A Family of Minimum Curvature Variable-Metric Methods for Unconstrained Optimization

Samir Y. Obeid
Instructor of Physics, at the Lebanese American University

THESIS

Submitted in partial fulfillment of the requirements for the degree of
Master of Science in Computer Science
at the Lebanese American University
May 1998

Dr. Issam Moghrabi (Supervisor)
Assistant Professor of Computer Science
Lebanese American University

Dr. May Abboud
Associate Professor of Mathematics
Lebanese American University

Dr. Samer Haber
Assistant Professor of Mathematics
Lebanese American University

Abstract

The major objective of the work in this thesis concentrates on deriving a Variable-Metric family of minimum curvature multi-step quasi-Newton methods for unconstrained optimization. The aim is to provide further gains in numerical performance to those obtained by the multi-step methods developed earlier, as well as to develop a general framework that encompasses all possible multi-step minimum curvature algorithms generated by appropriate parameter choices.

The derivation will be based on a selected rational model with a free parameter, oriented towards securing a "smooth" interpolation of the points defining the multi-step curve. Different choices of the metric used in parameterizing the curves needed for updating the inverse Hessian approximation at each iteration, are tested and numerical results are then reported.

Acknowledgment

As a faculty member already involved in teaching Physics at the Lebanese American University since a long time, I could not consider myself as a student in the traditional sense since I started working towards a degree in Computer Science. This is evidently, because I am not like any freshman student who normally has applied for the first time to the university after making up his mind to choose a specific major that is going to constitute the foundation for his future career. That is on the contrary for me, because it was really a diversified rare experience mixed with all the feelings of embarrassment, haziness, patience, seriousness, challenge, privilege and renovation.

In fact, it was not at all an easy task to adapt myself to the student-faculty idea, especially when I got started with the few first prerequisite undergraduate courses, and mostly when I realized that some of my students turned out to become my classmates. Then all through the long way and slow motion that I followed, I felt a real conflict of commitment with the different responsabilities that I willingly have decided to carry at various instants of my life. After all, and in similar cases like mine , one should be alert and careful to show at all times the highest degree of seriousness and achievement, by being always ready at the expected level, keeping himself prepared for all kinds of evaluation tests and fulfilling all the duties and obligations.

But, nevertheless so many other factors have contributed to make the overall job as happy and fruitful as it can be. Mainly, when considering the real help and the big support that I felt in the very loving homely atmosphere of the university. Partly it is, because my instructors were fortunately my colleagues in the same department, and therefore I had always the opportunity to share ideas with them, chiefly when working on my thesis. Indeed, acquiring throughout my experience a broader angle of vision and a better analytical thinking, it is by now a time where feeling at maximum ease and confidence, far from any frustration or competition, I could enjoy every moment I spent in this program which was totally new for me. And it is then, when I was convinced that a good educator should learn how to be also a good student by all means.

For all this, I would like first to thank God who made this task possible for me. Then, I acknowledge with gratitude the invaluable advise offered by Dr. Issam Moghrabi who to my pride accepted unhesitantly to supervise this work where I tried to build and add depending on his

previous extensive research in the field of my interest. And at this conclusive step, it is a real time to remember all those who helped me and sustained my work, especially our previous Dean Dr. Raja Hajjar who was the first person to encourage me going into computer studies, and Dr Leila Khoury who backed me firmly all through my way. Of course, I also thank the LAU administration, the Chairman and the faculty members of the Natural Science division especially those involved in the Computer Science program who contributed surely to enrich my teaching experience and my knowledge in their course subjects. I acknowledge here my debt to the two outstanding readers Dr. May Abboud and DR. Samer Haber, and to Mr. Malek Halabi who has developed an object-oriented software in the same domain of research. Lastly, I wish to acknowledge the support of my family and friends, especially that of my wife Laure and my daughter Hiba who showed a very rare patience, sharing with me the whole experience.

Table of contents

CHAPTER ONE

GENERAL INTRODUCTION

1.1 Optimization: Applicability And Background :

The applicability of optimization methods is widespread reaching almost into every activity in which numerical information is processed. The optimization theory, which is interested in finding the "best" way to carry out a given action, has been studied since the beginning of formal mathematics. But it has assumed practical significance only since the development of the digital computer, when it appeared as a fascinating blend of theory and experiment. Since then, improvements in optimization methods have in several cases outpaced even the dramatic gains in computing speed, with the result that entire complex enterprises are guided today by a combination of advanced optimization techniques and high-performance computers. Researchers are still working in this area hoping to get improvements in efficiency and reliability, where they are involved in studying optimality criteria for problems, the determination of algorithmic methods of solution, the study of the structure of such methods, and computer experimentation with methods both under trial conditions and real life problems.

Techniques associated with linear algebra are at the very heart of modern optimization methods, including ordinary and partial differential equations as well as approximation. Indeed in the subject of optimization one cannot ignore the role of theoretical studies, revealing to the computer scientist better ways to implement numerical techniques effectively.

In its early stage (i.e. before 1940), least squares calculations, steepest descent type as well as the Newton method were used. Then (between 1940-1950), the very important subject of "linear programming" was developed, and originally it was aimed to deal with optimal planning. Later on, the subject was revolutionized (in 1959) when W. C. Davidon introduced variable metric methods [9].

