-0.00150 .0501$ | 0 | - | -0.5 20 | 0 | -0.002 0.05 | -3E-06 |  |
| 282 N | x | h | F |  | J(wi) |  |  | J (wi) ${ }^{-1}$ |  | K1 |  | wj+0.5k1) |  | i(wi+0.5k1] |  | k2 |  | j $\left.{ }^{\text {j }} \mathrm{j}+0.5 \mathrm{Fk} 2\right)$ |  | i $(\mathrm{w}+0.0 .5 \mathrm{k} 2)^{-1}$ | k3 |  | j $(\mathrm{wj} j+\mathrm{k} 3)$ |  | $(\text { (wi+k3 })^{-1}$ | k4 | L |
| 28369 | 0.5 | 0.3 | 0 | 3 | 0 | -0 | 0.33 | 2E-06 | 0 | 3E-09 |  | 0 -0 | 0.33 | -2E-06 | 9E-08 | -4E-09 | 3 | -1E-07 | 0.333 | 3E-06 | 4E-09 | 3 | -5E-08 | 0.33 | 2E-06 -0 | 3E-09 | 4E-05 |
| 284 | 0 | 0.3 | -0.01 | 1 | -16.3 | 0.9 | 0.02 | -0.062 | 0 | -0 |  | 160.87 | -0.02 | 20.061 | -0 | 1E-04 | 1 | -16.3 0.866 | 0.021 | $-0.06150 .0027$ | -0 | 1 | -16.2 0.866 | 0.02 | -0.062 0.003 | -1E-04 |  |
| 285 | -0.5 | 0.3 | 0 | -0 | -0.5 | 20 | 0 | -0.002 | 0.05 | -0 | -0 | -0.5 20 | -0 | 0.002 | 0.05 | 3E-06 | 0 | -0.5 20 | 5E-04 | -0.0015 0.0501 | 0 | - | -0.5 20 | 0 | -0.002 0.05 | -3E-06 |  |
| 286 N | X | h | F |  | J(wi) |  |  | $\mathrm{J}_{\text {(wil }}{ }^{-1}$ |  | K1 |  | wi+0.5k1) |  | ilwi $+0.5 \mathrm{k} 1{ }^{\text {a }}$ |  | k2 |  | ilwi+0.5k2) |  | i(wi+0.5k2) ${ }^{-1}$ | k3 |  | i(wi+k3) |  | $(\text { (wi+k } 3)^{-1}$ | k4 | $\mathrm{L}_{n}$ |
| 28770 | 0.5 | 0.3 | 0 | 3 | 0 | -0 | 0.33 | 2E-06 | 0 | 3E-09 |  | 0 -0 | 0.33 | -2E-06 | 8E-08 | -3E-09 | 3 | -9E-08 | 0.333 | 2E-06 | 3E-09 | 3 | -4E-08 | 0.33 | 2E-06 -0 | 2E-09 | 3E-05 |
| 288 | 0 | 0.3 | -0.01 | 1 | -16.3 | 0.9 | 0.02 | -0.062 | 0 | -0 |  | 160.87 | -0.02 | 0.061 | 0 | 9E-05 | 1 | -16.3 0.866 | 0.021 | $-0.06160 .0027$ | -0 | 1 | 16.20 .866 | 0.02 | -0.062 0.003 | -9E-05 |  |
| 289 | -0.5 | 0.3 | 0 | - 0 | -0.5 | 20 | 0 | -0.002 | 0.05 | -0 | -0 | -0.5 20 | -0 | 0.002 | 0.05 | 2E-06 | 0 | -0.5 20 | 5E-04 | -0.0015 0.0501 | - 0 | - | -0.5 20 | 0 | -0.002 0.05 | -2E-06 |  |
| 290 N | $x$ | h | F |  | J(wi) |  |  | J (wi) ${ }^{-1}$ |  | K1 |  | (wj+0.5k1) |  | i(wi $+0.5 \mathrm{k} 1{ }^{\text {] }}$ |  | k2 |  | j $\left.{ }^{\text {mj}}+0.5 \mathrm{Fk} 2\right)$ |  | i $(\mathrm{w}+0.5 \mathrm{sk} 2)^{-1}$ | k3 |  | j(wj+k3) |  | $(\text { (wi+k3 })^{-1}$ | k4 | Ls |
| 29171 | 0.5 | 0.3 | 0 | 3 | 0 | -0 | 0.33 | 2E-06 | 0 | 2E-09 |  | 0 -0 | 0.33 | -2E-06 | 7E-08 | -3E-09 | 3 | -8E-08 | 0.333 | 2E-06 -9E-08 | 3E-09 | 3 | -3E-08 | 0.33 | 1E-06 | 2E-09 | 3E-05 |
| 292 | 0 | 0.3 | -0.01 | 1 | -16.3 | 0.9 | 0.02 | -0.062 | 0 | -0 |  | 160.87 | -0.02 | 0.061 | 0 | 9E-05 | 1 | -16.3 0.866 | 0.021 | $-0.06160 .0027$ | 0 | 1 | -16.2 0.866 | 0.02 | -0.062 0.003 | -9E-05 |  |
| 293 | -0.5 | 0.3 | 0 | -0 | -0.5 | 20 | 0 | -0.002 | 0.05 | -0 | -0 | -0.5 20 | 0 | 0.002 | 0.05 | 2E-06 | 0 | -0.5 20 | 5E-04 | $-0.00150 .0501$ | 0 | - | -0.5 20 | 0 | -0.002 0.05 | -2E-06 |  |
| 294 N | x | h | F |  | J(wi) |  |  | J (wi) ${ }^{-1}$ |  | K1 |  | (wj+0.5k1) |  | i[wi +0.5 k 1 ] |  | k2 |  | j (wij+0.5k2) |  | i $(\mathrm{w}+0.50 .52)^{-1}$ | k3 |  | j(wjik3) |  | (wi+k3) ${ }^{-1}$ | k4 | $L_{0}$ |
| 29572 | 0.5 | 0.3 | 0 | 3 | 0 | -0 | 0.33 | 2E-06 | -0 | 2E-09 |  | 0 -0 | 0.33 | -2E-06 | 7E-08 | -2E-09 | 3 | -7E-08 | 0.333 | 2E-06 -9E-08 | 2E-09 | 3 | -3E-08 |  | 1E-06 -0 | 1E-09 | 3E-05 |
| 296 | 0 | 0.3 | -0.01 | 1 | -16.3 | 0.9 | 0.02 | -0.062 | 0 | -0 |  | 160.87 | -0.02 | 0.061 | - 0 | 8E-05 | 1 | -16.3 0.866 | 0.021 | $-0.06160 .0027$ | 0 | 1 | 16.20 .866 | 0.02 | -0.062 0.003 | -8E-05 |  |
| 297 | -0.5 | 0.3 | 0 | -0 | -0.5 | 20 | 0 | -0.002 | 0.05 | -0 | -0 | -0.5 20 | -0 | 0.002 | 0.05 | 2E-06 | 0 | -0.5 20 | 5E-04 | $-0.00150 .0501$ | 0 | - | -0.5 20 | 0 | -0.002 0.05 | -2E-06 |  |



