A Variable Storage Method For Conic Functions

Dimitrios Tegos

A Thesis

in

The Department

of

Mathematics

Presented in Partial Fulfillment of the Requirements
for the degree of Master of Science at
Concordia University
Montréal, Québec, Canada

August 1983

TABLE OF CONTENTS

ABSTRACT

A VARIABLE STORAGE METHOD FOR CONIC FUNCTIONS

Dimitrios Tegos

Most of the current optimization algorithms update quadratic approximations to their objective functions. A new class of algorithms has recently been developed using conic approximations. These algorithms require both function and gradient evaluations at each step to construct the approximating function, while the algorithms based on quadratics use only gradient values. Hence the new algorithms are likely to give better estimates than the ones based on quadratics. The concept of these algorithms was originated by Davidon in 1980. General features of the algorithms based on conics will be presented and discussed in this thesis. Conic algorithms due to Gourgeon and Nocedal and to Davidon will be described and investigated. It is shown here that these methods are identical on conics, which is stated without proof by Gourgeon and Nocedal and by Davidon. Also, a new method based on conics is introduced which is a combination of two algorithms introduced by Davidon. This new algorithm can use whatever storage is available and its derivation is based on the VSCG algorithm for quadratics which was introduced by Buckley and LeNir.

ACKNOWLEDGEMENT

This thesis is dedicated to my parents, Thomas and Anastasia, for their understanding and support during my studies.

I wish to express my sincere appreciation to Dr. A. Buckley for his assistance, encouragement and guidance in writing this thesis.

I also wish to thank Mrs. Judith Hickey for her excellent secretarial work.

CHAPTER I

INTRODUCTION

1.1 Background

A typical mathematical problem, which is often a model of physical reality, is to minimize or maximize a given differentiable function $f(x)$ of n real variables $x = (x_1,\ldots,x_n)^T$. The objective of this thesis is to study the unconstrained case of this problem, since firstly many of the methods used for the unconstrained case can be modified to solve many constrained problems, and secondly because constrained optimization problems usually involve solving unconstrained subproblems. Thus we are interested in the following type of problem:

$$\text{minimize} \quad f(x)\,, \quad x \in \mathbb{R}^n \quad . \tag{1.1}$$

Here $f(x)$ is assumed to be twice continuously differentiable. It is also assumed that the function and the first derivative values can be evaluated at any point x. Our interest will be in methods referred to as "gradient methods" for which there is no requirement of evaluating the Hessian. Also, there is no loss of generality in taking a minimizing problem only, since maximizing f is the same as minimizing $-f$.

Many of the minimization methods which have been developed in the last twenty years are derived in such a way as to minimize a quadratic function in a finite number of steps, a property which is known as finite (or occasionally quadratic) termination. Then, using the fact that a twice continuously differentiable function can be approximated by a quadratic near the minimum, these methods can be applied to minimize general functions. Our interest in this thesis will be in other, non-quadratic, models, namely conics.

The two basic types of gradient methods based on quadratic models are the quasi-Newton (QN) - also known as "variable metric" or "secant" - methods and the conjugate gradient (CG) method. Both of these require just the evaluation of the gradient at each step. The quasi-Newton method was introduced by Davidon in 1959 [5]. Since that time, several variations of the QN methods have been developed by many authors. The most important among these is the Broydon β-class introduced by Broyden in 1967 [2], and particularly the BFGS algorithm, which is a member of this class. The conjugate gradient method was introduced by Fletcher and Reeves in 1964 [10], but we will be interested in the modification of the CG method generally referred to as the preconditioned conjugate gradient method (PCG).

Each of these two basic methods has advantages and disadvantages. In particular, quasi-Newton methods require computation with and storage of an $n \times n$ matrix, since they calculate an approximation to the Hessian matrix at each step. Therefore computer storage of $O(n^2)$ locations is required. On the other hand, the conjugate gradient methods require much less storage, typically $3n$ to $7n$ locations, depending on the method used. However, it has been proved that in general the QN methods converge more rapidly and require fewer function evaluations than the CG methods. Therefore the CG methods find application when storage limitations occur, since they require much less storage than a QN method; but they are slower. So, recently, methods have been developed as a combination of these two in order to obtain an algorithm with good covergence properties and low storage requirements. In particular, we will be interested in the algorithm introduced by Buckley and LeNir [4] in 1982 and known as the VSCG algorithm. A main objective of the VSCG method is to allow the use of whatever storage is available since the use of more storage will improve the speed of convergence. The VSCG algorithm will be described in more detail in Chapter III.

In 1980, Davidon proposed a new class of algorithms for the unconstrained minimization problem (1.1). The idea is, instead of using a quadratic approximating function, to use a certain rational function called a conic. This new class of algorithms based on conics requires function and gradient evaluations at each step, while the methods based on quadratics require only gradient evaluations for building the method. Hence the methods based on conics make use of more information for general functions and therefore we hope to have better performance than the current ones based on quadratics.

Davidon introduced two basic conjugate direction algorithms based on conics. The first is a generalization of the conjugate gradient method and uses $O(n)$ operations per iteration; for that reason Davidon refers to it as "the $O(n)$ algorithm". For brevity, we will refer just to "the $O(n)$". The second one can be considered as a generalization of the QN method and particularly of the BFGS algorithm. It of course uses $O(n^2)$ operations per iteration and so Davidon called it "the $O(n^2)$ algorithm"; we will write "the $O(n^2)$".

An algorithm similar to $O(n)$, but independently developed from a different point of view, was given by Gourgeon and Nocedal in 1982 [11]. We will call this the GN algorithm. A primary purpose of this thesis is to clarify the relation between the GN method and Davidon's methods. For this reason we shall give detailed derivations of Davidon's methods, since his description in [6] is very brief. We will show that the

$O(n)$, $O(n^2)$ and GN methods are identical. In so doing, a careful description of the role of the "reference point" is required since it is a key feature of conic functions and is handled differently by Gourgeon and Nocedal and by Davidon.

The final purpose of this thesis is to introduce an algorithm based on conic models and similar to VSCG. The new algorithm, like VSCG, will be considered as a combination of the conjugate direction and variable metric methods, but based on conics, so that we will obtain good convergence properties and low storage requirements.

1.2 Preliminaries

Some of the common symbols to be used in this thesis are the following. Capital letters denote $n \times n$ matrices or transformations. Lower case letters denote column vectors or scalars; Greek letters denote certain common scalars also. A superscript ^{T} will indicate transposition, and all row vectors will be written with a transpose.

All methods which will be discussed are iterative, i.e. given an initial starting point x_o, a sequence of points $x_1, x_2, \ldots$ is produced until a local minimum x^* of a function $f(x)$ is reached (at least approximately). The construction of x_k will be by

$$x_k = x_{k-1} + t_k d_{k-1} , \tag{1.2}$$

where d_{k-1} is a direction of search determined according to the particular method being used. Then x_k is the k^{th} approximation to x^*.

We let $q(x)$ be a quadratic function given by

$$q(x) = \frac{1}{2} x^T Q x + z^T x + \tau$$

where Q is an $n \times n$ positive definite matrix, z is an arbitrary vector in $\mathbb{R}^n$ and τ is an arbitrary constant. We define g to be the gradient vector of $f(x)$ at x, i.e. $g \equiv \nabla f$, so

$$g_k \equiv g(x_k) = \nabla f(x_k) .$$

If we refer to the general quadratic function $q(x)$, then define

$$h_k \equiv h(x_k) = \nabla q(x_k) .$$

We are interested only in descent methods; therefore $d_k^T g_k < 0$ must hold for each k. The value $t_k > 0$ is chosen to minimize the one dimensional function

$$\phi(t) = f(x_{k-1} + td_{k-1}) .$$

The process of finding the minimum point on a given line is called the line search. If t_k is an exact local minimum of $\phi(t)$ then

$$\left.\frac{d\phi(t)}{dt}\right|_{t_k} \equiv \phi'(t_k) = d_{k-1}^T g_k = 0 ,$$

and the line search will be referred to as an exact line search (ELS). In this thesis we are interested in exact line searches since we have restricted the function class to conics. We will discuss this point in Chapter II.

CHAPTER II

INTRODUCTION TO CONICS

2.1 Basic Forms

This section introduces the basic terms, concepts and relations for conics which will be used later for the derivation of the algorithms based on them. We also take this opportunity to introduce much of our notation.

We now define the conic function, first proposed by Davidon [7] in 1980.

Definition 1: A smooth function $f: X \to \mathbb{R}$ where $X \subset \mathbb{R}^n$ is said to be conic iff it is a ratio of a quadratic to the square of a linear function.

This representation is not unique, as the conic may be expanded around different points, as we will see. An equivalent algebraic definition for a conic function is the following. Given a point x_o, called the reference point, a function $f: X \to \mathbb{R}$ is conic iff there is an $a_o \in \mathbb{R}^n$ and A_o, an $n \times n$ matrix, with $a_o^T(x-x_o) < 1$ and

$$f(x) = f_o + \frac{g_o^T s}{1 - a_o^T s} + \frac{1}{2}\frac{s^T A_o s}{(1 - a_o^T s)^2}, \quad \text{where } s = x - x_o . \quad (2.2)$$

The vector a_o will be referred to as the horizon vector of f relative to the reference point x_o. We assume that the matrix A_o is a positive definite matrix, and A_o is called the conjugacy matrix for f at x_o. Also we define the affine function $\gamma(x) \equiv 1 - a_o^T s$. We call $\gamma(x)$ the gauge of f at x relative to the reference point x_o. For abbreviation, we write $\gamma_k \equiv \gamma(x_k) = 1 - a_o^T s_k$ as the gauge of f at $x_k = x_o + s_k$, where $s_k = x_k - x_o$. The horizon vector a_o, the conjugacy matrix A_o and the gauge $\gamma(x)$ are all relative to the reference point x_o.

However the reference point is not unique, so we will next show how the horizon vector, the conjugacy matrix and the gauge change as the conic is expanded around different points. This will be a key point in the understanding of the conic algorithms. According to the above discussion we can define a conic as:

$$f(x) = \frac{q(x)}{[\gamma(x)]^2} = \frac{q_o + h_o^T s + \frac{1}{2} s^T Q_o s}{(1 - a_o^T s)^2} . \qquad (2.3)$$

Since for $s = 0$ we have $\gamma(x_o) = 1$, by using (2.3) we get $f_o = q_o$. By putting (2.2) over a common denominator and then comparing with (2.3), we obtain the following relations:

$$g_o = h_o + 2 f_o a_o \qquad (2.4)$$

$$A_o = Q_o + g_o a_o^T + a_o g_o^T - 2 f_o a_o a_o^T . \qquad (2.5)$$

Now we will change the reference point from x_o to x_k. First note that

$$\begin{aligned} \gamma(x) &= 1 - a_o^T s = 1 - a_o^T (x - x_o) \\ &= 1 - a_o^T (x - x_k + x_k - x_o) \\ &= 1 - a_o^T (x_k - x_o) - a_o^T (x - x_k) \\ &= \gamma(x_k) - a_o^T (x - x_k) = \gamma(x_k) \left[1 - \frac{a_o^T}{\gamma(x_k)} (x - x_k) \right] . \end{aligned} \qquad (2.6)$$

By using (2.6) we observe and define the following:

$$a_k \equiv \frac{a_o}{\gamma(x_k)} \qquad (2.7)$$

$$\gamma_k(x) \equiv 1 - a_k^T (x - x_k) , \qquad (2.8)$$

where a_k and $\gamma_k(x)$ denote the horizon vector and the gauge relative to the new reference point x_k. According to (2.6), we have the following relation between $\gamma_k(x)$ and $\gamma(x)$,

$$\gamma_k(x) = \frac{\gamma(x)}{\gamma(x_k)} = \frac{\gamma(x)}{\gamma_k} . \tag{2.9}$$

Note that only $\gamma(x_k)$ will be abbreviated to γ_k. To avoid confusion, $\gamma_k(x)$ is always written in full. Also, we can write f relative to x_k as

$$f(x) = \frac{f_k + \bar{h}_k^T \bar{s} + \frac{1}{2}\bar{s}^T Q_k \bar{s}}{\gamma_k^2(x)} , \text{ where } \bar{s} \equiv x - x_k . \tag{2.10}$$

Obviously, the corresponding relations to (2.4) and (2.5) for g_k and A_k relative to x_k are the following:

$$g_k = \bar{h}_k + 2 f_k a_k , \tag{2.11}$$

$$A_k = Q_k + g_k a_k^T + a_k g_k^T - 2 f_k a_k a_k^T . \tag{2.12}$$

By comparing (2,3) with (2.10), and using (2.9), we obtain

$$Q_k = \frac{Q_o}{\gamma^2(x_k)} , \tag{2.13}$$

$$\bar{h}_k = \frac{h_k}{\gamma^2(x_k)} \tag{2.14}$$

where $$h_k \equiv \nabla q(x_k) = h_o + Q_o s_k . \tag{2.15}$$

These relations (2.2) to (2.15) can now be used to get g_k and A_k directly in terms of g_o and A_o. This is important because it is in the form (2.2) that we will wish to study f. The following simple relation will be used:

$$\frac{a_o^T s_k}{\gamma_k} = \frac{1-\gamma_k}{\gamma_k} = \frac{1}{\gamma_k} - 1 . \tag{2.16}$$

First we state the desired relations.

$$1. \qquad g_k = J_k^T J^{-T}\left(g_o + \frac{A_o s_k^T}{\gamma_k}\right) \tag{2.17}$$

where $$J(x_k) \equiv J_k \equiv \frac{J}{\gamma_k}\left[I + \frac{s_k a_o^T}{\gamma_k}\right]. \tag{2.18}$$