1.2 Assistance From Some Program Libraries, And User's Contribution:

It is possible in optimization to arrange things into categories of

standard problems with related well-designed algorithms. So, at any time the user should know what subroutine to use for his case, where a good help may be obtained in choosing the method from some program libraries that give a decision tree in the documentation. And here, for the sake of getting a higher sensitivity in the solution, he may need sometimes to vary some parameters over wide ranges especially if the mathematical model is not a close approximation to reality, or if he cannot build his design to the same accuracy as the solution.

1.3 Defining Our Work:

Our work here falls mainly in the domain of unconstrained optimization, in which the optimum value is sought of an objective function of many variables, without any constraints, but where many of the ideas can carry over into other problems in constrained optimization. Our aim is ultimately here, to develop new mathematical techniques that can be applied to a wide class of problems, suitable for implementation on a computer, and independent of the particular machine used to perform the calculations. But we do not claim perfection of our developed methods. Rather, they suggest new ways of thinking about optimization that can be extended to more advanced and diverse problem spaces.

1.4 Basic Mathematical Tools:

For a better understanding of the detailed mathematical derivations, that are going to appear in the following chapters, one should have a kind of basic knowledge of the concepts and techniques of matrix algebra and numerical linear algebra. Vector spaces, in their turn are going to be used heavily, and in this respect, a point **x** in n-dimensional space (R^n) is the vector $(x_1, x_2, ..., x_n)^T$, x_k being the component in the k_{th} coordinate direction. In our work we are going to recur frequently to iterative methods for generating any sequence of points, (say $\underline{x}_{i+1}$).

Consequently, a line is the set of points $\underline{x}_{i+1} = \underline{x}(t_i) = \underline{x}_i + t_i \underline{p}_i$, where $\underline{x}_i$ is a fixed point along the line (for $t_i = 0$), and $\underline{p}_i$ is the direction of the line which is sometimes convenient to be normalized as $\underline{p}_i^t \underline{p}_i = 1$ modifying in that the value of t_i only.

A subject of importance also is the calculus of any function of many variables, where with each of which is associated some sort of contours. In general, the functions that we are going to consider are "smooth" meaning that for a continuous function $f(\underline{x})$ there exists at any point $\underline{x}$ a gradient vector where : $\underline{g}(\underline{x}) = \nabla f(\underline{x})$.

The Hessian matrix can exist in this case if the considered function is twice continuously differentiable (C^2) and it is denoted by:

$$G(\underline{x}) = \nabla^2 f(\underline{x}) \quad or \quad \nabla \underline{g}(\underline{x}) \quad or \quad \nabla(\nabla f(\underline{x})).$$

From here, it comes out that the slope of $f(\underline{x}(t_i))$ along the line at any point $\underline{x}(t_i)$ is:

$$\frac{df(\underline{x}(t_i))}{dt_i} = \nabla f(\underline{x}(t_i))^T \underline{p}_i = \underline{g}(\underline{x}(t_i))^T \underline{p}_i,$$

and the curvature along the line is: $\frac{d^2 f(\underline{x}(t_i))}{dt_i^2} = \underline{p}_i^T \nabla^2 f(\underline{x}(t_i)) = \underline{p}_i^T G \underline{p}_i$,

which is the scalar product of $\underline{p}_i$ and $G\underline{p}_i$. Here $\pm\frac{\underline{g}(\underline{x}_i)}{\|\underline{g}(\underline{x}_i)\|_2}$ are the derivatives of greatest and least slope at $\underline{x}_i$, over all directions for which the norm $\|\underline{p}_i\|_2 = 1$, and are orthogonal to the contour and tangent plane of $f(\underline{x})$ at $\underline{x}_i$.

The linear function is one of many variable functions that can be written as: $f(\underline{x}) = \sum_{i=1}^{n} a_i x_i + b = \underline{a}^T \underline{x} + b$, where $\underline{a}$ and b are constants.

Whereas, a general quadratic function can be written as:

$$f(\underline{x}) = \frac{1}{2}\underline{x}^T G \underline{x} + b^T \underline{x} + c \quad or \quad \frac{1}{2}(\underline{x} - \underline{x}')^T G(\underline{x} - \underline{x}') + c', \qquad (1.1)$$

in which $$G\underline{x}' = -b, \quad and \quad c' = c - \frac{1}{2}\underline{x}'^T G \underline{x}'. \qquad (1.2)$$

Hence, $$\nabla f(\underline{x}) = \frac{1}{2}(G + G^T)\underline{x} + b = G\underline{x} + b. \qquad (1.3)$$

Here by considering the symmetry of G, and after getting the expression for $\underline{g}(\underline{x})$, one can reach a widely used result that is expressed as: $$\underline{g}(\underline{x}_2) - \underline{g}(\underline{x}_1) = G(\underline{x}_2 - \underline{x}_1) \qquad (1.4)$$

One important technique that is worthwhile mentioning here is the Taylor series that helps in handling more general smooth functions of many variables where:

$$f(\underline{x}_i + t_i \underline{p}_i) = f(\underline{x}_i) + t_i \underline{p}_i^T \nabla f(\underline{x}_i) + \frac{1}{2} t_i^2 \underline{p}_i^T [\nabla^2 f(\underline{x}_i)] \underline{p}_i + \ldots \qquad (1.5)$$

and therefore: $\nabla f(\underline{x}_i + t_i \underline{p}_i) = \nabla f(\underline{x}_i) + [\nabla^2 f(\underline{x}_i)] t_i \underline{p}_i + \ldots$ is a Taylor series expansion of the gradient of f [8].