The Solution at iteration number 80.

### 4.3 Step by Step Excel Solution of Rosenbrock's Function

In general, to solve Problem (2) two things are involved, 1) Step length calculation and 2) Descent Direction calculation.

1) Step length Calculation: In general here, the Step length calculation involves expanding the equation, computing the equivalent equation in terms of $\alpha$ assuming $\mathrm{X}^{0}$ and $\mathrm{d}^{0}$ are known, then computing first and second derivative of the equation involving $\alpha$, and then at last applying Newton-Raphson method to solve the equation as follows,

$$
\left.\begin{array}{c}
\boldsymbol{F}(\boldsymbol{x})=100\left(x_{2}-x_{1}^{2}\right)^{2}+\left(1-x_{1}\right)^{2} \quad \text { Rosenbrock's function } \\
=100\left[\left(x_{2}-x_{1}^{2}\right)\left(x_{2}-x_{1}^{2}\right)\right]+\left[\left(1-x_{1}\right)\left(1-x_{1}\right)\right] \\
=100\left[x_{1}^{4}+x_{2}^{2}-2 x_{1}^{2} x_{2}\right]+\left[x_{1}^{2}-2 x_{1}+1\right] \\
\boldsymbol{F}(\boldsymbol{x})=100 x_{1}^{4}+100 x_{2}^{2}-200 x_{1}^{2} x_{2}+x_{1}^{2}-2 x_{1}+1 \text { Expansion } \\
\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}+\alpha d \\
\qquad \phi(\alpha)=f\left[\begin{array}{l}
x_{1}^{0}+\alpha d_{1}^{0} \\
x_{2}^{0}+\alpha d_{2}^{0}
\end{array}\right] \\
\phi(\alpha)=100\left(x_{1}^{0}+\alpha d_{1}^{0}\right)^{4}+100\left(x_{2}^{0}+\alpha d_{2}^{0}\right)^{2}-200\left(x_{1}^{0}+\alpha d_{1}^{0}\right)^{2}\left(x_{2}^{0}+\alpha d_{2}^{0}\right)+ \\
\left(x_{1}^{0}+\alpha d_{1}^{0}\right)^{2}-2\left(x_{1}^{0}+\alpha d_{1}^{0}\right)+1 \\
\quad \text { Equation in terms of } \alpha
\end{array}\right] \begin{aligned}
& g(\alpha)=\frac{d \phi(\alpha)}{d \alpha}=400\left(x_{1}^{0}+\alpha d_{1}^{0}\right)^{3} d_{1}^{0}+200\left(x_{2}^{0}+\alpha d_{2}^{0}\right) d_{2}^{0}-400\left(x_{1}^{0}+\alpha d_{1}^{0}\right) \\
& \left(x_{2}^{0}+\alpha d_{2}^{0}\right) d_{1}^{0}-200\left(x_{1}^{0}+\alpha d_{1}^{0}\right)^{2} d_{2}^{0}+2\left(x_{1}^{0}+\alpha d_{1}^{0}\right) d_{1}^{0}-2 d_{1}^{0}
\end{aligned}
$$

First derivative with respect to $\alpha$

$$
\begin{aligned}
& g^{\prime}(\alpha)=\frac{d^{2} \phi(\alpha)}{d \alpha^{2}} \\
& \quad=1200\left(x_{1}^{0}+\alpha d_{1}^{0}\right)^{2} d_{1}^{02}+200 d_{2}^{02}-400 d_{1}^{02}\left(x_{2}^{0}+\alpha d_{2}^{0}\right) \\
& \\
& -800 d_{1}^{0} d_{2}^{0}\left(x_{1}^{0}+\alpha d_{1}^{0}\right)+2 d_{1}^{02}
\end{aligned}
$$

Second derivative with respect to $\alpha$
The Solution in each iteration depends on the value of $\mathbf{x}$ and d. In each case after computing the first iteration, the rest of the iterations follow by copy and paste.

Given the $\mathbf{x}$ and d values, $\alpha$ is given the value $1, \mathrm{~g}(\alpha)=\frac{d \phi(\alpha)}{d \alpha}$ is calculated and also $g^{\prime}(\alpha)=\frac{d^{2} \phi(\alpha)}{d \alpha^{2}}$ is calculated. The next value of $\alpha$ is calculated by the formula $\alpha_{1}=\alpha_{0}-\frac{\mathrm{g}(\alpha)}{g^{\prime}(\alpha)}$

Table 2.5: Example of the Steplength Calculation Using Excel (Newton - Raphson method)


## 2) Descent Direction calculation

The methods used for Descent direction calculation are, (1) Steepest Descent method, (2) Conjugate gradient method, (3) Modified Newton's method and (4) Quasi-Newton's method (DFP and BFGS).

### 4.3.1 Solution Using Steepest Descent method

Given $\boldsymbol{x}^{(0)}$
Step 1: Compute $\mathbf{F}\left(\boldsymbol{x}^{(0)}\right)$ by typing the following command $\left(=100 * x_{1} \wedge 4+100 * x_{2} \wedge 2-\right.$ $\left.200 * x_{1} \wedge 2 * x_{2}+x_{1} \wedge 2-2 * x_{1}+1\right)$.

Step 2: Compute $\nabla \mathbf{F}(\mathbf{x})=\left[\begin{array}{c}400 x_{1}^{3}-400 x_{1} x_{2}+2 x_{1}-2 \\ 200 x_{2}-200 x_{1}^{2}\end{array}\right]$ by typing the following command, first cell $\left(=400 * x_{1} \wedge 3-400 * x_{1} * x_{2}+2 * x_{1}-2\right)$, and second cell $\left(=200 * x_{2}-200 * x_{1}\right.$ ^2).

Step 3: Compute $d_{k}=-\nabla \mathbf{F}(\mathbf{x})$ as follows,

1) Highlight 1 by 2 matrix
2) Enter the following command (=-(highlight $\nabla \mathbf{F}(\mathbf{x}))$ )
3) Press $f_{2}$ button and then press ctrl, shift, and enter (buttons) together.