Here J is an arbitrary invertible $n \times n$ matrix. The reason for this choice of the matrix J_k will become clear later in this section.

$$2. \qquad A_k = J_k^T J^{-T} A_o J^{-1} J_k . \tag{2.19}$$

We now prove these relations. To get (2.17), we first substitute (2.14), (2.15) and (2.7) into (2.11). Indeed

$$\begin{aligned} g_k &= \bar{h}_k + 2f_k a_k \\ &= \frac{h_k}{\gamma_k^2} + 2f_k \frac{a_o}{\gamma_k} \\ &= \frac{1}{\gamma_k}\left[\frac{h_o + Q_o(x_k - x_o)}{\gamma_k} + 2f_k a_o\right]. \end{aligned}$$

By substituting (2.4), (2.5) and (2.2) for $x = x_k$ into the last relation we obtain

$$\begin{aligned} g_k &= \frac{1}{\gamma_k}\left[\frac{1}{\gamma_k}\left(g_o - 2f_o a_o + [A_o - g_o a_o^T - a_o g_o^T + 2f_o a_o a_o^T] s_k\right)\right. \\ &\quad \left. + 2\left(f_o + \frac{g_o^T s_k}{\gamma_k} + \frac{1}{2}\frac{s_k^T A_o s_k}{\gamma_k^2}\right) a_o\right] \\ &= \frac{1}{\gamma_k}\left[\frac{g_o - 2f_o a_o + A_o s_k - g_o(a_o^T s_k) - (g_o^T s_k) a_o + 2f_o a_o (a_o^T s_k)}{\gamma_k}\right. \\ &\quad \left. + 2f_o a_o + \frac{2(g_o^T s_k) a_o}{\gamma_k} + \frac{a_o s_k^T A_o s_k}{\gamma_k^2}\right] \end{aligned}$$

and by using (2.16) we have

$$g_k = \frac{1}{\gamma_k}\left[\frac{g_o}{\gamma_k} - g_o\left(\frac{1}{\gamma_k}-1\right) - \frac{2f_o a_o}{\gamma_k} + 2f_o a_o\left(\frac{1}{\gamma_k}-1\right)\right.$$

$$\left. + 2f_o a_o + \frac{A_o s_k}{\gamma_k} + \frac{(g_o^T s_k)a_o}{\gamma_k} + \frac{(a_o s_k^T)}{\gamma_k}\frac{A_o s_k}{\gamma_k}\right]$$

$$= \frac{1}{\gamma_k}\left(g_o + \frac{a_o s_k^T}{\gamma_k} g_o + \frac{1}{\gamma_k} A_o s_k + \frac{a_o s_k^T}{\gamma_k}\frac{A_o s_k}{\gamma_k}\right)$$

$$= \frac{1}{\gamma_k}\left(I + \frac{a_o s_k^T}{\gamma_k}\right)\left(g_o + \frac{A_o s_k}{\gamma_k}\right)$$

$$= J_k^T J^{-1}\left(g_o + \frac{A_o s_k}{\gamma_k}\right).$$

To prove (2.19), first note according to (2.13), that $\gamma_k^2 Q_k = Q_o$.

Then, by substituting (2.5) and (2.12) into the last equality, we obtain

$$\gamma_k^2(A_k - g_k a_k^T - a_k g_k^T + 2f_k a_k a_k^T) = A_o - g_o a_o^T - a_o g_o^T + 2f_o a_o a_o^T .$$

Therefore

$$A_k = g_k a_k^T + a_k g_k^T - 2f_k a_k a_k^T + \frac{1}{\gamma_k^2}[A_o - g_o a_o^T - a_o g_o^T + 2f_o a_o a_o^T]$$

and by again substituting (2.2) for $x = x_k$, (2.7), and (2.17) into the last equality and by using also (2.18) we get

$$A_k = \frac{1}{\gamma_k}\left[\frac{g_o a_o^T}{\gamma_k} + \frac{a_o a_o^T}{\gamma_k^2} s_k^T g_o + \frac{1}{\gamma_k^2} A_o s_k a_o^T + \frac{s_k^T A_o s_k}{\gamma_k^3} a_o a_o^T\right]$$

$$+ \frac{1}{\gamma_k}\left[\frac{a_o g_o^T}{\gamma_k} + \frac{a_o a_o^T}{\gamma_k^2} s_k^T g_o + \frac{1}{\gamma_k^2} a_o s_k^T A_o + \frac{s_k^T A_o s_k}{\gamma_k^3} a_o a_o^T\right]$$

$$- \frac{2}{\gamma_k^2} f_o a_o a_o^T - \frac{2 g_o^T s_k}{\gamma_k^3} a_o a_o^T - \frac{s_k^T A_o s_k}{\gamma_k^4} a_o a_o^T$$

$$+ \frac{1}{\gamma_k^2}\left[A_o - g_o a_o^T - a_o g_o^T + 2f_o a_o a_o^T \right]$$

$$= \frac{1}{\gamma_k^2}(g_o a_o^T + a_o g_o^T) + 2\frac{g_o^T s_k}{\gamma_k^3} a_o a_o^T + \frac{1}{\gamma_k^3}(A_o s_k a_o^T + a_o s_k^T A_o)$$

$$+ 2\frac{s_k^T A_o s_k}{\gamma_k^4} a_o a_o^T - 2\frac{g_o^T s_k}{\gamma_k^3} a_o a_o^T - \frac{s_o^T A_o s_k}{\gamma_k^4} a_o a_o^T$$

$$+ \frac{1}{\gamma_k^2} A_o - \frac{1}{\gamma_k^2}(g_o a_o^T + a_o g_o^T)$$

$$= \frac{1}{\gamma_k^2}\left[\frac{A_o s_k a_o^T}{\gamma_k} + \frac{a_o s_k^T A_o}{\gamma_k} + \frac{a_o s_k^T A_o s_k a_o^T}{\gamma_k^2} + A_o \right]$$

$$= \frac{1}{\gamma_k^2}\left[I + \frac{a_o s_k^T}{\gamma_k} \right] A_o \left[I + \frac{s_k a_o^T}{\gamma_k} \right] .$$

$$= J_k^T J^{-T} A_o J^{-1} J_k .$$

We now wish to discuss the concept of a collinear-scaling. This is the second key to understanding conic algorithms. A conic function is closely related to a quadratic one for consider transforming the variables under a general transformation

$$w = T(x) \equiv \frac{J(x - x_o)}{1 - a_o^T(x - x_o)} = \frac{Js}{\gamma(x)} . \qquad (2.20)$$

This will make $f(w)$ quadratic as we will see. Here J is as before any invertible $n \times n$ matrix; it defines the transformation T. This transformation T is known as a collinear scaling of the variables, and was first introduced by Davidon in [7], where he discusses the motivation for this terminology. The Jacobian of T at x has the following form:

$$J(x) = \frac{J}{\gamma(x)} \left[I + \frac{sa_o^T}{\gamma(x)} \right] , \tag{2.21}$$

and the inverse of $J(x)$ is

$$J^{-1}(x) = \gamma(x) \, [I - sa_o^T] \, J^{-1} . \tag{2.22}$$

With a simple change to the formula (2.2) and by using (2.20) we see that

$$\begin{aligned} f(x) &= f_o + \frac{g_o^T J^{-1} Js}{\gamma(x)} + \frac{1}{2} \frac{s^T J^T J^{-T} A_o J^{-1} Js}{\gamma^2(x)} \\ &= f_o + (J^{-T} g_o)^T w + \frac{1}{2} w^T (J^{-T} A_o J^{-1}) \, w \\ &\equiv \bar{q}(w) , \end{aligned} \tag{2.23}$$

which is quadratic in w with Hessian $J^{-T}A_oJ^{-1}$. As a consequence a conic function has a unique minimizer in almost the same cases as a quadratic does, as discussed in Davidon [7]. In this case, when A_o is positive definite and $a_o^T A_o^{-1} g_o \neq 1$, the conic function will be called normal. In the remainder of this thesis, we assume f is normal, just as in the quadratic case, one assumes that the Hessian is positive definite.

To conclude this section on basics, we will now show how to compute values of the gauge without having to know the horizon vector a_o. First of all, Davidon defined the quantity

$$p_{ij} \equiv \frac{1}{2} \left(\frac{\gamma_j}{\gamma_i} g_j - \frac{\gamma_i}{\gamma_j} g_i \right)^T (x_j - x_i) , \tag{2.24}$$

for which there are alternate expressions, which Davidon also derived in [7] . These are the following:

$$p_{ij} = \frac{1}{2}\left(\frac{s_i}{\gamma_i} - \frac{s_j}{\gamma_j}\right)^T A_o\left(\frac{s_i}{\gamma_i} - \frac{s_j}{\gamma_j}\right) , \tag{2.25}$$

$$p_{ij} = f_i - f_j + \frac{\gamma_j}{\gamma_i} g_j^T(x_j - x_i)$$

$$= f_j - f_i - \frac{\gamma_i}{\gamma_j} g_i^T(x_j - x_i) , \tag{2.26}$$

$$p_{ij}^2 = (f_j - f_i)^2 - g_i^T(x_j - x_i)\, g_j^T(x_j - x_i) . \tag{2.27}$$

The above equivalent expressions for p_{ij} establish the following theorem, which is due to Davidon [1980]:

<u>Theorem 1</u>: Let f be a normal conic function and let x_i and x_j be two points in X such that $f_i < f_j$. Then

$$\frac{\gamma_i}{\gamma_j} = \frac{-g_i^T(x_i - x_j)}{f_j - f_i + p_{ij}} . \tag{2.28}$$

Note that the ratio γ_i/γ_j will be denoted by $r_{ij} \equiv \gamma_i/\gamma_j$. Therefore by applying Theorem 1 we can compute the ratio of gauges. In particular, for $i = k$ and $j = 0$ we can find the value of the gauge at any point x_k in terms of the function and gradient values at x_o and x_k since $\gamma(x_o) = 1$. Indeed

$$\gamma_k = \frac{\gamma_k}{\gamma_o} = \frac{-g_o^T(x_k - x_o)}{f_o - f_k + p_{ko}} .$$

Also, since γ is an affine function, knowledge of γ_o and γ_k allow us to find the value of γ at any point on the line joining x_o and x_k.

Davidon also established, by the following Lemma, that by evaluating the gradient at three collinear points, where one of these is the reference point, we can determine the horizon vector a_o.

<u>Lemma 1</u>: Let $x_1 = x_o + t_1 d$, $x_2 = x_o + t_2 d$ be two points in x . Then the horizon vector a_o is given by

$$a_o = \frac{\frac{\gamma_1}{t_1}(\gamma_1 g_1 - g_o) - \frac{\gamma_2}{t_2}(\gamma_2 g_2 - g_o)}{(\gamma_1^2 g_1 - \gamma_2^2 g_2)^T d} \quad . \tag{2.29}$$

<u>Proof</u>: See Gourgeon and Nocedal [11] .

2.2 Exact Line Search for Conics

The algorithms based on conics, to be presented in Chapters IV and V, will require a one-dimensional minimization at every step. So here we will determine which value of t gives the exact minimum on the line along d passing through x_{k-1} at the k^{th} step. First of all we consider the restriction of f to the line

$$\begin{aligned} x(t) &= x_{k-1} + (\bar{x} - x_{k-1})t \\ &= (1-t)x_{k-1} + t\bar{x} . \end{aligned} \tag{2.31}$$

Here
$$\bar{x} = x_{k-1} + \bar{t}d \tag{2.32}$$

is a point along that line chosen for consistency with notation to appear later. We also define

$$s(t) = (1-t)s_{k-1} + t\bar{s} \tag{2.33}$$

where
$$\bar{s} = \bar{x} - x_o \tag{2.34}$$

and x_o is the reference point. Also, since γ is an affine function,

$$\begin{aligned} \gamma(x(t)) &= (1-t)\gamma(x_{k-1}) + t\gamma(\bar{x}) \\ &= t_1\gamma_{k-1} + t\bar{\gamma} \end{aligned} \tag{2.35}$$

where $t_1 \equiv 1-t$. By using the algebraic representation (2.2) for a conic, on the line $x(t)$ we get

$$f(x(t)) = f_o + \frac{g_o^T s(t)}{\gamma(x(t))} + \frac{1}{2}\,\frac{s(t)^T A_o s(t)}{\gamma^2(x(t))} .$$

We will write γ for $\gamma(t)$ and by substituting (2.33) into this expression, we have

$$f(x(t)) = f_o + \frac{(t_1 g_o^T s_{k-1} + t g_o^T \bar{s})}{\gamma} + \frac{1}{2} \frac{(t_1^2 s_{k-1}^T A_o s_{k-1} + t^2 \bar{s}^T A_o \bar{s})}{\gamma^2} + \frac{t_1 t\, s_{k-1}^T A_o \bar{s}}{\gamma^2} . \quad (2.36)$$