1.5 The Minimization Problem, And Linear Convergence:

The main problem we shall consider is that of minimizing the objective function $f(\underline{x})$. In this respect, a minimizer might not exist or if exists it may not be unique. Considerable difficulties are met in the way when finding global minima, where one has to search for $f(\underline{x}) > f(\underline{x}^*)$ $(\forall \underline{x})$. Other difficulties emerge in the case of non-smooth functions where non-smooth minima do not satisfy the same condition as smooth minima. However, the existence and the continuity of the first and second derivatives are still to be considered here. And the two conditions that should be satisfied for all $\underline{p}_i$ by a local minimum are summarized by:

$$\underline{p}_i^T \underline{g}(\underline{x}^*) = 0 \text{ and } \underline{p}_i^T G^* \underline{p}_i \geq 0 \qquad (1.6)$$

where the first is referred to as a first order necessary condition, whereas the second states that G^* is a positive-definite matrix.

In fact, sufficient conditions for a strict and isolated local minimizer x* require that $\underline{g}(\underline{x}^*) = 0$ holds and that G^* is positive definite, (i.e. $\underline{p}^T G^* \underline{p} > 0 \quad \forall \underline{p} \neq 0$).

Related to some algorithmic properties, linear convergence can be found starting from the determination of the error $\Delta \underline{x}_k = \underline{x}_k - \underline{x}^*$, which measures how far it is from reaching the local minimum x*, and therefore it is an indication for the convergence (especially when $\Delta \underline{x}_k \rightarrow \mathbf{0}$). In fact, the first order convergence corresponds to $\frac{\|\Delta \underline{x}_{k+1}\|}{\|\Delta \underline{x}_k\|} \leq a$, and the second order convergence to $\frac{\|\Delta \underline{x}_{k+1}\|}{\|\Delta \underline{x}_k\|^2} \leq a$, where if the rate constant a is small enough (i.e. $a \leq \frac{1}{4}$) then linear convergence is only acceptable [9].

But, since the linear convergence does not often guarantee a good performance, that is why experimentation was an outlet in developing an optimization technique. Here, experience tells that a reliable indication of good performance is associated directly with well-chosen experimental testing.

1.6 The Material In The Following Chapters:

Following this introduction, chapter two will be dealing extensively with the Newton's method as well as with some line search algorithms. Then, chapter three will introduce quasi-Newton methods for unconstrained optimization and the Broyden family. The major concern of chapter four will be in developing the so-called multi-step quasi-Newton method as well as the more advanced minimum curvature multi-step quasi-Newton technique. Then, chapter five will be concerned in presenting the "new minimum curvature quasi-Newton method", where it includes a complete derivation of the general equation that opens the door to various special cases that will be discussed separately and tested at a later stage. After this detailed theoretical study, chapter six is going to show the results of the implementation of the "new minimum curvature" two-step quasi-Newton method, as well as of some cases depicted from the general equation derived in chapter five. After all, comparing their behavior with respect to each other as well as with respect to other previously tested algorithms (namely, the original BFGS and the best of the accumulative and fixed-point algorithmic refinements introduced before in the two-step minimum curvature problem). Finally, chapter seven is a conclusion where the results of the new techniques as compared to the previous ones will be evaluated, and some suggestions for improvement and for future research will be included.

CHAPTER TWO

NEWTON'S METHOD

AND SOME LINE SEARCH ALGORITHMS

2.1 Frequently Used Algorithms In Optimization:

A substantial class of optimization methods, is based on adopting a convenient approximation model to the objective function, where quadratic models showed reasonable performance in predicting the location of a local minimizer. In fact, many methods for unconstrained minimization are derived to work properly if applied to a quadratic function with positive definite Hessian G [9]. This is because:

a) quadratic functions are smooth and easy to manipulate, with a well determined minimum
b) methods based on quadratic models are expected to have a rapid rate of convergence; and these methods, to a large extent, can be made invariant under a linear transformation of the variables
c) as related to the Taylor series of $f(\underline{x})$ about an arbitrary point $\underline{x}_i$ taken to quadratic terms, quadratic information is more effective than linear one (like the case of steepest descent) in predicting directions along which substantial progress can be reached.

One case rising from the use of a quadratic model is the Newton-like methods (Newton, quasi-Newton and its derivatives) which approximate the Hessian matrix. And another case where a quadratic model will be used is when a method is derived having the property known as quadratic termination where the minimizer can be located in a known finite number of iterations. This quadratic termination has the prerequisite of the conjugacy of a set of non-zero vectors $\underline{p}_1, \underline{p}_2, \ldots, \underline{p}_n$ to a given positive definite matrix G; i.e.:

$$\underline{p}_i^T G \underline{p}_j = 0 \qquad \forall i \neq j.$$

2.2 Describing Newton's Method:

Newton's method is based on a quadratic model which requires the function $f(\underline{x})$ of our concern to be available at any point, as well as its first and second derivatives. This becomes very clear after looking at

the quadratic model that is obtained from considering the first few terms of the Taylor series expansion of $f(\underline{x})$ about $\underline{x}_i$, which can be written as:

$$f(\underline{x}_i + \underline{p}) \approx h_i(\underline{p}) = f(\underline{x}_i) + \underline{g}(\underline{x}_i)^T \underline{p} + \frac{1}{2}\underline{p}^T G(\underline{x}_i)\underline{p} \qquad (2.1)$$

where $\underline{p} = \underline{x} - \underline{x}_i$, and $h_i(\underline{p})$ is the result of the quadratic approximation obtained in the i_{th} iteration [8]. Accordingly, as a result of this method one gets that: $\underline{x}_{i+1} = \underline{x}_i + t_i \underline{p}_i$.