Step 4: Compute $\alpha d_{k}, \alpha$ is the Step length which is calculated as we have seen before (Using Newton-Raphson method). Therefore, $\alpha d_{k}$ is calculated by the following command $\left(=\alpha^{*}\left(\right.\right.$ highlight $\left.\left.d_{k}\right)\right)$, press $\mathrm{f}_{2}$ and then ctrl, shift and enter together.

Step 5: Copy and paste until the stopping criterion is attained
Step 6: Stopping criterion $\left(\mathrm{f}(\mathbf{x}) \leq 10^{-5}\right)$.

Table 2.6: Benchmark 2 solution using Excel (Steepest Descent Method)









Until


There are 947 iterations.

Note that, the detail of the step length calculation at each iteration is skipped here for the sake of convinience.

### 4.3.2 Solution Using Conjugate gradient method

Step 1: Given $\boldsymbol{x}^{(0)}$ Calculate $\mathbf{F}\left(\boldsymbol{x}^{(0)}\right), \nabla \mathbf{F}(\mathbf{x})$ and $d_{k}=-\nabla \mathbf{F}(\mathbf{x})$ as in the Steepestdescent method

Step 2: Set $\beta$ and $\boldsymbol{d}^{k-1}$ as zeros.
Step 3: Calculate $\boldsymbol{d}_{k}=-\nabla \mathrm{f}\left(\boldsymbol{x}^{0}\right)+\beta \boldsymbol{d}^{k-1}$ using the following command $\left(=\left(\right.\right.$ highlight $\left.-\nabla \mathbf{F}\left(\boldsymbol{x}^{0}\right)\right)+\beta *\left(\right.$ highlight $\left.\left.\boldsymbol{d}^{k-1}\right)\right)$. press $\mathrm{f}_{2}$ and then ctrl, shift and enter together.

Step 4: Calculate $\alpha \boldsymbol{d}_{\boldsymbol{k}}$ as in the steepest descent method.
Step 5: Copy and paste until the stopping criterion is attained.
Where $\beta=\frac{\left[\nabla f\left(x^{k}\right)\right]^{T} \nabla f\left(x^{k}\right)}{\left[\nabla f\left(x^{k-1}\right)\right]^{T} \nabla f\left(x^{k-1}\right)}$

Table 2.7: Benchmark 2 solution using Excel (Conjugate Gradient Method)


There are 9 iterations
Note that, the detail of the step length calculation is skipped here for convinience

### 4.3.3 Solution Using Modified Newton's method

Given $\boldsymbol{x}^{(0)}$

Step 1: Calculate $\mathbf{F}\left(\boldsymbol{x}^{(0)}\right)$, and $\nabla \mathbf{F}(\mathbf{x})$ as in the Steepest- descent method.
Step 2: Calculate $H=\left[\begin{array}{cc}\frac{\partial^{2} f}{\partial x_{1}^{2}} & \frac{\partial^{2} f}{\partial x_{1} x_{2}} \\ \frac{\partial^{2} f}{\partial x_{2} x_{1}} & \frac{\partial^{2} f}{\partial x_{2}^{2}}\end{array}\right]=\left[\begin{array}{cc}1200 x_{1}^{2}-400 x_{2}+2 & -400 x_{1} \\ -400 x_{1} & 200\end{array}\right]$ by typing the following command, $H_{11}\left(=1200 * x_{1} \wedge 2-400 * x_{2}+2\right), H_{12}\left(=-400 * x_{1}\right)$,

$$
H_{21}\left(=-400 * x_{1}\right), H_{22}(=200) .
$$

Step 3: Calculate H inverse by,

1) Highlight 2 by 2 matrix.
2) Write the command (=(Minverse (highlight H)))
3) Press $f_{2}$ button and then press ctrl, shift, and enter (buttons) together.

Step 4: Calculate $d^{k}=-\left[H\left(\boldsymbol{x}^{(0)}\right)\right]^{-1} \nabla \boldsymbol{F}\left(\boldsymbol{x}^{(0)}\right)$ by,

1) Highlight 1 by 2 matrix.
2) Write the following command $\left(=-\operatorname{Mmult}\left(\left[H\left(\boldsymbol{x}^{(0)}\right)\right]^{-1}, \nabla \boldsymbol{F}\left(\boldsymbol{x}^{(0)}\right)\right)\right)$
3) Press $f_{2}$ button and then press ctrl, shift, and enter (buttons) together.

Step 5: Calculate $\alpha d_{k}$ as in the Steepest-descent method.
Step 6: Copy and paste until the stopping criterion is attained.