There are several relations which will be used to reduce the form (2.36). These are

$$t_1 t \gamma_{k-1} \bar{\gamma} p_{12} = \frac{1}{2} t_1 t \left[\frac{\bar{\gamma}}{\gamma_{k-1}} s_{k-1}^T A_o s_{k-1} + \frac{\gamma_{k-1}}{\bar{\gamma}} \bar{s}^T A_o \bar{s} \right] - t_1 t (s_{k-1}^T A_o \bar{s}) \quad (2.37)$$

$$\frac{t_1 t \bar{\gamma}}{\gamma_{k-1}} = t_1 \left[\frac{\gamma}{\gamma_{k-1}} - t_1 \right] \quad (2.38)$$

$$\frac{t_1 t \gamma_{k-1}}{\bar{\gamma}} = t \left[\frac{\gamma}{\bar{\gamma}} - t \right] \quad (2.39)$$

$$\frac{1}{2} \frac{(t_1^2 s_{k-1}^T A_o s_{k-1} + t^2 \bar{s}^T A_o \bar{s})}{\gamma^2} + t_1 t \frac{(s_{k-1}^T A_o \bar{s})}{\gamma^2}$$

$$= \frac{1}{2} \left(\frac{t_1}{\gamma_{k-1} \gamma} s_{k-1}^T A_o s_{k-1} + \frac{t}{\bar{\gamma} \gamma} \bar{s}^T A_o \bar{s} \right) - \frac{t_1 t \gamma_{k-1} \bar{\gamma} p_{12}}{\gamma^2} , \quad (2.40)$$

where $p_{12} = \frac{1}{2} \left[\frac{s_{k-1}}{\gamma_{k-1}} - \frac{\bar{s}}{\bar{\gamma}} \right]^T A_o \left[\frac{s_{k-1}}{\gamma_{k-1}} - \frac{\bar{s}}{\bar{\gamma}} \right]$, from (2.25). The first relation (2.37) follows by multiplying both sides of (2.25) by $t_1 t \gamma_{k-1} \bar{\gamma}$. The next two relations (2.38) and (2.39) follow directly by using (2.35). To prove the last relation (2.40) we substitute (2.38) and (2.39) into (2.37). Indeed

$$t_1 t \gamma_{k-1} \bar{\gamma} p_{12} = \frac{1}{2} \left[\frac{t_1 t \bar{\gamma}}{\gamma_{k-1}} s_{k-1}^T A_o s_{k-1} + \frac{t_1 t \gamma_{k-1}}{\bar{\gamma}} \bar{s}^T A_o \bar{s} \right] - t_1 t (s_{k-1}^T A_o \bar{s})$$

$$= \frac{1}{2} \left[\frac{t_1 \gamma}{\gamma_{k-1}} s_{k-1}^T A_o s_{k-1} - t_1^2 s_{k-1}^T A_o s_{k-1} + \frac{t\gamma}{\bar{\gamma}} \bar{s}^T A_o \bar{s} - t^2 \bar{s}^T A_o \bar{s} \right] - t_1 t (s_{k-1}^T A_o \bar{s}) ,$$

and by dividing both sides by γ^2, we get (2.40). Next by using the relations (2.35) to (2.40) and (2.2) for x_{k-1} and $\bar{x}$ we can simplify (2.36) to get an explicit conic in t :

$$f(x(t)) = f_o + \frac{(t_1 g_o^T s_{k-1} + t g_o^T \bar{s})}{\gamma} + \frac{1}{2}\left[\frac{t_1}{\gamma_{k-1}\gamma} s_{k-1}^T A_o s_{k-1} + \frac{t}{\bar{\gamma}\gamma}\bar{s}^T A_o \bar{s}\right]$$

$$- \frac{t_1 t \gamma_{k-1}\bar{\gamma} p_{12}}{\gamma^2}$$

$$= \frac{f_o\gamma + t_1 g_o^T s_{k-1} + t g_o^T \bar{s} + \frac{1}{2}\frac{t_1}{\gamma_{k-1}} s_{k-1}^T A_o s_{k-1} + \frac{1}{2}\frac{t}{\bar{\gamma}}\bar{s}^T A_o \bar{s}}{\gamma}$$

$$- \frac{t_1 t \gamma_{k-1}\bar{\gamma} p_{12}}{\gamma^2}$$

$$= \frac{f_o t_1 \gamma_{k-1} + f_o t\bar{\gamma} + t_1 g_o^T s_{k-1} + t g_o^T \bar{s} + \frac{1}{2}\frac{t_1}{\gamma_{k-1}} s_{k-1}^T A_o s_{k-1} + \frac{1}{2}\frac{t}{\bar{\gamma}}\bar{s}^T A_o \bar{s}}{\gamma}$$

$$- \frac{t_1 t \gamma_{k-1}\bar{\gamma} p_{12}}{\gamma^2}$$

$$= \frac{t_1\gamma_{k-1}\left(f_o + \frac{g_o^T s_{k-1}}{\gamma_{k-1}} + \frac{1}{2}\frac{s_{k-1}^T A_o s_{k-1}}{\gamma_{k-1}^2}\right) + t\bar{\gamma}\left(f_o + \frac{g_o^T \bar{s}}{\bar{\gamma}} + \frac{1}{2}\frac{\bar{s}^T A_o \bar{s}}{\bar{\gamma}^2}\right)}{\gamma}$$

$$- \frac{t_1 t \gamma_{k-1}\bar{\gamma} p_{12}}{\gamma^2}$$

$$= \frac{(1-t)\gamma_{k-1} f_{k-1} + t\bar{\gamma}\bar{f}}{\gamma} - \frac{(1-t)t\gamma_{k-1}\bar{\gamma} p_{12}}{\gamma^2}, \qquad (2.41)$$

where $\bar{f} \equiv f(\bar{x})$ and $\gamma = \gamma(t)$. Differentiating (2.41) and using (2.35), and eliminating p_{12} with (2.26) where we replace x_i by x_{k-1} and x_j by $\bar{x}$ and by recalling that $\bar{r}_{k-1} \equiv \bar{\gamma}/\gamma_{k-1}$ we obtain

$$\frac{df(x(t))}{dt} = \frac{[(1-t)g_{k-1} + t\bar{r}_{k-1}^{-3}\bar{g}]^T}{(1 - t + \bar{r}_{k-1}t)^3}(\bar{s} - s_{k-1}) \quad (2.42)$$

From (2.42) we can see that the derivative vanishes at

$$t^* = \frac{-g_{k-1}^T(\bar{s} - s_{k-1})}{(\bar{r}_{k-1}^{-3}\bar{g} - g_{k-1})^T(\bar{s} - s_{k-1})} \quad (2.43)$$

or, by using (2.32) and (2.34) at

$$t^* = \frac{-g_{k-1}^T d}{(\bar{r}_{k-1}^{-3}\bar{g} - g_{k-1})^T d} \quad (2.44)$$

Therefore by using (2.31) the step length t_k from x_{k-1} to the exact minimum x_k will be

$$t_k = \bar{t}\,t^*$$

$$= \frac{-\bar{t}g_{k-1}^T d}{(\bar{r}_{k-1}^{-3}\bar{g} - g_{k-1})^T d} \quad (2.45)$$

The last formula (2.45) will be used in Chapters IV and V where we state the algorithms based on conics.

CHAPTER III

MULTIPLE UPDATES: THE VSCG ALGORITHM

3.1 Conjugate Gradient Methods

The purpose of this chapter is to summarize the conjugate direction algorithms based on quadratics for reference in the remaining chapters. In this section we will describe the conjugate gradient (CG) method based on quadratic models which was first introduced by Fletcher and Reeves [10] in 1964. We will describe the modification of the CG algorithm known as the preconditioned conjugate gradient method (PCG), in which we are in fact more interested.

First of all, in order to avoid confusion with notation, we have to make some observations. A standard notation encountered in many publications for the algorithms based on quadratic models is that x denotes a point, d is the search direction, s is the step between two consecutive points, $g(x)$ is the gradient value at x and y denotes the difference of the gradients at two consecutive points. But, since we use similar but conflicting notation to discuss the algorithms based on conics in order to be consistent with the notation of Gourgeon and Nocedal [see section 4.4], we will here replace d by v, x by w and g by h. Note that y is unchanged, because if we reduce a conic function to a quadratic, then all γ's become equal to one. Also the step s is unchanged in our notation since we think that will not cause confusion in the notation. Thus, for example, the following formula for the search direction of the CG algorithm,

$$d_k = -g_k + \frac{g_k^T y_k}{d_{k-1}^T y_k} d_{k-1} ,$$

is replaced according to the above discussion by

$$v_k = -h_k + \frac{h_k^T y_k}{v_{k-1}^T y_k} v_{k-1} .$$

We will now introduce the CG algorithm. First we will give an important definition.

Definition 3.1 Given a symmetric positive definite matrix Q, the finite set of non-zero vectors $v_1, v_2, \ldots, v_k$ is said to be conjugate if

$$v_i^T Q v_j = 0 \quad \text{for all} \quad i \neq j .$$

The importance of conjugate vectors is given by the following theorem, known as the Expanding Subspace Theorem (EST):

Theorem 3.1 (EST) Let v_i, $i = 1,2,\ldots,n$ be a set of conjugate vectors in $\mathbb{R}^n$ and let Z_k be the subspace of $\mathbb{R}^n$ spanned by $v_1, v_2, \ldots, v_k$. Then for any $w_0 \in \mathbb{R}^n$ the sequence $\{w_k\}$ generated according to

$$w_k = w_{k-1} + \mu_k v_{k-1} ,$$

has the property that w_k minimizes the quadratic function q on the linear variety $w_0 + Z_k$, provided all line searches are exact.

Proof See Luenbeger [13] .

A consequence of the EST is that, since w_n minimizes the quadratic function q over $w_0 + Z_n$, the global minimum w^* of q will be found in at most n steps, a property which is known as finite termination. The CG algorithm is based on a successive construction of conjugate search directions by using the Gram-Schmidt process (see Luenberger [13]). Thus the CG algorithm has the following form: Given w_o, define $v_o = -h_o$ and for $k = 1,2,\dots$ iterate with

$$w_k = w_{k-1} + \mu_k v_{k-1} \quad , \tag{3.1a}$$

$$\beta_k = \frac{h_k^T y_k}{v_{k-1}^T y_k} \quad , \tag{3.1b}$$

$$v_k = -h_k + \beta_k v_{k-1} \quad . \tag{3.1c}$$

The formula (3.1b) is known as the ~~Hestenes~~ and Stiefel [12] form of β_k.

The CG algorithm has the following orthogonality properties:

$$v_i^T h_j = 0 \qquad 0 \le i < j \quad ; \tag{3.2}$$

$$(w_i - w_o)^T h_j = 0 \qquad 0 \le i \le j \quad ; \tag{3.3}$$

$$h_i^T h_j = 0 \qquad 0 \le i < j \quad . \tag{3.4}$$

The first relation (3.2) follows by applying the EST. In fact, we note that after j line searches along conjugate directions, we will find the minimum of q in the hyperplane spanned by $v_1, v_2, \dots, v_j$. Therefore h_j must be orthogonal to the hyperplane, i.e. equation (3.2) holds. The second equation is obviously true since $w_i - w_o = \sum_{j=1}^{i} \mu_j v_{j-1}$. Also, since the directions v_i, $i = 0,1,\dots,j-1$ were constructed as a linear combination of h_i, $i = 0,1,2,\dots,j-1$, then by using (3.2) the relation (3.4) is true. Note also that the CG algorithm (3.1) generates downhill directions even when it is applied to a general function, provided that ELS are used, since

$$v_k^T h_k = -h_k^T h_k + \beta_k v_{k-1}^T h_k = -\|h_k\|^2 < 0 \ .$$

Finally, we will describe the PCG algorithm. It was first introduced by Axelsson [1] in 1974 and was derived by transforming the variables of the CG method, such that $\bar{w} = H^{\frac{1}{2}}w$ where H is a symmetric positive definite matrix. Then by first applying the CG algorithm in the new $\bar{w}$ - coordinates and then transforming the resulting steps back into the original w - coordinates we obtain the PCG algorithm. In particular, the standard CG algorithm (3.1) in the $\bar{w}$ - coordinates is: Given $\bar{w}_o$, define $\bar{v}_o = -\bar{h}_o$, and for $k = 1,2,\ldots,$ iterate with

$$\bar{w}_k = \bar{w}_{k-1} + \bar{\mu}_k\bar{v}_{k-1} , \tag{3.5a}$$

$$\bar{\beta}_k = \frac{\bar{h}_k^T\bar{y}_k}{\bar{v}_{k-1}^T\bar{y}_k} , \tag{3.5b}$$

$$\bar{v}_k = -\bar{h}_k + \bar{\beta}_k\bar{v}_{k-1} , \tag{3.5c}$$

where $\bar{h} \equiv h(\bar{w})$, $\bar{y}_k = \bar{h}_k - \bar{h}_{k-1}$. Now, in order to transform each step (3.5) into the w - coordinates, the relation between the gradients $\bar{h}$ and h is required. This can be easily derived and is given by

$$\bar{h}(w) = H^{\frac{1}{2}}h(w) \quad \text{and} \quad \bar{y} = H^{\frac{1}{2}}y . \tag{3.6}$$

According to this and since $\bar{v}_k = H^{-\frac{1}{2}}v_k$, the inner product $\bar{v}_k^T\bar{h}_k$ becomes

$$\begin{aligned} \bar{v}_k^T\bar{h}_k &= (H^{-\frac{1}{2}}v_k)^T(H^{\frac{1}{2}}h_k) \\ &= v_k^TH^{-\frac{1}{2}}H^{\frac{1}{2}}h_k = v_k^Th_k . \end{aligned} \tag{3.7}$$

The relation (3.7) implies that the ELS are the same in both coordinate systems, i.e. if both use ELS then the same point is reached in either set of coordinates. Substituting now (3.6) and (3.7) into (3.5b) and (3.5c) we get

$$\bar{\beta}_k = \frac{h_k^T H y_k}{v_{k-1}^T y_k} ,$$

$$v_k = -Hh_k + \bar{\beta}_k v_{k-1} .$$

Finally the PCG algorithm can be summarized. Given w_o and a positive definite matrix H, let $v_o = -Hh_o$ and for $k = 1,2,\ldots$ iterate with

$$w_k = w_{k-1} + \mu_k v_{k-1} , \tag{3.8a}$$

$$\bar{\beta}_k = \frac{h_k^T H y_k}{v_{k-1}^T y_k} , \tag{3.8b}$$

$$v_k = -Hh_k + \bar{\beta}_k v_{k-1} . \tag{3.8c}$$

Note that with $H = I$, we obtain the regular CG algorithm (3.1). Since the PCG algorithm is just the regular CG algorithm in the $\bar{w}$-coordinates, it also has the finite termination property. We can also easily derive orthogonality properties which are similar to those of the CG algorithm:

$$v_i^T h_j = 0 , \qquad 0 \le i < j ; \tag{3.9}$$

$$(w_i - w_o)^T h_j = 0 , \qquad 1 \le i \le j ; \tag{3.10}$$

$$h_i^T H h_j = 0 , \qquad 0 \le i < j . \tag{3.11}$$

Finally, the PCG algorithm (3.8) also generates downhill directions if ELS are used since H is positive definite, for

$$v_k^T h_k = -h_k^T H h_k + \bar{\beta}_k \bar{v}_{k-1}^T h_k = -h_k^T H h_k < 0 .$$

3.2 Quasi-Newton Methods

The QN methods were first described by Davidon [5] in 1959 when he introduced the basic ideas of the DFP algorithm; these were later developed into the DFP algorithm by Feltcher and Powell [9] in 1963. Later again, Broyden [2] in 1967 generalized the DFP algorithm to a family of algorithms which is usually referred to as the Broyden β-class. This section uses the same notation which was introduced in Section 3.1.