Here, in the process of searching for a minimizer which is unique by virtue of an expected positive definite G, $\underline{p}_i$ will be defined by satisfying the condition that $\nabla h(\underline{p}_i) = 0$, leading us to describe Newton's method by the i_{th} iteration in a procedural form summarized by:

a)solving the $n \times n$ system of linear equations involved in $G(\underline{x}_i)\underline{p}_i = -\underline{g}(\underline{x}_i)$ for $\underline{p}_i$, then

b)setting $\underline{x}_{i+1} = \underline{x}_i + \underline{p}_i$.

Concerning the convergence of the method, it is useful to know that if f is a function which has a continuous partial derivative of order 3 (i.e. $f \in \mathbb{C}^3$) with $\underline{x}^*$ being a minimum of f, then if it happened that the starting point $\underline{x}_0$ is sufficiently close to $\underline{x}^*$ then convergence of order 2 will hold at each step for Newton's method [9], where:

$$\exists \; c \in \Re > 0 \| \underline{x}_{i+1} - \underline{x}^* \| \leq \| \underline{x}_i - \underline{x}^* \|^2 .$$

Many deficiencies of Newton's method as a practical algorithm for optimization are summarized by the following points:

In many practical applications the Hessian of the objective function may be too costly to evaluate, since it requires $\frac{1}{6}n^3 + O(n^2)$ multiplications per iteration [9].As well as it may even be unavailable in explicit form. Also, when the i_{th} iterate $\underline{x}_i$ is far from the minimizer then the calculated Hessian G may not be positive definite, and accordingly the basic Newton method will not be suitable for a general purpose algorithm. And in this respect, the convergence of the method is going to be affected too, since a fast rate convergence is typical of what happens when the initial starting point $\underline{x}_1$ is close to a local minimizing point $\underline{x}^*$.

On the other hand, $f(\underline{x}_i)$ in its turn, may not be able to show any decreasing behavior; that is why one can make use in such similar situations of Newton's method which if associated with a specific line search technique should help in generating a correct step length. This involves solving the equation: $\underline{p}_i = -G^{-1}\underline{g}(\underline{x}_i)$, where a sufficient condition for the descent property requires that G^{-1} (and therefore G) be a positive definite matrix.

But, it is important to know that the domain of definition of Newton's method with line search can be extended in a way to include mal-behaved cases, where G may be singular or has one negative eigenvalue. Here, the stationary point of the approximating quadratic function $h_i(\underline{p})$ is not any more a minimizing point, and therefore some serious doubts rise as related to the relevance of searching in such a direction.

Many remedial steps have been thought of as being efficient means to overcome this major disadvantage. And some of these suggested modifications that aim to ensure that the Hessian matrix be positive definite or else to deal with situations where this Hessian is indefinite, are listed in the following steps:

1) By using the steepest descent direction with $\underline{p}_i = -\underline{g}(\underline{x}_i)$ whenever G is not positive definite. But, by ignoring the quadratic function model, Goldstein and Price [9] noticed here some slow oscillatory behavior.

2) By introducing a modification to the Newton search direction, through adding a multiple of the unit matrix to G which is close to being positive definite. Therefore by solving the system $(G+vI)\underline{p}_i = -\underline{g}(\underline{x}_i)$ as proposed by Levenberg and Marquardt [9] and implemented later on by Goldfeld, it will be helpful in biasing the direction towards the steepest descent vector $-\underline{g}(\underline{x}_i)$.

3) By computing a negative curvature descent direction of search satisfying the two inequalities $\underline{p}_i^T G \underline{p}_i < 0$ and $\underline{p}_i^T \underline{g}(\underline{x}_i) \leq 0$. Where, Fiacco and Mccormick [9] used the Hessian G in obtaining the LDL^T factors (which can show, unfortunately here, some unstability due to the indefi-

niteness of G), then solving afterwards the equation $L^T \underline{p}_i = \sigma \underline{a}$, where if $d_{ii} < 0 \;\; a_i = 1, \;\; else \;\; 0$, and $\sigma = \pm 1$ is chosen so that $\underline{p}_i^T \underline{g}(\underline{x}_i) \leq 0$.

4) And since Cholesky factors LL^T do not exist in the case of a non positive definite matrix, a modification suggested by Murray and Hebden [9] can be made aiming at a possible factorization by using a diagonal matrix D in addition to the Hessian, by simply using a positive definite matrix $G+D$ as a result.

The following is a rough description of the algorithm of Newton's method for minimization:

step 1. given the starting points $\underline{x}_0$ (for $i=0$)

step 2. evaluate $\underline{g}(\underline{x}_i)$

step 3. solve the system $G(\underline{x}_i).\underline{p}_i = -\underline{g}(\underline{x}_i)$ for $\underline{p}_i$

step 4. set $\underline{x}_{i+1} = \underline{x}_i + \underline{p}_i$

step 5. if $\|\underline{g}(\underline{x}_{i+1})\| < \varepsilon$ then stop

else $i:=i+1$

go to step 3.