Table 2.8: Benchmark 2 solution using Excel (Modified Newton's Method)


There are 7 iterations.

Note that, the detail of the step length calculation is skipped here for convinience

### 4.3.4 Solution Using Quasi-Newton's DFP (Davidon, Fletcher, Powell) method

Given $\boldsymbol{x}^{(0)}$

Step 1: Calculate $\mathbf{F}\left(\boldsymbol{x}^{(0)}\right)$, and $\nabla \mathbf{F}(\mathbf{x})$ as in the Steepest- descent method.

Step 2: Set $\boldsymbol{s}^{k}$ and $\boldsymbol{q}^{k}$ as zeros and $Q^{k}=\left[\begin{array}{ll}1 & 0 \\ 0 & 1\end{array}\right]$
Step 3: Calculate $\boldsymbol{d}^{k}=-Q^{k} \nabla\left(\boldsymbol{x}^{(k)}\right)$ by,

1) Highlight 2 by 1 Matrix
2) Write the following command (= - Mmult $\left.\left(Q^{k}, \nabla\left(\boldsymbol{x}^{(0)}\right)\right)\right)$..
3) Press $f_{2}$ button and then press ctrl, shift, and enter (buttons) together.

Step 4: Calculate $\alpha \boldsymbol{d}_{k}$ as in the Steepest-descent method.

For the second iteration,
Step 5: Compute $\boldsymbol{q}^{k}$ by clicking the command ( $=\nabla \mathrm{F}\left(\boldsymbol{x}_{1}^{(1)}\right)$ cell $-\nabla \mathrm{F}\left(\boldsymbol{x}_{1}^{(0)}\right)$ cell $)$ and drag down, and compute $\boldsymbol{s}^{k}$ by clicking the command ( $=\boldsymbol{x}_{1}^{(1)}$ cell $-\boldsymbol{x}_{1}^{(0)}$ cell ) and drag down.

Step 7: Calculate $Q^{k+1}=Q^{k}+\frac{s^{k}\left(s^{k}\right)^{T}}{\left(q^{k}\right)^{T} s^{k}}-\frac{\left(Q^{k} q^{k}\right)\left(Q^{k} q^{k}\right)^{T}}{\left(q^{k}\right)^{T} Q^{k} q^{k}}$ by,

1) Highlight 2 by 2 matrix
2) Enter the following command
$\left(=Q^{k}+\left(\frac{\operatorname{Mmult}\left(s^{k}, \operatorname{transpose}\left(s^{k}\right)\right)}{\operatorname{Mmult}\left(\operatorname{transpose}\left(q^{k}\right), s^{k}\right)}\right)-\left(\frac{\left(\operatorname{Mmult}\left(Q^{k}, q^{k}\right)\right) *\left(\operatorname{transpose}\left(\operatorname{Mmult}\left(Q^{k}, q^{k}\right)\right)\right)}{\left(\left(\operatorname{Mmult}\left(\operatorname{transpose}\left(q^{k}\right), \operatorname{Mmult}\left(Q^{k}, q^{k}\right)\right)\right)\right.}\right)\right)$
3) Press $f_{2}$ button and then press ctrl, shift, and enter (buttons) together.

Step 7: compute d and $\alpha \mathrm{d}$ by copy and paste.
Step 8: Copy and paste until the stopping criterion is attained.

Table 2.9: Benchmark 2 solution using Excel (DFP Method)


There are 9 iterations.

Note that, the detail of the step length calculation is skipped here for convinience.

### 4.3.5 Solution Using Quasi-Newton's BFGS (Broyden, Fletcher, Goldfarb, and Shanno ) Method

Given $\boldsymbol{x}^{(0)}$

Step 1: Calculate $\mathbf{F}\left(\boldsymbol{x}^{(0)}\right)$, and $\nabla \mathbf{F}(\mathbf{x})$ as in the Steepest descent method.
Step 2: Set $\boldsymbol{s}^{k}$ and $\boldsymbol{q}^{k}$ as zeros and $Q^{k}$ as identity 2 by 2 Matrix.
Step 3: Calculate $\boldsymbol{d}^{k}=-Q^{k} \nabla\left(\boldsymbol{x}^{(k)}\right)$ by,

1) Highlight 2 by 1 Matrix
2) Write the following command ( $\left.=-\operatorname{Mmult}\left(Q^{k}, \nabla\left(\boldsymbol{x}^{(0)}\right)\right)\right)$..
3) Press $f_{2}$ button and then press ctrl, shift, and enter (buttons) together.