The basic idea of the QN methods is to update and use an approximation H_k to the inverse Hessian at x_k in place of the true inverse that is required in Newton's method [see Fletcher [8]], which is often impractical to compute. The form of the approximation varies depending on the particular algorithm. However the so-called "quasi-Newton equation" (or "secant equation")

$$H_k y_k = s_k \ , \tag{3.12}$$

must always hold; we recall that $s_k = w_k - w_{k-1}$. Note that equation (3.12) is exactly satisfied if we replace H_k by the inverse Hessian of a quadratic function q .

We will restrict our attention to the Broyden family of algorithms, and in particular to the most important member of this class which was suggested by Broyden (1970), Fletcher (1970), Goldfarb (1970) and Shanno (1970), and is known as the BFGS algorithm. The BFGS update formula has the following form: Given H , define

$$D(H,i) \equiv H - \frac{Hys^T + sy^T H}{s^T y} + \left[1 + \frac{y^T H y}{s^T y} \right] \frac{ss^T}{s^T y} \ . \tag{3.13a}$$

In the quadratic case, by using the secant equation $Qs = y$, we get

$$D(H,i) \equiv H - \frac{HQss^T + ss^TQH}{s^TQs} + \left(1 + \frac{s^TQHQs}{s^TQs}\right)\frac{ss^T}{s^TQs} , \tag{3.13b}$$

a form necessary for Section 5.1. We can also derive an update formula for the inerse matrix H. If H^{-1} is denoted by B, then it can be verified that the inverse of the update D is given by

$$\bar{D}(B,i) \equiv B - \frac{Bss^TB}{s^TBs} + \frac{Qss^TQ}{s^TQs} , \tag{3.14}$$

by easily establishing that $D(H,i)\bar{D}(H,i) = I$.

We now give a single iteration of the BFGS algorithm. Given is w_o and a positive definite matrix H_o (an initial approximation to the inverse Hessian). For $k = 0,1,2,\ldots,$ iterate with

$$v_k = -H_k h_k \tag{3.15a}$$

$$w_{k+1} = w_k + \mu_{k+1} v_k \tag{3.15b}$$

$$H_{k+1} = D(H_k, k+1) . \tag{3.15c}$$

The BFGS algorithm has some important properties (which are also satisfied by the other members of the Broyden β-class). First, if we apply the BFGS algorithm to a quadratic function q and ELS are used, then it terminates in at most n iterations, with $H_n = Q^{-1}$ (unless termination is permature) [see Fletcher [8]]. Under the same assumptions, the BFGS method (3.15) generates conjugate directions, and orthogonal gradients if $H_o = I$. Finally, if $s_k^T y_k > 0$ for all k, then the BFGS formula (3.15c) generates positive matrices H_k, if H_o is positive definite.

We must prove this property since we will refer to the proof in

Section 4.4. Obviously, it is sufficient to show that the update $\bar{D}(B,i)$ generates positive definite matrices B_k if $B_o \equiv H_o^{-1}$ is positive definite. The proof is inductive and shows directly that $z^T B_k z > 0$ for all $z \neq 0$ such that $z \in \mathbb{R}^n$. The result is true for B_o by choice. Assume it is true for some $k-1 > 0$. Then for any $z \in \mathbb{R}^n$, by using (3.14) we obtain

$$z^T B_k z = z^T B z - \frac{(z^T Bs)(s^T Bz)}{s^T Bs} + \frac{(z^T Qs)(s^T Qz)}{s^T Qs}$$

$$= \frac{(z^T Bz)(s^T Bs) - (z^T Bs)^2}{s^T Bs} + \frac{(z^T Qs)^2}{s^T Qs} ,$$

where $B \equiv B_{k-1}$ and $s \equiv s_k$. From this we can see that it is sufficient to show

$$(z^T Bz)(s^T Bs) - (z^T Bs)^2 > 0 , \tag{3.16a}$$

or $$z^T Qs \neq 0 , \tag{3.16b}$$

since $s^T Bs > 0$ by assumption. To prove (3.16a) we use Cauchy's inequality,

$$\|u\|^2 \|\bar{u}\|^2 - (u^T \bar{u})^2 \geq 0 \tag{3.17}$$

for any $u, \bar{u} \in \mathbb{R}^n$, with $u = B^{\frac{1}{2}} z$, $\bar{u} = B^{\frac{1}{2}} s$. Equality holds only when z is a multiple of s. Then (3.16b) holds. This completes the induction and consequently our proof.

Finally we observe according to the last property that any value of μ_k chosen so that $s_k^T y_k > 0$ will guarantee positive definiteness in the next update, and hence a downhill direction v_k given by (3.15a) if ELS are used.

3.3 Combined CG and QN Methods

The purpose of this section is to demonstrate the relationship that exists between the PCG algirothm and the BFGS method, in order to derive, in the next section, the VSCG algorithm. In particular, we will show how the PCG algorithm can be written in the same manner as the BFGS method.

To derive such a relation we assume as usual that the above methods are applied to a quadratic function q and ELS are used, i.e. $s_i^T h_i = 0$ for all $i = 1,2,\ldots$. First we recall the BFGS update formula (3.15c),

$$H_k = H_{k-1} - \frac{H_{k-1} y s^T + s y^T H_{k-1}}{s^T y} + \left(1 + \frac{y^T H_{k-1} y}{s^T y}\right) \frac{s s^T}{s^T y} ,$$

where again the missing subscripts are k . By substituting this formula into (3.15a), and taking into consideration that ELS are used, we obtain

$$\begin{aligned} v_k &= - H_k h_k \\ &= - H_{k-1} h_k + \frac{H_{k-1} y (s^T h_k) + s (y^T H_{k-1} h_k)}{s^T y} - \left(1 + \frac{y^T H_{k-1} y}{s^T y}\right) \frac{s(s^T h_k)}{s^T y} \\ &= - H_{k-1} h_k + \left[\frac{y^T H_{k-1} h_k}{s^T y}\right] s , \\ &= - H_{k-1} h_k + \left[\frac{y^T H_{k-1} h_k}{v_{k-1}^T y}\right] v_{k-1} , \end{aligned} \qquad (3.18)$$

since $s \equiv s_k = \mu_k v_{k-1}$. By comparing (3.18) and the search direction (3.8c) for the PCG algorithm, we can see that both formulas are the same if we choose $H_{k-1} \equiv H$, i.e. if $H_i = D(H,i)$ for each

$i = 1,2,\ldots$.

Hence, we observe that under the assumption that ELS are used the PCG method can be interpreted as a BFGS method, in which a fixed preconditioning matrix H is updated at each step.

3.4 The VSCG Algorithm

To complete this chapter, we will give a brief description of the variable storage conjugate gradient algorithm (VSCG) which was introduced by Buckley and LeNir [4] in 1982, since a main purpose in this thesis is to derive a similar algorithm but based on conics. The VSCG algorithm has been derived in such a way as to combine the CG and QN algorithms and to obtain one with better performance.

Suppose that we choose a starting point w_o and a positive definite matrix H_o. Then the VSCG algorithm is given by two parts as follows.

QN-part: For $i = 1,\ldots,m$ iterate with

$$H_i = D(H_{i-1},i) \quad , \tag{3.19a}$$

$$v_i = -H_i h_i \quad , \tag{3.19b}$$

$$w_i = w_{i-1} + \mu_i v_{i-1} \quad .$$

CG-part: From the point w_{m+1} reached by the QN-part, and using the fixed matrix H_m as preconditioner, iterate for $i = m+1, m+2, \ldots$:

$$H_i = D(H_m,i) \quad , \tag{3.20a}$$

$$v_i = -H_i h_i \quad , \tag{3.20b}$$

$$w_i = w_{i-1} + \mu_i v_{i-1} \quad .$$

By way of comment on the VSCG algorithm, we can observe that the two parts are different only in the definition of H_i. In the QN-part (for the first m steps), we update H_i from the previous matrix H_{i-1}. Then in the CG-part, by using the fixed matrix H_m, we always update H_i from H_m. Therefore the CG-part can be viewed as the PCG algorithm in the form described in Section 3.3; here the preconditioner is H_m .

There are some important properties of the VSCG algorithm. First, according to the observations which we made in Section 3.2, the update formulas (3.19a) and (3.20) always produce positive definite matrices, for in practice one can ensure that $w_i^T y_i > 0$. Then it is clear from (3.19b) and (3.20b) that all the directions v_i are descent directions. Also, when f is quadratic and ELS are used, according to Theorem 3 of Buckley in [3], finite termination is obtained in at most n steps, counting from the very beginning. Finally, we note that the matrices H_i are not themselves needed, but only products $H_i u$ for $u \in \mathbb{R}^n$ are required. Therefore according to the derivation of Buckley and LeNir in [4], we don't have to store each matrix H_i but only some vectors, so that the total storage needed for $H_1, \ldots, H_m$ is $m(2n+2)$ locations, which is explained in the context of conics in Section 5.2. Provided $m \ll n$, this storage is just a multiple of n.

In conclusion, the VSCG algorithm does not require one to store a matrix, but only certain vectors, and it can use a variable amount of storage. In terms of both performance and structure we can classify it between the CG and QN methods.

CHAPTER IV

CONIC APPROXIMATIONS

4.1 Basic Relations

In this section we state and prove some basic relations which are required to describe and to investigate the $O(n)$ and $O(n^2)$ conjugate direction algorithms. We will also introduce more of our notation.

First we define the $n \times n$ matrix $G(x)$ given by

$$G(x) \equiv \frac{1}{\gamma(x)}\left[I + \frac{s a_o^T}{\gamma(x)} \right], \tag{4.1a}$$

where $s = x - x_o$ and a_o is the horizon vector corresponding to the reference point x_o. For brevity, when $x = x_k$, it will be written

$$G_k \equiv G(x_k) = \frac{1}{\gamma_k}\left[I + \frac{s_k a_o^T}{\gamma_k} \right]. \tag{4.1b}$$

Since it is always assumed that $\gamma_k \neq 0$ G_k^{-1} exists and is given by

$$G_k^{-1} = \gamma_k [I - s_k a_o^T] . \tag{4.2}$$

The reason for this choice of the matrix G will become clear in section (4.3), where we will describe the (GN) algorithm. We also define

$$L_k \equiv \frac{1}{r_+}\left[I + \frac{t_k d_{k-1} a_o^T}{\gamma_k} \right] = \frac{1}{r_+}\left[I + t_k d_{k-1} a_k^T \right], \tag{4.3}$$

where $$r_+ \equiv \gamma_{k-1}(x_k) = \frac{\gamma(x_k)}{\gamma(x_{k-1})} = \frac{\gamma_k}{\gamma_{k-1}} . \tag{4.4}$$

Next, we will first state and then prove some relations between the matrices $G(x)$, the conjugacy matrix A_o, the displacements s, and the gradients $g(x)$, for we will need to refer to these relations later.