2.3 Line Search Algorithms:

One of the subproblems addressed in a general unconstrained optimization is the line search algorithm, which with other prototype algorithms will be concerned, when starting from an initial estimate $\underline{x}_i$, in how to use the model prediction in such a way as to obtain satisfactory convergence properties.

The termination of the line search depends on whether certain conditions for an approximate minimum along the line are satisfied. In this respect, a variety of tests that do not need the knowledge of the solution can be implemented at this point, where a related one to the line search is the descent property which satisfies $\underline{p}_i^T \underline{g}(\underline{x}_i) < 0$, and that guarantees the function reduction for some $t_i > 0$.

In general, line searches are expensive especially if accuracy is to be guaranteed. For the time being, a big variety of line search algorithms is

available, where for all of them a primary requirement is the existence of first derivatives that satisfy some conditions in the process of finding a step t_i which reduces f "significantly" on every iteration. Accordingly, "accurate" or "exact" line search may be substituted for by an "inaccurate" or "relaxed" line search where significant decrease conditions are to be satisfied.

Although the idea is conceptually useful, one should expect that the minimizing value t_i might not exist, or if existing it cannot be determined in practice in a finite number of operations. But more specifically for cases of unconstrained optimization, most of the numerical methods which are iterative, start with an initial estimate of a certain minimum point, proceeding afterwards in generating a sequence of iterates by means of a certain line search technique until reaching the desired minimum. And here, is a rough description of the general idea of a line search algorithm, where variations in the direction of search $\underline{p}_i$ can occur depending on the chosen model. And in a sequence of steps procedure, one is expected to:

a) supply an initial estimate $\underline{x}_i$

b) specify a direction of search $\underline{p}_i$

c) find t_i that minimizes $f(\underline{x}_i + t\underline{p}_i)$ with respect to t

d) set $\underline{x}_{i+1} = \underline{x}_i + t_i \underline{p}_i$ (which is the equation of a n-dimensional straight line joining the two points $\underline{x}_i$ and $\underline{x}_{i+1}$).

It is essential here to notice that part c) of this procedure is the core of the line search problem which will be carried through the sampling of $f(\underline{x})$ and its derivatives for different points $\underline{x} = \underline{x}_i + t\underline{p}_i$ along the line.

The descent method, is one of these techniques in which the condition $\frac{df}{dt} < 0$ at $t=0$, should be satisfied by the direction of search $\underline{p}_i$. And by making use of some general mathematical results, like $\frac{df}{dt} = \underline{p}^T \nabla f = \nabla f^T \underline{p}$ one obtains a modified form of the condition that can be expressed as: $\underline{p}_i^T \underline{g}(\underline{x}_i) < 0$, where in certain cases and through a suitable choice of line search conditions, it is possible to incorporate the descent property

into a convergence proof. In fact, for the sake of terminating the iteration by the help of a convergence test, one should care to check after every iteration if: $f(\underline{x}_i) - f(\underline{x}^*) \le \varepsilon$ or $|\underline{x}_i - \underline{x}^*| \le \varepsilon$, ($\varepsilon$ being a parameter supplied by the user). Using merely the more efficient descent property to force a decrease $f(\underline{x}_{i+1}) < f(\underline{x}_i)$ in the objective function on successive iterations, does not necessarily ensure global convergence.

The resulting conditions in a line search satisfied for a certain value of t, must be such that an acceptable point always exists and can be determined in a finite number of steps. Here, the special step t_i found to give a significant reduction in f on each iteration, is not close to the extremes of the interval $[0, \bar{t}_i]$ where $\bar{t}_i$ denotes the least positive value of t for which $f(\underline{x}_i + t\underline{p}_i) = f(\underline{x}_i)$.

Denoting $f(\underline{x}_i + t\underline{p}_i)$ by $f(t)$, $f(\underline{x}_i)$ by $f(0)$ and the descent condition by $f'(0) < 0$, Goldstein stated two necessary conditions on t_i that meet the requirements already mentioned [9]. And if implemented properly, these conditions will act successively, to help in excluding the right hand extreme first and then the left hand extreme of $[0, \bar{t}_i]$, where they can be formulated as: $f(t) \le f(0) + t\rho f'(0)$ (2.2)
(known as the ρ-*condition* or the first condition), and consequently $f(t) \ge f(0) + t(1-\rho)f'(0)$. ρ being a fixed parameter such as $\rho \in (0, \frac{1}{2})$, allowing the property that the reduction in a function is acceptable. This first condition defines implicitly the so-called ρ-*line* that must be intersected by the graph of f (except when $f \to -\infty$), following consequently if satisfied by t_i, that the resulting reduction in $f(\underline{x})$ satisfies the following inequality:

$$f(\underline{x}_i) - f(\underline{x}_{i+1}) \ge -\rho \underline{g}(\underline{x}_i)^T t_i \underline{p}_i \quad \text{where} \quad t_i \underline{p}_i = \underline{x}_{i+1} - \underline{x}_i .$$

But for non-quadratic functions, it was suggested by Wolfe [9] that the second condition of Goldstein which may exclude the minimizing point of $f(t)$, can be replaced by a different test on the slopes, stating that:
$f'(t) \ge \sigma f'(0)$ where $\sigma \in (\rho, 1)$ is another parameter (2.3)
(known as the σ-*condition* or the second condition).