Step 4: Calculate $\alpha \boldsymbol{d}_{k}$ as in the Steepest-descent method.

For the second iteration,

Step 5: Compute $\boldsymbol{q}^{k}$ by clicking the command $\left(=\nabla \mathrm{F}\left(\boldsymbol{x}_{1}^{(1)}\right)\right.$ cell $-\nabla \mathrm{F}\left(\boldsymbol{x}_{1}^{(0)}\right)$ cell $)$ and drag down, and compute $\boldsymbol{s}^{k}$ by clicking the command ( $=\boldsymbol{x}_{1}^{1}$ cell $-\boldsymbol{x}_{1}^{0}$ cell ) and drag down.

Step 7: Calculate
$Q^{k+1}=Q^{k}+\left(1+\frac{\left(q^{k}\right)^{T} Q^{k} q^{k}}{\left(q^{k}\right)^{T} s^{k}}\right) \frac{s^{k}\left(s^{k}\right)^{T}}{\left(q^{k}\right)^{T} s^{k}}-\frac{1}{\left(q^{k}\right)^{T} s^{k}}\left[\left[s^{k}\left(q^{k}\right)^{T} Q^{k}\right]^{T}+s^{k}\left(q^{k}\right)^{T} Q^{k}\right]$ by,

1) Highlight 2 by 2 matrix
2) Enter the following

$$
\begin{aligned}
& \operatorname{Command}\left(=Q^{k}+\left(1+\left(\operatorname{Mmult}\left(\operatorname{transpose}\left(q^{k}\right), \operatorname{Mmult}\left(Q^{k}, q^{k}\right)\right) /\right.\right.\right. \\
& \left.\operatorname{Mmult}\left(\operatorname{transpose}\left(q^{k}\right), s^{k}\right)\right)\left(\operatorname { M m u l t } \left(s^{k}, \operatorname{transpose}\left(s^{k}\right) /\right.\right. \\
& \left.\operatorname{Mmult}\left(\operatorname{transpose}\left(q^{k}\right), s^{k}\right)\right)-\left(1 / \operatorname{Mmult}\left(\operatorname{transpose}\left(q^{k}\right), s^{k}\right)\right) *
\end{aligned}
$$

$$
\begin{aligned}
& \quad\left(\operatorname{transpose}\left(\operatorname{Mmult}\left(s^{k}, \operatorname{Mmult}\left(\operatorname{transpose}\left(q^{k}\right), Q^{k}\right)\right)\right)+\right. \\
& \left.\left.\left.\operatorname{Mmult}\left(s^{k}, \operatorname{Mmult}\left(\operatorname{transpose}\left(q^{k}\right), Q^{k}\right)\right)\right)\right)\right)
\end{aligned}
$$

3) Press $f_{2}$ button and then press ctrl, shift, and enter (buttons) together.

Step 7: compute d and $\alpha \mathrm{d}$ by copy and paste.
Step 8: Copy and paste until the stopping criterion is attained.

Table 2.10: Benchmark 2 solution using Excel (BFGS Method)


There are 10 iterations.
Note that, the detail of the step length calculation is skipped here for convinience.

## CHAPTER 5

## CONCLUSION, DISCUSSION AND RECOMMENDATION

In Chapter 3, it was concluded that most of the unconstrained optimization and the nonlinear systems of equations algorithms have well established convergence theory. Building algorithms on the convergence theory described in 3 lead to efficient implementations that behaved predictably according to their theoretical results. Also some of the individual strengths and possible weakness of the reviewed algorithms were identified.

The use of Excel spreadsheet was discussed in chapter 4. It was illustrated that the use of Excel especially as a teaching tool will enhance the students understanding of the algorithms discussed and the way they work. The effect of changing any cell value can be clearly observed on all the cells that are dependent on this cell value. Also error can directly be traced. The availability and the user friendliness of the Excel spreadsheet were shown to be among the advantages of its use.

Evaluating the performance of a certain algorithm is indeed a difficult task. There is definite answer to which algorithm has the best overall performance. It should be emphasized that all test results depend on the choice of the benchmark problems and the choice of tolerance for the stopping criteria. For example Tests on high dimensional or noisy functions were not carried out in this thesis.