$$G_k = G_{k-1}L_k \tag{4.5}$$

$$G_k s_j = \frac{s_j}{\gamma_k} + \frac{(1-\gamma_j)s_k}{\gamma_k^2} \tag{4.6}$$

$$G_k^{-1}s_k = \gamma_k^2 s_k \tag{4.7}$$

$$G_{k-1}d_{k-1} = \frac{r_+}{t_k}\left(\frac{s_k}{\gamma_k} - \frac{s_{k-1}}{\gamma_{k-1}}\right) \tag{4.8}$$

$$G_k d_{k-1} = \frac{1}{t_k r_+}\left(\frac{s_k}{\gamma_k} - \frac{s_{k-1}}{\gamma_{k-1}}\right) \tag{4.9}$$

$$r_+^2 G_k d_{k-1} = G_{k-1}d_{k-1} \tag{4.10}$$

$$g_k = G_k^T\left(g_o + \frac{A_o s_k}{\gamma_k}\right) \tag{4.11}$$

$$g_k = L_k^T g_{k-1} + \frac{t_k}{r_+} G_k^T A_o G_{k-1} d_{k-1} \tag{4.12}$$

$$A_o\left(\frac{s_k}{\gamma_k} - \frac{s_{k-1}}{\gamma_{k-1}}\right) = G_k^{-T} g_k - G_{k-1}^{-T} g_{k-1} \tag{4.13}$$

$$A_o G_k d_{k-1} = \frac{1}{t_k r_+}(G_k^{-T} g_k - G_{k-1}^{-T} g_{k-1}) \tag{4.14}$$

$$s_k = \sum_{i=o}^{k-1} \theta_i (G_i d_i) \quad \text{for suitable } \theta_i \;. \tag{4.15}$$

The proofs of the above relations require primarily direct manipulation and the basic relations:

$$a_o^T s_i = 1 - \gamma_i \quad , \tag{4.16a}$$

$$s_i - s_{i-1} = t_i d_{i-1} \quad , \tag{4.16b}$$

which are true for any $i = 1,2,\ldots$. The first equation, (4.5) relates the matrix G_k with the previous G_{k-1} and the proof follows by using (4.16). In particular

$$G_{k-1}L_k = \frac{1}{\gamma_{k-1}}\left[I + \frac{s_{k-1}a_o^T}{\gamma_{k-1}}\right]\frac{1}{r_+}\left[I + \frac{t_k d_{k-1} a_o^T}{\gamma_k}\right]$$

$$= \frac{1}{\gamma_{k-1}} \cdot \frac{\gamma_{k-1}}{\gamma_k}\left[I + \frac{s_{k-1}a_o^T}{\gamma_{k-1}}\right]\left[I + \frac{(s_k - s_{k-1})a_o^T}{\gamma_k}\right]$$

$$= \frac{1}{\gamma_k}\left[I + \frac{s_{k-1}a_o^T}{\gamma_{k-1}} + \frac{s_k a_o^T}{\gamma_k} - \frac{s_{k-1}a_o^T}{\gamma_k} + \frac{s_{k-1}a_o^T(a_o^T s_k)}{\gamma_{k-1}\gamma_k} - \frac{s_{k-1}a_o^T(a_o^T s_{k-1})}{\gamma_{k-1}\gamma_k}\right]$$

$$= \frac{1}{\gamma_k}\left[I + \frac{s_{k-1}a_o^T}{\gamma_{k-1}} + \frac{s_k a_o^T}{\gamma_k} - \frac{s_{k-1}a_o^T}{\gamma_k} + \frac{s_{k-1}a_o^T(1-\gamma_k)}{\gamma_{k-1}\gamma_k} - \frac{s_{k-1}a_o^T(1-\gamma_{k-1})}{\gamma_{k-1}\gamma_k}\right]$$

$$= \frac{1}{\gamma_k}\left[I + \frac{s_{k-1}a_o^T}{\gamma_{k-1}} + \frac{s_k a_o^T}{\gamma_k} - \frac{s_{k-1}a_o^T}{\gamma_k} + \frac{s_{k-1}a_o^T}{\gamma_{k-1}\gamma_k} - \frac{s_{k-1}a_o^T}{\gamma_{k-1}} - \frac{s_{k-1}a_o^T}{\gamma_{k-1}\gamma_k} + \frac{s_{k-1}a_o^T}{\gamma_k}\right]$$

$$= \frac{1}{\gamma_k}\left[I + \frac{s_k a_o^T}{\gamma_k}\right]$$

$$= G_k .$$

The second relation is also verified by using (4.16a):

$$G_k s_j = \frac{1}{\gamma_k}\left[I + \frac{s_k a_o^T}{\gamma_k}\right] s_j$$

$$= \frac{1}{\gamma_k}\left[s_j + \frac{s_k(a_o^T s_j)}{\gamma_k} \right]$$

$$= \frac{1}{\gamma_k}\left[s_j + \frac{s_k(1-\gamma_j)}{\gamma_k} \right]$$

$$= \frac{s_j}{\gamma_k} + \frac{(1-\gamma_j)s_k}{\gamma_k^2} .$$

By using (4.6) for $j = k$ we get (4.7). To prove (4.8) first we use (4.16b) for $i = k$. Indeed,

$$G_{k-1}d_{k-1} = G_{k-1}\left(\frac{s_k - s_{k-1}}{t_k}\right)$$

$$= \frac{1}{t_k}\left[G_{k-1}s_k - G_{k-1}s_{k-1} \right] .$$

Then by using (4.6) twice for appropriate j we obtain

$$G_{k-1}d_{k-1} = \frac{1}{t_k}\left[\frac{s_k}{\gamma_{k-1}} + \frac{1-\gamma_k}{\gamma_{k-1}^2} s_{k-1} - \frac{1}{\gamma_{k-1}^2} s_{k-1} \right]$$

$$= \frac{1}{t_k}\left[\frac{s_k}{\gamma_{k-1}} - \frac{\gamma_k s_{k-1}}{\gamma_{k-1}^2} \right]$$

$$= \frac{r_+}{t_k}\left(\frac{s_k}{\gamma_k} - \frac{s_{k-1}}{\gamma_{k-1}} \right) .$$

The proof for (4.9) is similar. The relation (4.10) follows directly by combining (4.8) and (4.9). It has been shown in Chapter II that

$$g_k = \frac{1}{\gamma_k}\left(I + \frac{a_o s_k^T}{\gamma_k} \right)\left(g_o + \frac{A_o s_k}{\gamma_k} \right) ;$$

Therefore according to (4.1b),

$$g_k = G_k^T\left(g_o + \frac{A_o s_k}{\gamma_k}\right)$$

which is exactly the relation (4.11). To prove (4.12), first note according to (4.11),

$$g_{k-1} = G_{k-1}^T\left(g_o + \frac{A_o s_{k-1}}{\gamma_{k-1}}\right) .$$

Therefore,

$$g_o = G_{k-1}^{-T} g_{k-1} - \frac{A_o s_{k-1}}{\gamma_{k-1}}$$

Then, by substituting the last equality into (4.11), we have

$$\begin{aligned} g_k &= G_k^T\left[G_{k-1}^{-T} g_{k-1} - \frac{A_o s_{k-1}}{\gamma_{k-1}} + \frac{A_o s_k}{\gamma_k}\right] \\ &= G_k^T\left[G_{k-1}^{-T} g_{k-1} + A_o\left(\frac{s_k}{\gamma_k} - \frac{s_{k-1}}{\gamma_{k-1}}\right)\right] . \end{aligned}$$

Finally by using (4.8) we obtain

$$\begin{aligned} g_k &= G_k^T\left[G_{k-1}^{-T} g_{k-1} + \frac{t_k \gamma_{k-1}}{\gamma_k} A_o G_{k-1} d_{k-1}\right] \\ &= L_k^T g_{k-1} + \frac{t_k}{r_+} G_k^T A_o G_{k-1} d_{k-1} \end{aligned}$$

since $G_k^T G_{k-1}^{-T} = L_k^T$ follows according to (4.5). The next relation (4.13) is similar to the "secand equation" for a quadratic function and the proof follows by using (4.11) twice. Indeed,

$$\begin{aligned} G_k^{-T} g_k - G_{k-1}^{-T} g_{k-1} &= g_o + \frac{A_o s_k}{\gamma_k} - g_o - \frac{A_o s_{k-1}}{\gamma_{k-1}} \\ &= A_o\left(\frac{s_k}{\gamma_k} - \frac{s_{k-1}}{\gamma_{k-1}}\right) . \end{aligned}$$

Relation (4.14) is similar and it is easy to verify it by combining (4.9), (4.10) and (4.13). The last relation (4.15) follows from (4.8) by solving it in terms of s_k first, i.e.

$$s_k = t_k \gamma_{k-1} G_{k-1} d_{k-1} + \frac{\gamma_k}{\gamma_{k-1}} s_{k-1} .$$

Next by using the same formula again for s_{k-1} and substituting in this equality we have,

$$s_k = t_k \gamma_{k-1} G_{k-1} d_{k-1} + \frac{\gamma_k}{\gamma_{k-1}} \left[t_{k-1} \gamma_{k-2} G_{k-2} d_{k-2} + \frac{\gamma_{k-1}}{\gamma_{k-2}} s_{k-2} \right]$$

$$= t_k \gamma_{k-1} G_{k-1} d_{k-1} + \frac{\gamma_k \gamma_{k-2}}{\gamma_{k-1}} t_{k-1} G_{k-2} d_{k-2} + \frac{\gamma_k}{\gamma_{k-2}} s_{k-2} .$$

If we repeat by always substituting for s_i for $i = k-1, k-2, \ldots, 1$, finally we obtain

$$s_k = \theta_{k-1} G_{k-1} d_{k-1} + \theta_{k-2} G_{k-2} d_{k-2} + \ldots + \theta_1 G_1 d_1 + \theta_0 G_0 d_0$$
$$= \sum_{i=0}^{k-1} \theta_i (G_i d_i) ,$$

for suitable constants θ_i. This concludes the proofs of the above basic relations.

However we will now give some more relations which hold only under the assumption that ELS are used. Then the following relations are true.

If $\quad s_j^T g_i = 0$ for $j \le i$, $\quad G_j^{-T} g_i = \gamma_j g_i$ (4.17a)

and $\quad G_j^T g_i = \gamma_j g_i$. (4.17b)

If $\quad d_{k-1}^T g_k = 0$, $\quad L_k^T g_k = \frac{1}{r_+} g_k$. (4.18)

Relation (4.17a) can be easily verified by direct expansion. Indeed,

$$G_j^{-T} g_i = \gamma_j [I - a_o s_j^T] g_i$$

$$= \gamma_j g_i - a_o s_j^T g_i$$

$$= \gamma_j g_i \; .$$

Also (4.17b) follows since G_j^{-T} is invertible. The proof for relation (4.18) is similar.

Finally, some observations will now be discussed which we will use to clarify Davidon's strategy and notation for the $O(n)$ and $O(n^2)$ algorithms.

Davison's basic idea for both the $O(n)$ and $O(n^2)$ algorithms is to use the gradients at three collinear points to determine the horizon vector a and hence the subsequent search direction. Therefore it will require the evaluation of the gradient at three collinear points on all iterations, since the intention is to construct a conic model at each step when we apply these algorithms to a general objective function. Davidon [6] introduced and proved several basic formulas in this case where the three collinear points are given and one of these is selected as the reference point. We will state these formulas since we want to clarify our notation. We now consider the three collinear points to be x_o, $\hat{x}$ and $\tilde{x}$ where x_o is taken as the reference point. First we will define the following: $\cdot$ may be $\hat{}$ or $\tilde{}$,

1. $\dot{g} \equiv g(\dot{x})$
2. $g_o \equiv g(x_o)$
3. $\dot{\gamma} \equiv \gamma(\dot{x})$
4. $\dot{u} \equiv \dfrac{\dot{\gamma}\dot{g} - g_o}{\dot{t}(\hat{\gamma}^2\hat{\sigma} - \tilde{\gamma}^2\tilde{\sigma})}$

5. $\dot{\sigma} = d^T\dot{g}$,

where $\dot{x} = x_o + \dot{t}d$. Then the basic formulas given by Davidon are:

$$a_o = \hat{\gamma}\hat{u} - \tilde{\gamma}\tilde{u} \tag{4.19}$$

$$A_o d = \hat{\gamma}\tilde{\gamma}(\hat{\gamma}\hat{\sigma}\tilde{u} - \tilde{\gamma}\tilde{\sigma}\hat{u}) . \tag{4.20}$$

$$d^T A_o d = \frac{\hat{\gamma}\tilde{\gamma}}{\hat{t} - \tilde{t}} (\hat{\gamma}^2\hat{\sigma} - \tilde{\gamma}^2\tilde{\sigma}) . \tag{4.21}$$

We choose this notation so as not to conflict with other notation for the k^{th} step.