Consequently, this implies that $\underline{x}_{i+1}$ will satisfy the following inequality where: $\underline{g}(\underline{x}_{i+1})^T t_i \underline{p}_i \geq \sigma \underline{g}(\underline{x}_i)^T t_i \underline{p}_i$ helping, by the fact that $\sigma < 1$, to exclude the left-hand extreme of $[0, \bar{t}]$. And consequently, only finite number of steps are required to locate the acceptably existing points by the restriction $\sigma > \rho$.

At this point, one can see how a sequence of estimates t_i can be generated iteratively by using a certain line search algorithm, where, in the first bracketing phase of the procedure, a search is done to find a non-trivial bracket $[t_{lo}, t_{up}]$ containing an interval of acceptable points. Then. in the following sectioning phase the bracket is divided so as to generate a sequence of brackets whose length tends to zero. Afterwards, the sequence terminates when an iterate is located satisfying some standard conditions for an acceptable point.

And since it is preferable to find an acceptable point close to a local minimizer of *f(t)*, some form of interpolation is also desirable. There, one will consider fitting quadratic or cubic polynomials in *t* to some known data, and then choosing the following iterate t_{i+1} so as to minimize the polynomial.

In the process of implementing the line search algorithm requiring the evaluation of the function $f(\underline{x}_i + t\underline{p}_i)$ and of the gradient $\underline{g}(\underline{x}_i + t\underline{p}_i)$ at each iteration, the "cubic interpolation" algorithm will be used to obtain a good choice for *t* since the *natural* choice of assigning to *t* a unit step did not prove to be the best solution [9]. This interpolatory algorithm has shown in many instances, when compared with other interpolation procedures, to have a better performance in the domain of general unconstrained minimization.

In fact, "Cubic interpolation" that starts with a certain search interval $[t_{lo}, t_{up}]$, uses a cubic polynomial to interpolate the data given in the iterative process. Here, t_{lo} which is considered to correspond to the initial starting point $\underline{x}_i$, is chosen to be equal to 0; while t_{up} considered as the *natural* trial step, is set as equal to 1. The algorithm will be terminated,

and the latter trial step will be chosen as the *relaxed* minimum after testing the sufficient decrease conditions of Goldstein. In this respect, some bracketing refinements should be done to guarantee that the two conditions be satisfied at all times. Accordingly, if the Goldstein $\rho-condition$ is satisfied but not the other $\sigma-condition$, then one should set $t_{lo}=t_{up}$, and by a *step-doubling* the new interval bracket will become as $[t_{up},2t_{up}]$. Else, if the $\rho-condition$ is not satisfied, then one should assign a new value for t_{up}, where:

$$\text{the new } t_{up}=t_{up}-(t_{up}-t_{lo})\left[\frac{\phi'(t_{up})+w-z}{\phi'(t_{up})-\phi'(t_{lo})+2w}\right] \tag{2.4}$$

in which

$$z=\phi'(t_{lo})+\phi'(t_{up})-3\left[\frac{\phi(t_{up})-\phi(t_{lo})}{t_{up}-t_{lo}}\right], \text{ and } w=\sqrt{z^2-\phi'(t_{lo})\phi'(t_{up})}\,. \tag{2.5}$$

The Goldstein conditions are satisfied with a very small value for ρ (where $\rho=10^{-4}$), and a large value for σ (where σ=*0.9*).

2.4 "Cubic Interpolation" Algorithm:

The "cubic interpolation" algorithm can be described in a procedural way as follows:

step 1. $t_{lo}\leftarrow 0$, $find\phi_0\leftarrow\phi(0)$, $\phi'_0\leftarrow\phi'(0)$, $\phi_{lo}\leftarrow\phi_0$, $\phi'_{lo}\leftarrow\phi'_0$,
$t_{up}\leftarrow 1$, $\rho\leftarrow 10^{-4}$, $\sigma\leftarrow 0.9$, $done\leftarrow false$, $tup\_not\_acceptable\leftarrow false$,

repeat

step 2. Find $\phi_{up}\leftarrow\phi(t_{up})$, $\phi'_{up}\leftarrow\phi'(t_{up})$

step 3. if $\phi_{up}\le\phi_0+\rho[t_u\phi'_0]$ then

if $\phi'_{up}>\sigma\phi'_0$ then

$t_j\leftarrow t_{up}$, $done\leftarrow true$ $\{acceptable\}$

else

$t_{lo}\leftarrow t_{up}$, $\phi_{lo}\leftarrow\phi_{up}$, $\phi'_{lo}\leftarrow\phi'_{up}$, $t_{up}\leftarrow 2t_{up}$

endif

else

$tup\_not\_acceptable\leftarrow true$

endi

until (*done*) or (*tu_not_acceptable*)
while not (*done*) *do*

step 4.

$$z \leftarrow \phi'_{lo} + \phi'_{up} - 3\left[\frac{\phi_{up} - \phi_{lo}}{t_{up} - t_{lo}}\right]$$

$$w \leftarrow \sqrt{z^2 - \phi'_{lo}\phi'_{up}}$$

$$t_j \leftarrow t_{up} - (t_{up} - t_{lo})\left[\frac{\phi'_{up} + w - z}{\phi'_{up} - \phi'_{lo} + 2w}\right]$$

step 5. $\phi_j \leftarrow \phi(t_j),\ \ \phi'_j \leftarrow \phi'(t_j)$

step 6. if $\phi_j \le \phi_0 + \rho\left[t_j\phi'_0\right]$ then

if $\phi'_j > \sigma\phi'_0$ then

$done \leftarrow true$ $\quad \{t_j \ \ is \ \ acceptable\}$

else

$t_{lo} \leftarrow t_j,\ \ \phi_{lo} \leftarrow \phi_j,\ \ \phi'_{lo} \leftarrow \phi'_j$

endif

else

$t_{up} \leftarrow t_j,\ \ \phi_{up} \leftarrow \phi_j,\ \ \phi'_{up} \leftarrow \phi'_j$

endif

enddo

return t_j.