The solutions obtained by the methods for nonlinear systems of equations using the benchmark problem, yielded the following results; Newton's method converges at $4^{\text {th }}$ iteration, Quasi-Newton's method converges at $6^{\text {th }}$ iteration, Diagonal Broyden-like method converges at $6^{\text {th }}$ iteration, and Homotopy and Continuation method converges at $80^{\text {th }}$ iteration. The above result implied that Newton method performed better among the methods, whereas Homotopy and Continuation method have the worst performance.

Quasi-Newton's method and Diagonal Broyden-like method have shown good performances given that they do not require the derivative evaluations.

On the other hand for unconstrained optimization with Rosenbrock's function as the benchmark problem, it was shown that; steepest descent method converges at 947th iteration, conjugate gradient method converges at $9^{\text {th }}$ iteration, modified newton method converges at $7^{\text {th }}$ iteration, and both quasi-newton DFP and BFGS methods converges at $10^{\text {th }}$ iteration. Clearly modified newton's method performed better, followed by conjugate gradient method, and then DFP and BFGS method.

Due to the inconvenience derived by attaching the calculation of steplength at each step of unconstrained optimization, it was skipped. It should be noted that the calculation is mandatory and it can be done on the same sheet. The algorithm for the calculation was given in chapter 4.3.

One Excel spreadsheet contains $1,048,576$ rows and 16,384 columns. It is observed that, the memory allocated for the rows is independent of the memory allocated for the columns. It is also observed that, the number of iterations a sheet can take depends on the number of rows used for steplength calculation. Also the number of rows for the steplength calculation depends on the dimension of the objective function and the initial/starting point used. For example when handling an objective function (Rosenbrock's function) of dimension 2, requiring 21 rows for the steplength calculation Excel was able to produce 45481 iterations. Whereas for an objective function (Extended Rosenbrock's function) of dimension 4, requiring 48 rows for the steplength calculation Excel was able to produce only 21,399 iterations.