Both algorithms, $0(n)$ and $0(n^2)$, begin their k^{th} iteration with f_{k-1} and g_{k-1} at x_{k-1}, and they use these along with $\bar{f}$ and $\bar{g}$ at a trial point $\bar{x}$ to locate the minimizer x_k on the line through x_{k-1} and $\bar{x}$, i.e.

$$\bar{x} = x_o + \bar{t}d$$

where $\bar{t}$ is the trial step length,

$$x_k = x_o + t_k d ,$$

where t_k is the length which gives the exact minimum on the line d . Then he uses the three gradients g_{k-1}, g_k and $\bar{g}$ at the collinear points x_{k-1}, x_k and $\bar{x}$ to determine the new horizon vector a_k where x_k is taken as the reference point. In order to do that we have to get the equivalent formulas of (4.17), (4.18) and (4.19) by making the following appropriate replacement:

$$x_o \to x_k$$

$$\hat{x} \to x_{k-1} = x_k + (-t_k)d$$

$$\tilde{x} \to \bar{x} = x_k + (\bar{t} - t_k)d ,$$

where " → " means "is replaced by". Also we can see that $\hat{t} = -t_k$ and $\tilde{t} = \bar{t} - t_k$. But since $x_o \to x_k$, $\gamma(x) \to \gamma_k(x)$. Therefore

$$\hat{\gamma} \to \gamma_k(\hat{x}) = \gamma_k(x_{k-1})$$

$$= \frac{\gamma(x_{k-1})}{\gamma(x_k)}$$

$$= \frac{\gamma_{k-1}}{\gamma_k}$$

$$\equiv \frac{1}{r_+} \quad (4.22)$$

Similarly

$$\tilde{\gamma} \to \gamma_k(\bar{x}) = \frac{\gamma(\bar{x})}{\gamma_k(\bar{x})}$$

$$= \frac{\bar{\gamma}}{\gamma_k} . \quad (4.23)$$

Also $g_o \to g_k$, and so we have

$$\hat{\sigma} = d^T\hat{g} \to d^T g_{k-1} = \sigma_{k-1} \quad (4.24)$$

$$\tilde{\sigma} = d^T\tilde{g} \to d^T\bar{g} = \bar{\sigma} \quad (4.25)$$

$$\hat{u} = \frac{\hat{\gamma}\hat{g} - g_o}{\hat{t}(\hat{\gamma}^2\hat{\sigma} - \tilde{\gamma}^2\tilde{\sigma})} \to \frac{\frac{\gamma_{k-1}}{\gamma_k} g_{k-1} - g_k}{(-t_k)\left[\frac{\gamma_{k-1}^2}{\gamma_k^2}\sigma_{k-1} - \frac{\bar{\gamma}^2}{\gamma_k^2}\bar{\sigma}\right]} = \frac{\gamma_k}{\gamma_{k-1}} \cdot \frac{g_{k-1} - r_+ g_k}{(-t_k)(\sigma_{k-1} - \bar{r}_{k-1}\bar{\sigma})}$$

$$\equiv r_+ u_{k-1} . \quad (4.26)$$

In a similar we can find

$$\tilde{u} = \frac{\tilde{\gamma}\tilde{g} - g_o}{\tilde{t}(\hat{\gamma}^2\hat{\sigma} - \tilde{\gamma}^2\tilde{\sigma})} \to \frac{\frac{\gamma_k}{\gamma_{k-1}} \frac{\bar{\gamma}}{\gamma_{k-1}} \bar{g} - r_+ g_k}{(\bar{t} - t_k)(\sigma_{k-1} - \bar{r}_{k-1}\bar{\sigma})}$$

$$= r_+ \frac{\bar{r}_{k-1}\bar{g} - r_+ g_k}{(\bar{t} - t_k)(\sigma_{k-1} - \bar{r}_{k-1}\bar{\sigma})}$$

$$\equiv r_+\bar{u} \quad . \tag{4.27}$$

Finally by substituting the above replacements (4.22) - (4.27) into (4.17), (4.18) and (4.19) we obtain,

$$a_k = \frac{\gamma_{k-1}}{\gamma_k} r_+ u_{k-1} - \frac{\bar{\gamma}}{\gamma_k} r_+ \bar{u}$$

$$= u_{k-1} - \bar{r}_{k-1}\bar{u} \tag{4.28}$$

$$A_k d = \frac{\gamma_{k-1}}{\gamma_k} \cdot \frac{\bar{\gamma}}{\gamma_k}\left[\frac{\gamma_{k-1}}{\gamma_k}\sigma_{k-1}(r_+\bar{u}) - \frac{\bar{\gamma}}{\gamma_k}\bar{\sigma}(r_+ u_{k-1})\right]$$

$$= \frac{\gamma_{k-1}\bar{\gamma}}{\gamma_k^2}(\sigma_{k-1}\bar{u} - \bar{r}_{k-1}\bar{\sigma}u_{k-1}) \tag{4.29}$$

$$d^T A_k d = \frac{\frac{\gamma_{k-1}}{\gamma_k} \cdot \frac{\bar{\gamma}}{\gamma_k}}{-t_k - (\bar{t} - t_k)} \left[\frac{\gamma_{k-1}^2}{\gamma_k^2}\sigma_{k-1} - \frac{\bar{\gamma}^2}{\gamma_k^2}\bar{\sigma}\right]$$

$$= \frac{\gamma_{k-1}\bar{\gamma}}{\bar{t}\,\gamma_k^4}(\bar{\gamma}^2\bar{\sigma} - \gamma_{k-1}^2\sigma_{k-1}) \; . \tag{4.30}$$

As we mentioned at the beginning of this section, the relations and the notations which we have established in this section will be used in the next sections to derive the $0(n)$ and $0(n^2)$ algorithms.

4.2 Davidon's O(n) Algorithm

The algorithm which will now be described can be considered as a generalization of the PCG algorithm (3.1), using a conic approximation to the objective function. It is known as "the O(n) algorithm" and first appeared in Davidon's paper [6] in 1981.

As we mentioned in the previous section (4.1), the basic idea of the O(n) algorithm is to construct a conic model at each step using the current point as the reference point at each iteration. We will now describe a single iteration of this algorithm; and a discussion will follow.

At the beginning of the k^{th} iteration, we assume that the point x_{k-1}, the direction d_{k-1}, the gradient $g_{k-1} \equiv g(x_{k-1})$, the function value $f_{k-1} \equiv f(x_{k-1})$, the gauge value $\gamma_{k-1} \equiv \gamma(x_{k-1})$ and a symmetric, positive definite $n \times n$ matrix H_o are given. We also assume that the first search direction $d_o = -H_o g_o$ where g_o is the gradient at the point x_o, the starting point. For simplicity of our notation we denote by d the direction d_{k-1}, i.e. $d \equiv d_{k-1}$. Initially $k = 1$.

The algorithm O(n):

STEP 1. Evaluate $\sigma_{k-1} = d^T g_{k-1}$

STEP 2. Choose $\bar{t}$ (one possible choice is $\bar{t} = 1$), set $\bar{x} = x_{k-1} + \bar{t}d$ and evaluate $\bar{f} = f(\bar{x})$, $\bar{g} \equiv g(\bar{x})$ and $\bar{\sigma} = d^T\bar{g}$.

STEP 3. Compute first

$$p = [(\bar{f} - f_{k-1})^2 - \sigma_{k-1}\bar{\sigma}\bar{t}^2]^{\frac{1}{2}}$$

and then

$$\bar{r}_{k-1} = \frac{-\bar{t}\,\sigma_{k-1}}{f_{k-1} - \bar{f} + p} .$$

Note that by $\bar{r}_{k-1}$ we denote the ratio $\gamma(\bar{x})/\gamma(x_{k-1})$.

STEP 4. Set

$$t_k = \frac{-\bar{t}\,\sigma_{k-1}}{\bar{r}^3_{k-1}\,\bar{\sigma} - \sigma_{k-1}}$$

and

$$r_+ = \frac{\bar{r}_{k-1}(\bar{r}^2_{k-1}\,\bar{\sigma} - \sigma_{k-1})}{\bar{r}^3_{k-1}\,\bar{\sigma} - \sigma_{k-1}} .$$

After that evaluate $x_k = x_{k-1} + t_k d$, g_k and f_k

STEP 5. Compute

$$u_{k-1} = \frac{g_{k-1} - r_+ g_k}{-t_k(\sigma_{k-1} - \bar{r}_{k-1}\bar{\sigma})} ,$$

$$\bar{u} = \frac{\bar{r}_{k-1}\bar{g} - r_+ g_k}{(\bar{t} - t_k)(\sigma_{k-1} - \bar{r}_{k-1}\bar{\sigma})} .$$

STEP 6. Compute

$$a_k = u_{k-1} - \bar{r}_{k-1}\bar{u} .$$

STEP 7. Compute

$$b_k = \sigma_{k-1}\bar{u} - \bar{r}_{k-1}\bar{\sigma}u_{k-1} . \tag{4.31}$$

STEP 8. Set

$$\gamma_k = \gamma_{k-1} r_+$$

STEP 9. Compute

$$c_k \equiv \gamma_k^2(H_o g_k - \gamma_k s_k a_k^T H_o g_k) \tag{4.32a}$$

and,
$$d_k = -c_k + \frac{b_k^T c_k}{b_k^T d}\, d \tag{4.32b}$$

and then return to step 1.

Note that c_k also reduces to the form

$$c_k = \gamma_k G_k^{-1} H_o g_k \quad . \tag{4.32b}$$

We will now comment on this algorithm. In general the derivation of the $O(n)$ algorithm is based on the observations which we made in section (4.1). The ratio of the gauges at step 3, denoted by $\bar{r}_{k-1}$, is computed according to Theorem 2.1. In the next step, the length t_k gives the exact minimum on the line d, that is $d^T g_k = 0$. The formula for t_k is derived according to the discussion in section (2.2). Relation (4.31), by comparing with (4.29), shows that

$$b_k = \frac{\gamma_k^2}{\gamma_{k-1}\bar{\gamma}} A_k d_{k-1} , \tag{4.33a}$$

and hence

$$b_k^T d_k = \frac{\gamma_k^2}{\gamma_{k-1}\bar{\gamma}} d_k^T A_k d_{k-1} . \tag{4.33b}$$

But by the very choice of d_k, it also follows that

$$b_k^T d_k = 0 ; \tag{4.34}$$

therefore according to (4.33b)

$$d_k^T A_k d_{k-1} = 0 \quad . \tag{4.35}$$

The $O(n)$ conjugate direction algorithm presented here has the following properties in common with previous quadratically based ones (see section (3.1)). First, the k^{th} step $x_k - x_{k-1} = t_k d$ is a linear combination of previous steps and $H_o g_{k-1}$. Second, the line through x_k and x_{k-1} is conjugate [in a sense that it will be discussed later] to one line through x_j and x_{j-1} for all $j < k$. Finally the point

x_k is the minimizer of the restriction of f to the line through x_k and x_{k-1} and, more generally, is the minimizer of the restriction of f to the manifold spanned by all previous directions. We will establish these properties in the next theorem. But first some simple observations will be given which will be used in the proof of the theorem, and which initiate the induction used in applying the theorem.

01: $\quad d_o^T g_1 = 0$.

02: $\quad g_o^T H_o g_1 = 0$.

03: $\quad s_1^T g_1 = 0$.

04: $(G_o d_o)^T A_o (G_1 g_1) = 0$, where A_o is the conjugacy matrix corresponding to the starting point x_o .

05: $\quad s_1^T A_o (G_1 d_1) = 0$.

The first equation (01) is true since it is assumed that ELS are used. By using $d_o = -H_o g_o$ and (01) we get (02). The proof of (03) is similar since $s_1 = t_1 d_o$. To prove (04) first note according to (4.10) for $k = 1$,

$$r_+ G_1 d_o = G_o d_o \quad .$$

Then we have

$$\begin{aligned}(G_o d_o)^T A_o (G_1 d_1) &= r_+ (G_1 d_o)^T A_o (G_1 d_1) \\ &= r_+ d_o^T (G_1^T A_o G_1) d_1 \\ &= r_+ d_o^T A_1 d_1 ,\end{aligned}$$

since $G_1^T A_o G_1 = A_1$ according to (2.19) . But $d_o^T A_1 d_1 = 0$ by choice of d_1 , as noted in (4.35). Therefore (04) is true. The last equation (05) follows by combining (4.15) and (04). Now we can state our main

Theorem.

Theorem 4.2: Suppose $k > 1$ and $x_i \neq x^*$ for $i < k$, where x^* is the global minimum of f. Assume

A1: $s_j^T g_i = 0$, $1 \leq j \leq i < k$;

A2: $d_j^T g_i = 0$, $0 \leq j < i < k$;

A3: $g_j^T H_o g_i = 0$, $0 \leq j < i < k$;

A4: $(G_j d_j)^T A_o (G_i d_i) = 0$, $0 \leq j < i < k$;

A5: $s_j^T A_o (G_i d_i) = 0$, $1 \leq j \leq i < k$,

and that we have just reached x_k by an ELS along d_{k-1}. Then either $g_k = 0$ (so $x_k = x^*$) or A1 to A5 hold with k replaced by $k+1$.

Proof: First of all we can see, according to our assumption, that it is sufficient to show that A1 to A5 hold with $i = k$. In particular

S1: $s_j^T g_k = 0$, $j \leq k$;

S2: $d_j^T g_k = 0$, $j < k$;

S3: $g_j^T H_o g_k = 0$, $j < k$;

S4: $(G_j d_j)^T A_o (G_k d_k) = 0$, $j < k$;

S5: $s_j^T A_o (G_k d_k) = 0$, $j \leq k$.

First, we will state and prove some relations which hold only under the theorem's assumptions. The reason is that we will need to refer to these in the course of our proof. Note that the relations R7 to R10 are the most important. Here G_k serves as an intermediary in R3 and R4 to get the desired results.

R1. $d_{k-1}^T g_k = 0$.

R2. $g_k^T s_k = g_k^T s_{k-1}$.

R3. $g_k^T G_k^{-1} s_{k-1} = \gamma_k \gamma_{k-1} g_k^T s_k$.

R4. $g_k^T G_k^{-1} s_k = \gamma_k^2 g_k^T s_k$.

R5. $g_k^T G_k^{-1} \left(\frac{s_k}{\gamma_k} - \frac{s_{k-1}}{\gamma_{k-1}}\right) = 0$.

R6. $g_k^T G_k^{-1} s_{k-1} = 0$.

R7. $g_k^T s_k = 0$.

R8. $g_k^T s_{k-1} = 0$.

R9. $g_k^T G_k^{-1} s_j = 0, \quad j < k-1$.

R10. $(G_{k-1} d_{k-1})^T A_o (G_k d_k) = 0$.