CHAPTER THREE

QUASI-NEWTON METHODS

FOR UNCONSTRAINED OPTIMIZATION

AND THE BROYDEN FAMILY

3.1 Motivation Of The Quasi-Newton Methods:

Many disadvantages of the Newton's method and as of other closely related methods, were proved to be avoidable through the use of the more advanced technique enhanced by the class of quasi-Newton methods, which widened the scope of problem solving.

In fact, many of those disadvantages were pointed out in the previous chapter, as well as some of the suggested modifications that can help to ensure the global convergence for a considered function, the positive definiteness of the Hessian and the reduction of the tedious computations of the Hessian at every iteration. In this respect, the finite difference Newton method can serve as an example of an attempt aiming towards a real improvement, where a summary of all worries and limitations can be obtained by considering some of the possible resulting drawbacks. Here, it was proposed to estimate the Hessian $G(\underline{x}_i)$ by differences in gradient vectors, where the i_{th} column of the matrix which is expressed as $\overline{G}$ can be evaluated as : $\frac{\underline{g}(\underline{x}_i + h_i \underline{e}_i) - \underline{g}(\underline{x}_i)}{h_i}$, h_i being an increment considered in a coordinate direction $\underline{e}_i$. No doubt, as pointed by Curtis [9] this method have shown usefulness especially in the case of large sparse problems with reduced differencing amounts; but still we have to do n gradient evaluations and a set of linear equations that should be solved at each iteration. Added to this, is the fact that the Hessian may not be positive definite even after making it symmetric by taking $\frac{1}{2}(\overline{G} + \overline{G}^T)$.

On the other hand, when shifting to quasi-Newton methods where only first derivatives are required, most of the work is centered on the updating formula that provides a mean to calculate the Hessian at every iteration from the one which was used in the previous iteration.It is worth

mentioning at this point that the Hessian is surely positive definite, implying the descent property. Here, one way to avoid the solution of a system of equations at every iteration is achieved by considering the inverse Hessian matrix $H(\underline{x}_i) = B^{-1}(\underline{x}_i)$, where $B(\underline{x}_i)$ is a symmetric positive definite matrix approximating the original $G(\underline{x}_i)$. To start, any positive definite matrix can serve as the initial inverse Hessian matrix $H(\underline{x}_1)$, accordingly $H(\underline{x}_1) = I$ could be a good choice. Indeed, in this respect, one major achievement reached by using the approximation matrix to G^{-1} rather than to G, is a reduced number of multiplications per iteration which leads to $O(n^2)$.

Consequently, the algorithm at the i_{th} iteration is described by the following steps:

1) set $\underline{p}_i = -H(\underline{x}_i)\underline{g}(\underline{x}_i)$

2) perform line search along $\underline{p}_i$ to determine t_i in $\underline{x}_{i+1} = \underline{x}_i + t_i \underline{p}_i$

3) update H_i giving H_{i+1}

3.2 The "Secant Equation":

Originally, a path X was considered to be a straight line L, of the form $\underline{x}(t) = \underline{x}_i + t\underline{s}_i$ helping to generate a new iterate $\underline{x}_{i+1}$ from a previous iterate $\underline{x}_i$. And consequently, a set vector $\underline{s}_i$ can be defined when t is considered to be equal to *one*, as: $\underline{s}_i \cong \underline{x}_{i+1} - \underline{x}_i$, where also: $\forall t \in \Re, \frac{d\underline{x}(t)}{dt} = \underline{s}_i$. (3.1)

A popular choice of a *t-value* equal to *one* leads to the generation of a new iterate $\underline{x}_{i+1} = \underline{x}(1)$ from the previous one $\underline{x}_i = \underline{x}(0)$ for $t = 0$. Then, by making use of the first order Taylor's series:

$$\underline{g}(\underline{x}(t)) = \underline{g}(\underline{x}(0)) + t\left|\underline{g}(\underline{x}(1)) - \underline{g}(\underline{x}(0))\right| \Rightarrow \left.\frac{d\underline{g}}{dt}\right|_{t=1} \cong \underline{g}_{i+1} - \underline{g}_i = \underline{y}_i \qquad (3.2)$$

where: $\underline{g}_{i+1} = \underline{g}(\underline{x}(1)) = \underline{g}(\underline{x}_{i+1})$ and $\underline{g}_i = \underline{g}(\underline{x}(0)) = \underline{g}(\underline{x}_i)$

and considering thereafter an objective function $f(\underline{x})$, where $(f:\Re^n \to \Re | \underline{x} \in \Re^n)$, $\{\underline{x}(t)\}$ denoting a differentiable path in $\Re^n$ in which