## REFERENCES

[1] I. Newton (1664-1671), Methodus fluxionum et serierum infinitarum.
[2] J. Raphson (1690), Analysis Aequationum Universalis, London.
[3] H. Qiu (1993), A Robust Examination of the Newton-Raphson Method With Strong Global Convergence Properties, Master's Thesis, University Of Florida.
[4] J.E.Dennis, and Jr., Robert B.Schnabel (1996). Numerical Methods For Unconstrained Optimization and Non-Linear Equations. SIAM.
[5] J. Bunch, and J. Hopcroft (1974), Triangular Factorization by fast matrix multiplication, Math.Comp, 28, pp.231-236.
[6] D. Copper Smith, and S.Winograd (1990), Matrix Multiplication via Arithmetic Progression, J.Symbolic Comput, 9, pp.93-108.
[7] E. Spedicato, and J. Xia (1992), Finding General Solutions of the Quasi-Newton's Equation via The ABS Approach, Optim. Methods Softw, 14, pp.243-251.
[8] E. Spedicato and J. Zhao (1993), Explicit General Solution of the Quasi-Newton's Equation with Sparsity and Symmetry, Optim. Methods Softw, pp.311-319.
[9] E. Spedicato, and J. Greenstadt (1978), On Some Classes Of Variationally Derived Quasi-Newton's Methods for Systems Of Non-Linear Algebraic Equations, Numerische Mathematic, 14, pp.363-380.
[10] C. Broyden (1965), A Class Of Methods For Solving Non-Linear Simultaneous Equations, Maths. Comp, 19, pp.577-593
[11] E. Spedicato (1983), A Class of Rank-One Positive Definite Quasi-Newton Updates for Unconstrained Minimization, Optimization, 14, pp.61-70.
[12] M. Al-Baali (2011), Convergence Analysis Of Family Of Quasi-Newton's Methods For Non-Linear Optimization, Reaserch Report Domas 11 (2), Sultan Qaboos University, Oman.
[13] D.M.Gay (1979), Some Convergence Properties Of Broyden's Method, SIAM J.Numer.Anal, 14, pp.623-630.
[14] D.P.O’Leary (1995), Why Broyden Non-Symmetric Method Terminates On Linear Equations, SIAM J.Optim., 5,pp.231-235.
[15] Z.Ping (1998), A Note On The Termination Of Broyden Method For Linear Systems Of Equations, Journal Of China Agricultural University,14,pp.24-27.
[16] M. Adib, N. Mahdavi-Amiri, and E. Spedicato (1999), Broyden's Methods as an ABS Method, Publications University Of Miskolc, Mathematics, 14, pp.3-13.
[17] E.Feng, Z. Xia, and L.Zhang (1998), ABS Methods For Optimization, Dalian University Press, Chinese.
[18] C. Broyden, J. Dennis and J.J.More (1973), On The Local And Superlinear Convergence Of Quasi-Newton Methods, IMA,J.Appl.Math, 77, pp.223-245.
[19] M.Y.Waziri, W.J.Leong, and M.A.Hassan (2012), Diagonal Broyden-Like Method For Large-Scale Systems Of Non-Linear Equations, M.J.M.S, 6(1),pp.59-73.
[20] M.A. Hassan, W.J.Leong, and M.Farid (2009), A New Gradient Method via QuasiCauchy Relation Which Guarantees Descent , J. Comput. Appl. Math, 230,pp.300-305.
[21] W.J. Leong, and M.A. Hassan (2009), A Restarting Approach For The Symmetric Rank One Update For Unconstrained Optimization, Computational Optimization and Applications, 43,pp.327-334.