The first equation is obvious since ELS are used. By combining (4.16b) for $i = k$ and (R1) we get (R2). The relation (R3) follows by using the definition of G_k^{-1}, (4.16a) for $i = k-1$ and (R2):

$$
\begin{aligned}
g_k^T G_k^{-1} s_{k-1} &= \gamma_k g_k^T [I - s_k a_o^T] s_{k-1} \\
&= \gamma_k g_k^T [s_{k-1} - s_k (a_o^T s_{k-1})] \\
&= \gamma_k [g_k^T s_{k-1} - g_k^T s_k (1 - \gamma_{k-1})] \\
&= \gamma_k [g_k^T s_k - g_k^T s_k (1 - \gamma_{k-1})] \\
&= \gamma_k \gamma_{k-1} g_k^T s_k .
\end{aligned}
$$

By using (4.7) we get (R4). The relation (R5) follows by using (R3) and (R4). To prove (R6) we substitute first in terms of g_k according to (4.12), and then use (4.5) and (4.17a):

$$\begin{aligned}
g_k^T G_k^{-1} s_{k-1} &= (G_k^{-T}\ g_k)^T s_{k-1} \\
&= (G_k^{-T} L_k^T g_{k-1} + \frac{t_k}{r_+} G_k^{-T}\ G_k A_o G_{k-1} d_{k-1})^T s_{k-1} \\
&= (G_{k-1}^{-T} g_{k-1} + \frac{t_k}{r_+} A_o G_{k-1} d_{k-1})^T s_{k-1} \\
&= (\gamma_{k-1} g_{k-1} + \frac{t_k}{r_+} A_o G_{k-1} d_{k-1})^T s_{k-1} \\
&= \gamma_{k-1}(g_{k-1}^T s_{k-1}) + \frac{t_k}{r_+}(G_{k-1} d_{k-1})^T A_o s_{k-1} \\
&= 0\ .
\end{aligned}$$

The last equality is true by the assumptions (A1) and (A5). The relation (R7) follows directly by combining (R3) and (R6). Similarly (R8) follows directly by combining (R2) and (R7). The proof for (R9) is exactly similar to the proof for (R6). The last relation (R10) follows by combining (4.10), (2.19) and (4.35). In particular,

$$\begin{aligned}
(G_{k-1} d_{k-1})^T A_o (G_k d_k) & \\
&= r_+ (G_k d_{k-1})^T A_o (G_k d_k) \\
&= r_+ d_{k-1}^T (G_k^T A_o G_k) d_k \\
&= r_+ d_{k-1}^T A_k d_k = 0\ .
\end{aligned}$$

Having verified the relations (R1) to (R10), we can now use these to complete our proof.

To prove (S1) we observe, according to (R7) and (R8), that it is enough to show it for $j < k - 1$. In this case, by combining (4.17a) and (R9) we obtain

$$g_k^T s_j = \frac{1}{\gamma_k} g_k^T G_k^{-1} s_j = 0\ .$$

Relation (S2) follows by using (4.16b), for (S1)

$$g_k^T d_j = \frac{1}{t_{j+1}} g_k^T (s_{j+1} - s_j) = \frac{1}{t_{j+1}} (g_k^T s_{j+1} - g_k^T s_j) = 0 .$$

To prove (S3), first we recall the formula for the direction d_j,

$$d_j = -\gamma_j G_j^{-1} H_o g_j + \beta_j d_{j-1}$$

where $\beta_j = b_j^T c_j / b_j^T d_{j-1}$. Then by multiplying both sides by g_k we obtain

$$d_j^T g_k = -\gamma_j g_j^T H_o G_j^{-T} g_k + \beta_j d_{j-1}^T g_k$$

and by using (4.17a) we get

$$d_j^T g_k = -\gamma_j^2 g_j^T H_o g_k + \beta_j d_{j-1}^T g_k .$$

Then by solving in terms of $g_j^T H_o g_k$ we have

$$g_j^T H_o g_k = \frac{\beta_j}{\gamma_j^2} d_{j-1}^T g_k - \frac{1}{\gamma_j^2} d_j^T g_k = 0 ,$$

since $d_{k-1}^T g_k = d_j^T g_k = 0$ according to (S2). Now to verify (S4), according to (R10), it is again enough to prove it for $j < k-1$. Indeed by first using the definition of d_k, (4.10) and the assumption (A4) we obtain

$$\begin{aligned}(G_j d_j)^T A_o (G_k d_k) &= d_j^T G_j^T A_o G_k(-\gamma_k G_k^{-1} H_o g_k + \beta_k d_{k-1}) \\ &= d_j^T G_j^T A_o (-\gamma_k H_o g_k + \beta_k G_k d_{k-1}) \\ &= -\gamma_k (d_j^T G_j^T A_o H_o g_k) + \frac{\beta_k}{r_+^2} (G_j d_j)^T A_o (G_{k-1} d_{k-1}) \\ &= -\gamma_k (d_j^T G_j^T A_o H_o g_k) .\end{aligned}$$

But by using (4.10), (4.14), (4.17a) and (S3) the right hand side of the last relation is zero, i.e.

$$d_j^T G_j^T A_o H_o g_k = r_+^2 \, d_j^T \, G_{j+1}^T \, A_o H_o \, g_k$$

$$= r_+^2 \, (A_o G_{j+1} d_j)^T \, H_o \, g_k$$

$$= \frac{r_+}{t_j} \, (G_{j+1}^{-T} \, g_{j+1} - G_j^{-T} g_j)^T \, H_o g_k$$

$$= \frac{r_+}{t_j} \, (\gamma_{j+1} g_{j+1} - \gamma_j g_j)^T H_o g_k$$

$$= \frac{r_+}{t_j} \, [\, \gamma_{j+1} (g_{j+1}^T \, H_o g_k) - \gamma_j (g_j^T H_o g_k)] = 0 \; .$$

Therefore (S4) is true. The last relation (S5) follows by combining (4.15) for $k = j$ and (S4). In fact

$$s_j^T A_o (G_k d_k) = \sum_{i=o}^{j-1} \theta_i (G_i d_i)^T A_o (G_k d_k) = \sum_{i=o}^{j-1} \theta_i 0 = 0 \; .$$

This concludes the proofs of the relations (S1) to (SS) and thus the proof of the theorem.

From an inductive application of this theorem, we can conclude that the relations (A1) to (A5) hold for any $k = 1,2,\ldots$ if ELS are used, at least until $x_k = x^*$. Although the conjugacy relations are different [according to (3.1) and (A4)], the same orthogonality relations are obtained between the directions d_j and the subsequent gradients g_k.

We now wish to derive a revised formula for the directions d_k by using the fact that ELS are used and that the subsequent relation (A1) to (A5) hold. This formula will be derived by computing the quantities $b_k^T c_k$ and $b_k^T d \equiv b_k^T d_{k-1}$. In order to do that we have to state and prove to the following relations:

B1: $$\bar{u} = \frac{1}{\bar{r}_{k-1}}(u_{k-1} - a_k) \;;$$

B2: $$b_k = \frac{\tau}{\bar{r}_{k-1}} u_{k-1} - \frac{\sigma_{k-1}}{\bar{r}_{k-1}} a_k \;;$$

B3: $$u_{k-1}^T H_o g_k = \frac{r_+ g_k^T H_o g_k}{t_k \tau} \;;$$

B4: $$u_{k-1}^T g_k = -\frac{\sigma_{k-1}}{\tau} \;;$$

B5: $$u_{k-1}^T c_k = \frac{\gamma_k^2}{\tau}\left[\frac{r_+ g_k^T H_o g_k}{t_k} + \gamma_k \sigma_{k-1} a_k^T H_o g_k\right];$$

B6: $$a_k^T c_k = \gamma_k^3 a_k^T H_o g_k \;;$$

B7: $$u_{k-1}^T d = -\frac{\sigma_{k-1}}{t_k \tau} \;;$$

B8: $$\frac{\gamma_{k-1}}{\gamma_k} = 1 + t_k a_k^T d$$

where τ is defined to be $\tau \equiv \sigma_{k-1} - \bar{r}_{k-1}^2 \bar{\sigma}$. (4.36)

Again $d \equiv d_{k-1}$. The proofs of the above relations require primarily direct manipulation, the basic relations (4.22) to (4.28), (4.31) and the results of Theorem 2. The first relation (B1) follows directly by solving (4.28) in terms of $\bar{u}$. Relations (B2) follows by substituting (B1) into (4.31). To prove (B3) we use (4.26), (4.3) and (A3). In particular

$$u_{k-1}^T H_o g_k = \frac{g_{k-1}^T H_o g_k - r_+ g_k^T H_o g_k}{-t_k(\sigma_{k-1} - \bar{r}_{k-1}^2 \bar{\sigma})}$$

$$= \frac{-r_+ g_o^T H_o g_k}{-t_k \tau}$$

$$= \frac{r_+ g_k^T H_o g_k}{t_k \tau} \quad .$$

The derivation of (B4) is similar. The relation (B5) follows directly by using (4.26) and (4.32c). For the next relation (B6) we again use (4.32c) and the basic relation (2.7). Indeed

$$\begin{aligned}
a_k^T c_k &= \gamma_k^2 a_k^T (H_o g_k - \gamma_k s_k a_o^T H_o g_k) \\
&= \gamma_k^2 [(a_k^T H_o g_k) - \gamma_k a_k^T s_k (a_o^T H_o g_k)] \\
&= \gamma_k^2 a_k^T H_o g_k (1 - \gamma_k a_k^T s_k) \\
&= \gamma_k^2 a_k^T H_o g_k (1 - a_o^T s_k) \\
&= \gamma_k^3 a_k^T H_o g_k \, .
\end{aligned}$$

The relation (B7) follows directly by combining (4.26) and (A2). The last relation can be proved by using the basic relations (2.7), (4.16a) and (4.16b). We can now obtain expressions for $b_k^T c_k$ and $b_k^T d$. Indeed

$$\begin{aligned}
b_k^T c_k &= \left[\frac{\tau}{\bar{r}_{k-1}} u_{k-1} - \frac{\sigma_{k-1}}{\bar{r}_{k-1}} a_k \right]^T c_k \\
&= \frac{\tau}{\bar{r}_{k-1}} u_{k-1}^T c_k - \frac{\sigma_{k-1}}{\bar{r}_{k-1}} a_k^T c_k \\
&= \frac{\tau}{\bar{r}_{k-1}} \left[\frac{\gamma_k^2}{\tau} \left(\frac{r_+ g_k^T H_o g_k}{t_k} + \gamma_k \sigma_{k-1} a_k^T H_o g_k \right) \right] - \frac{\sigma_{k-1}}{\bar{r}_{k-1}} \gamma_k^3 a_k^T H_o g_k \\
&= \frac{\gamma_k^2 r_+}{\bar{r}_{k-1} t_k} g_k^T H_o g_k + \frac{\gamma_k^3 \sigma_{k-1}}{\bar{r}_{k-1}} a_k^T H_o g_k - \frac{\gamma_k^3 \sigma_{k-1}}{\bar{r}_{k-1}} a_k^T H_o g_k \\
&= \frac{\gamma_k^2 r_+}{\bar{r}_{k-1} t_k} g_k^T H_o g_k \quad , \qquad (4.37)
\end{aligned}$$

and

$$b_k^T d = \left[\frac{\tau}{\bar{r}_{k-1}} u_{k-1} - \frac{\sigma_{k-1}}{\bar{r}_{k-1}} a_k\right]^T d$$

$$= \frac{\tau}{\bar{r}_{k-1}} u_{k-1}^T d - \frac{\sigma_{k-1}}{\bar{r}_{k-1}} a_k^T d$$

$$= -\frac{\tau}{\bar{r}_{k-1}} \cdot \frac{\sigma_{k-1}}{t_k \tau} - \frac{\sigma_{k-1}}{\bar{r}_{k-1}} a_k^T d$$

$$= -\frac{\sigma_{k-1}}{t_k \bar{r}_{k-1}} (1 + t_k a_k^T d)$$

$$= -\frac{\sigma_{k-1}\gamma_{k-1}}{t_k \bar{r}_{k-1}\gamma_k} . \tag{4.38}$$

Finally then by using (4.37), (4.38) and (4.32c), the formula for d_k becomes

$$d_k = -c_k + \frac{b_k^T c_k}{b_k^T d} d$$

$$= -c_k - \frac{\gamma_k^2 r_+}{\bar{r}_{k-1} t_k} \cdot \frac{t_k \bar{r}_{k-1}\gamma_k}{\sigma_{k-1}\gamma_{k-1}} \cdot g_k^T H_o g_k \cdot d$$

$$= -c_k - \frac{\gamma_k^2 \gamma_k}{\gamma_{k-1}} \cdot \frac{\gamma_k}{\gamma_{k-1}} \cdot g_k^T H_o g_k \cdot d$$

$$= -\gamma_k G_k^{-1} H_o g_k - \frac{\gamma_k^4}{\gamma_{k-1}^2} \cdot \frac{g_k^T H_o g_k}{\sigma_{k-1}} \cdot d \tag{4.39}$$

The reason for which we have derived the last formula (4.39) will become clear in the next section (4.3) where we will establish that the O(n) and GN algorithms are identical if they are applied to normal conic functions and ELS are used. Also (4.39) is much closer in form to the regular CG equation (see (3.1c)).

4.3 The Gourgeon-Nocedal Algorithm

In this section we will introduce another algorithm that will minimize a normal conic function in a finite number of steps. It can also be considered as a generalization of the PCG algorithm [see section (3.1)], and has similar orthogonality properties. The derivation of the algorithm by Gourgeon and Nocedal in [11] and ours are quite similar, but we generalize the algorithm somewhat and clarify the description. The GN algorithm occurs as a special case.