$t \in \Re$; then applying as a next step the chain rule to the gradient vector $\underline{g}(\underline{x}(t))$ with respect to t, one gets the following expression:

$$\frac{d\underline{g}}{dt} = G(\underline{x}(t)).\frac{d\underline{x}}{dt}$$

which must be satisfied for any value of t, resulting in the so-called "Newton's equation", such as at $t = t_i$ it can be expressed as:

$$\left.\frac{d\underline{g}}{dt}\right|_{t=t_i} = G(\underline{x}(t)).\left.\frac{d\underline{x}}{dt}\right|_{t=t_i} \tag{3.3}$$

where by substituting the derived values of $\left.\frac{d\underline{g}(\underline{x}(t))}{dt}\right|_{t=1}$ and $\left.\frac{d\underline{x}(t)}{dt}\right|_{t=1}$ one gets an equation which is an approximation of the form:

$$\underline{y}_i \cong G(\underline{x}_{i+1}).\underline{s}_i \tag{3.4}$$

which, after substituting the inverse Hessian matrix $G^{-1}(\underline{x}_{i+1})$ by its approximate H_{i+1}, leads to the secant equation expressed as:

$$H_{i+1}.\underline{y}_i = \underline{s}_i \tag{3.5}$$

known also as the quasi-Newton condition [7].

3.3 Possible Ways For Achieving The Quasi-Newton Condition:

In fact there are an infinite number of possibilities for choosing the H_{i+1} symmetric matrix with $(n^2+n)/2$ unknowns. Consequently, an essential task aims here at finding ways for updating the Hessian, H_i / B_i to obtain H_{i+1} / B_{i+1}. It is useful at this point to introduce a correction symmetric matrix C_i where:

$$H_{i+1} = H_i + C_i . \tag{3.6}$$

In this respect, a symmetric rank one formula can be generated to fit in the secant equation, where also, for the sake of satisfying the symmetry property, C_i should be of the form: $C_i = \gamma(\underline{s}_i - H_i\underline{y}_i)(\underline{s}_i - H_i\underline{y}_i)^T$, implying that:

$$\gamma(\underline{s}_i - H_i\underline{y}_i)(\underline{s}_i - H_i\underline{y}_i)^T\underline{y}_i = \underline{s}_i - H_i\underline{y}_i \tag{3.7}$$

leading in its turn to an expression for γ as: $\gamma = \frac{1}{(\underline{s}_i - H_i\underline{y}_i)^T\underline{y}_i}$, and

therefore: $$H_{i+1} = H_i + \frac{(\underline{s}_i + H_i \underline{y}_i)(\underline{s}_i + H_i \underline{y}_i)^T}{(\underline{s}_i - H_i \underline{y}_i)^T \underline{y}_i}. \qquad (3.8)$$

In fact, if it is well defined, this rank one method terminates on a quadratic function in at most $(n+1)$ searches with $H_{n+1} = G^{-1}$, presumed that $\underline{s}_1, \underline{s}_2, ..., \underline{s}_n$ are independent. A direct implication of this is that it does not require exact line searches, and not even to use $\underline{p}_i = -H_i . \underline{g}(\underline{x}_i)$ except on the last $(n+1)_{st}$ iteration.

Further, by considering a rank two correction, then a more flexible formula will be obtained, expressed as: $H_{i+1} = H_i + C_i$
where: $C_i = a\underline{u}_i \underline{u}_i^T + b\underline{v}_i \underline{v}_i^T$, and $\underline{s}_i = (H_i + a\underline{u}_i \underline{u}_i^T + b\underline{v}_i \underline{v}_i^T).\underline{y}_i$ should be satisfied.

An obvious choice here, is to substitute $\underline{u}_i$ by $\underline{s}_i$, and $\underline{v}_i$ by $H_i \underline{y}_i$, leading to: $C_i = a\underline{s}_i \underline{s}_i^T + b(H_i \underline{y}_i)(H_i \underline{y}_i)^T$, which if substituted in $C_i \underline{y}_i = \underline{s}_i - H_i \underline{y}_i$ gives

$$\left[a\underline{s}_i \underline{s}_i^T + b(H_i \underline{y}_i)(H_i \underline{y}_i)^T\right]\underline{y}_i = \underline{s}_i - H_i \underline{y}_i$$

or $\left[a(\underline{s}_i^T \underline{y}_i)\right]\underline{s}_i + \left[b\underline{y}_i^T H_i \underline{y}_i\right]H_i \underline{y}_i = \underline{s}_i - H_i \underline{y}_i$ where by comparing the two sides of this equation, one gets that: $a\underline{s}_i^T \underline{y}_i = 1$ and $b\underline{y}_i^T H_i \underline{y}_i = -1$

i.e. $$a = \frac{1}{\underline{s}_i^T \underline{y}_i} \quad \text{and} \quad b = \frac{-1}{\underline{y}_i^T H_i \underline{y}_i},$$

consequently: $$C_i = \frac{\underline{s}_i \underline{s}_i^T}{\underline{s}_i^T \underline{y}_i} - \frac{(H_i \underline{y}_i)(H_i \underline{y}_i)^T}{\underline{y}_i^T H_i \underline{y}_i}$$

where if substituting this last expression in the updating form of H_{i+1}, it generates the DFP formula initiated by Davidon, and presented at a later stage by Fletcher and Powell (in 1963) [13].

Applying the method in the case of quadratic functions can lead to:
a) a termination in at most n iterations with $H_{n+1} = G^{-1}$
b) a preservation of the quasi-Newton conditions
c) a generation of conjugate directions, and conjugate gradients when $H_0 = I$.