[22] W.J. Leong, and M.A. Hassan (2011a), A New Gradient Method via Least Change Secant Update. International Journal of Computer and Mathematics, 88, pp.816-828.
[23] F.Moderres, M.A, and W.J.Leong (2011), Improved Hessian Approximation With Modified Secant Equations For Symmetric Rank-One Method, J.Comput.Appl, 61, pp.33123318.
[24] M.Y.Waziri, W.J.Leong, M.A.Hassan, and M.Monsi (2010a), A New Newton's Method With Diagonal Jacobian Approximation For Systems Of Non-Linear Equations, J. Mathematics and Statistics, 6, pp.246-252.
[25] R.L. Burden, and J.D.Faires (1996), Numerical Analysis $9{ }^{\text {th }}$ Edition, Youngstown State University, Ohio, pp.660-666.
[26] Ya-Xiang Yuan (2006), A New Step Size for the Steepest Descent Method, Journal Of Computational Mathematics, Vol.24, N0.2, pp.149-156.
[27] M.A.Bhati (2000), Practical Optimization Methods With Mathematica Application, Springer Books Archives, Vol.1, pp.262-272.
[28] W.C.Davidon (1959), Variable Metric Method For Minimization (revised), Technical Report ANL-5990, Argonne National Laboratory, USA.
[29] R.Fletcher and M.J.D.Powell (1963), A rapidly Convergent Descent Method For Minimization, Comput.J, 6, pp.163-168.
[30] W.C.Davidon (1991), Variable Metric Method For Minimization, SIAM J.Optim, 1,pp.1-17.
[31] R.Fletcher (1987 reprinted in 2000), Practical Methods of Optimization (2 $2^{\text {nd }}$ Edition), Wiley, Chichester, England.
[32] A.R.Conn, N.I.M. Gould, and Ph.L.Toint (2000), Trust-Region Methods, SIAM Series On Optimization, Philadelphia, USA.
[33] J.Nocedal (1992), Theory of Algorithms for Unconstrained Optimization, Acta Numer.1, pp.199-242.
[34] R.Fletcher (1994), An Overview Of Unconstrained Optimization In Algorithms For Continuous Optimization, The State Of The Art, Editor, Kluwer Academic, Boston, pp.109143.
[35] A.R.Conn, N.I.M.Gould, and Ph.L. Toint (1994), Large Scale Nonlinear Constrained Optimization, A Current Survey In The State Of The Art (E.Spedicato,ed), NATO ASI SERIES C, Mathematical And Physical Sciences, Kluwer, Dordrecht, Netherland 434, pp.287-332.
[36] J.Nocedal (1996), Large Scale Unconstrained Optimization, In The State Of The Art in Numerical Analysis, Oxford University Press, pp.311-338.
[37] L.Luksan, and E. Spedicato (2000), Variable Metric Methods for Unconstrained Optimization and Nonlinear Least Squares, J.Compt. Appl.Math, 124, pp.61-95.
[38] N.I.M. Gould, D. Orban and Ph.L. Toint (2004) Numerical Methods For Large-Scale Nonlinear Optimization, RAL-TR-2004-032, Oxford, UK.
[39] M.J.D.Powell (2010), On Non-Linear Optimization since 1959, In the Birth of Numerical Analysis, Editors Adhemar Bultheel and Ronald Cools.
[40] L.M.G. Drummond, A.N.Iusem, and B.F.Svaiter, Full Convergence Of The Steepest Descent Method with Inexact Line Searches, Instituto de Mathematica Pura e Aplicada Estrada Dona Castorina, 110,Rio de Janeiro, RJ, CEP 22460-320, Brazil.