The method is based on the following idea. By initially transforming the variables x of the conic function f under a collinear scaling T as defined in (2.20), we get the quadratic function $\bar{q}(w)$ (2.23). Next, instead of minimizing the normal conic function f, we apply the PCG algorithm with a preconditioning matrix H_o [see section (3.1)] to the quadratic $\bar{q}(w)$. Then by transforming the iterates back, we obtain a minimization algorithm for the conic f.

First of all, in order to explain and derive the above strategy, we need to introduce some notation. For the quadratic problem with $\bar{q}$, we will denote the search directions by $v_o, v_1, \ldots, v_k$, the displacements are given by $\mu_1 v_o, \mu_2 v_1, \ldots, \mu_{k+1} v_k$ and the points obtained are denoted by $w_o, w_1, \ldots, w_k$. For the conic f we will use the earlier notation. Note that by w_i we may also denote the total displacement, since $w_o = 0$, i.e.

$$w_i = \sum_{j=1}^{i} \mu_j v_{j-1} \quad . \tag{4.40}$$

The relation between s_k and w_k is given, according to (2.20), by:

$$w_i = \frac{Js_i}{\gamma_i} , \qquad i = 0,1,2,\ldots \quad . \tag{4.41}$$

Our purpose is to derive a formula for the search direction d_k for the conic problem. In order to do this, we need the following relations. With $h_k \equiv \nabla\bar{q}(w_1)$, we have

$$h_k = J_k^{-T} g_k , \tag{4.42}$$

$$J^{-T}A_oJ^{-1}(\mu_k v_{k-1}) = h_k - h_{k-1} , \tag{4.43}$$

$$d_k = \frac{\gamma_{k+1}\mu_{k+1}}{\gamma_k t_{k+1}} J_k^{-1} v_k , \tag{4.44}$$

$$\gamma_{k+1} = \frac{\gamma_k}{1 + \gamma_k a_o^T J^{-1}(\mu_{k+1} v_k)} , \tag{4.45}$$

$$t_{k+1} d_k = \left[\frac{\mu_{k+1}(1+\gamma_k a_o^T J^{-1} v_k)}{(1+\gamma_k \mu_k a_o^T J^{-1} v_k)} \right] \frac{J_k^{-1} v_k}{1+\gamma_k a_o^T J^{-1} v_k} . \tag{4.46}$$

The first equation (4.42), relates the gradients of the quadratic and conic, and the proof follows directly by applying the chain rule to f. Relation (4.43) is verified by using (3.12) for the quadratic $\bar{q}(w)$. This statement is just the secant equation for the quadratic $\bar{q}$ since the Hessian of f is $J^{-T} A J^{-1}$. Equation (4.44), which relates d_k and v_k follows by using (4.41), (4.8) and the basic relation that $J_k = JG_k$. In particular

$$\begin{aligned} \mu_{k+1} v_k &= w_{k+1} - w_k \\ &= J\left[\frac{s_{k+1}}{\gamma_{k+1}} - \frac{s_k}{\gamma_k} \right] \\ &= \frac{t_{k+1}}{r_+} JG_k d_k \\ &= \frac{t_{k+1}\gamma_k}{\gamma_{k+1}} J_k d_k , \end{aligned}$$

and by solving the last equation for d_k we obtain (4.44). It is the key in obtaining the desired direction d_k. To prove (4.45) we use (4.41) and the basic equation (4.16a). Here

$$
\begin{aligned}
1 + \gamma_k a_o^T J^{-1}(\mu_{k+1} v_k) \\
&= 1 + \gamma_k a_o^T J^{-1}(w_{k+1} - w_k) \\
&= 1 + \gamma_k a_o^T J^{-1}\left[\frac{J s_{k+1}}{\gamma_{k+1}} - \frac{J s_k}{\gamma_k}\right] \\
&= 1 + \gamma_k \frac{a_o^T s_{k+1}}{\gamma_{k+1}} - \gamma_k \frac{a_o^T s_k}{\gamma_k} \\
&= 1 + \gamma_k \left[\frac{1}{\gamma_{k+1}} - 1\right] - 1 + \gamma_k \\
&= \frac{\gamma_k}{\gamma_{k+1}} ,
\end{aligned}
$$

from which we may solve the last equation for γ_{k+1}. The last relation (4.46) follows by substituting the equation (4.45) into (4.44) and multiplying and dividing by the same quantity $(1 + \gamma_k a_o^T J^{-1} v_k)$.

Following Gourgeon and Nocedal, we split the relation (4.46) as

$$d_k = \frac{J_k^{-1} v_k}{1 + \gamma_k a_o^T J^{-1} v_k} , \tag{4.47}$$

$$t_{k+1} = \frac{\mu_{k+1}(1 + \gamma_k a_o^T J^{-1} v_k)}{(1 + \gamma_k \mu_{k+1} a_o^T J^{-1} v_k)} , \tag{4.48}$$

so that $\mu_{k+1} = 1$ corresponds to $t_{k+1} = 1$. Therefore from the direction v_k we can get the direction d_k.

We will now derive the algorithm. First we apply the conjugate gradient algorithm to the quadratic function $\bar{q}(w)$ with preconditioner H_o and we assume that ELS are used. Therefore, according to our discussion in section (3.1), the following properties hold:

$$h_j^T w_i = 0 \quad , \qquad 0 \le i \le j \; ; \tag{4.49}$$

$$h_j^T v_i = 0 \quad , \qquad 0 \le i < j \; ; \tag{4.50}$$

$$h_j^T H_o h_i = 0 \quad , \qquad 0 \le i < j \; . \tag{4.51}$$

By using the relation between the search directions d_k and v_k, and the transformation T, we can find that, with the given initial point x_o [x_o is taken as the reference point], the first search direction for the conic f will be $d_o = -J^{-1} H_o J^{-T} g_o$.

We will now give a description of the general k^{th} step where we assume we have just reached the point x_{k-1}. First we choose a trial step length $\bar{t}$ and we evaluate the function and the gradient at $\bar{x} = x_{k-1} + \bar{t} d_{k-1}$. By using the function and gradient values at the points x_{k-1} and $\bar{x}$, we can determine the minimizer x_k along the direction d_{k-1} [see section (2.2)]. We also evaluate the function and the gradient at x_k. With this information we will be able to determine the next search direction d_k. Note that, at the first iteration only, with the above information we can also determine the horizon vector a_o by using Lemma 1.

At the k^{th} step, to find the formula for the search direction d_k we have to transform everything from the quadratic-space to the conic-space by using T. But in quadratic-space the conjugate gradient search direction v_k is given by:

$$v_k = -H_o h_k + \beta_k v_{k-1} \tag{4.52}$$

where

$$\beta_k = \frac{h_k^T H_o (h_k - h_{k-1})}{v_{k-1}^T (h_k - h_{k-1})} \tag{4.53}$$

[see section (3.1)]. By substituting (4.42), (4.43) and (4.53) into (4.52) we get

$$v_k = -H_o J_k^{-T} g_k + \frac{g_k^T J_k^{-1} H_o (J^{-T} A_o J^{-1}) v_{k-1}}{v_{k-1}^T (J^{-T} A_o J^{-1}) v_{k-1}} . \tag{4.54}$$

In order to do as much simplification as possible in getting the search direction d_k, we will prove the following orthogonality relations:

R1 $\quad g_j^T s_i = 0 , \qquad 1 \le i \le j ;$

R2 $\quad g_j^T d_i = 0 , \qquad 0 \le i < j ;$

R3 $\quad g_i^T J^{-1} H_o J^{-T} g_i = 0 , \qquad 0 \le i < j .$

These follow from (4.49) - (4.51). The first equation R1 follows by using (4.41), (4.42) and particularly (4.49). Indeed

$$\begin{aligned} g_j^T s_i &= \gamma_i h_j^T J_j J^{-1} w_i \\ &= \gamma_i h_j^T J \left[I - \frac{s_j a_o^T}{\gamma_j} \right] J^{-1} w_i \\ &= \gamma_i h_j^T [w_i - \frac{J s_j}{\gamma_j} (a_o^T J^{-1} w_i)] \\ &= \gamma_i h_j^T [w_i - w_j (a_o^T J^{-1} w_i)] \\ &= \gamma_i (h_j^T w_i) - (h_j^T w_j)(a_o^T J^{-1} w_i) \\ &= 0 . \end{aligned}$$

Relation (R2) follows directly from (R1). For the last equation (R3), first, by using (R1) and (4.17a), we have

$$J_k^{-T} g_k = \gamma_k J^{-T} g_k \tag{4.55}$$

and by substituting the last equation into (4.42) we obtain

$$h_k = \gamma_k J^{-T} g_k . \tag{4.56}$$

Then the proof for (R3) follows by using (4.51) and (4.56) twice for $k = i$ and $k = j$. We let $Y_k \equiv \gamma_k g_k - \gamma_{k-1} g_{k-1}$. As a consequence of the above orthogonality properties (R1), (R2) and (R3),

$$J^{-T} A_o J^{-1} (\mu_k v_{k-1}) = J^{-T} Y_k \quad , \tag{4.57}$$

$$J_{k-1}^T J^{-T} y_k = \frac{1}{\gamma_{k-1}} y_k \quad , \tag{4.58}$$

$$d_{k-1}^T y_k = \gamma_{k-1} d_{k-1}^T g_{k-1} \quad , \tag{4.59}$$

$$g_k^T J^{-1} H_o J^{-T} y_k = \gamma_k g_k^T J^{-1} H_o J^{-T} g_k \quad , \tag{4.60}$$

which will be used to simplify the formula (4.54) for the direction v_k. The first relation follows directly by combining (4.43) and (4.56). The proof for (4.58) follows by using (4.17b) for $j = k-1$ and $i = k$ and (4.55) for $k-1$. In particular

$$\begin{aligned} J_{k-1}^T J^{-T} y_k &= J_{k-1}^T J^{-T} (\gamma_k g_k - \gamma_{k-1} g_{k-1}) \\ &= \gamma_k J_{k-1}^T J^{-T} g_k - \gamma_{k-1} J_{k-1}^T J^{-T} g_{k-1} \\ &= \gamma_k G_{k-1}^T g_k - \frac{\gamma_{k-1}}{\gamma_{k-1}} J_{k-1}^T J_{k-1}^{-T} g_{k-1} \\ &= \frac{\gamma_k}{\gamma_{k-1}} g_k - \frac{\gamma_{k-1}}{\gamma_{k-1}} g_{k-1} \quad . \\ &= \frac{1}{\gamma_{k-1}} y_k \quad . \end{aligned}$$

Relations (4.59) and (4.60) follow directly by using (R2) and (R3).

We can now reduce the formula (4.54). First by using (4.55), (4.57), (4,58) and (4.44) for $k-1$ we get

$$v_k = -\gamma_k H_o J^{-T} g_k + \frac{\gamma_k g_k^T J^{-1} H_o J^{-T} y_k}{v_{k-1}^T J^{-T} y_k} v_{k-1} \quad .$$

$$= -\gamma_k H_o J^{-T} g_k + \frac{\gamma_k g_k^T J^{-1} H_o J^{-T} y_k}{d_{k-1}^T J_{k-1}^T J^{-T} y_k} J_{k-1} d_{k-1}$$

$$= -\gamma_k H_o J^{-T} g_k + \frac{\gamma_k g_k^T J^{-1} H_o J^{-T} y_k}{d_{k-1}^T y_k / \gamma_{k-1}} J_{k-1} d_{k-1} \equiv z_k \qquad (4.61)$$

Next we multiply both sides of (4.61) by J_k^{-1} and by using the obvious alternate formula of (4.10), $r_+^2 J_k d_{k-1} = J_{k-1} d_{k-1}$, we obtain

$$J_k^{-1} v_k = -\gamma_k J_k^{-1} H_o J^{-T} g_k + \frac{\gamma_k g_k^T J^{-1} H_o J^{-T} y_k}{d_{k-1}^T y_k / \gamma_{k-1}} J_k^{-1} J_{k-1} d_{k-1}$$

$$= -\gamma_k G_k^{-1} J^{-1} H_o J^{-T} g_k + \frac{\gamma_k g_k^T J^{-1} H_o J^{-T} y_k}{d_{k-1}^T y_k / \gamma_{k-1}} r_+^2 J_k^{-1} J_k d_{k-1}$$

$$= -\gamma_k G_k^{-1} (J^{-1} H_o J^{-T}) g_k + \frac{\gamma_k^3 g_k^T J^{-1} H_o J^{-T} y_k}{\gamma_{k-1} d_{k-1}^T y_k} d_{k-1} .$$

Then, by using (4.47) we have the formula for d_k. In fact,

$$d_k = n_k [-\gamma_k G_k^{-1} (J^{-1} H_o J^{-T}) g_k + \frac{\gamma_k^3 g_k^T J^{-1} H_o J^{-T} y_k}{\gamma_{k-1} d_{k-1}^T y_k} d_{k-1}]$$

where the constant n_k, according to (4.46), is

$$n_k \equiv \frac{1}{1 + \gamma_k g_o^T J^{-1} z_k} . \qquad (4.62)$$

Some more simplifications, by using the orthogonality properties (R2) and (R3) give the final form for d_k

$$d_k = n_k [-\gamma_k G_k^{-1} (J^{-1} H_o J^{-T}) g_k + \frac{\gamma_k^4 g_k^T J^{-1} H_o J^{-1} g_k}{\gamma_{k-1}^2 \, d_{k-1}^T g_{k-1}} d_{k-1}]$$

$$= n_k[-\gamma_k G_k^{-1}(J^{-1} H_o J^{-T}) g_k + \frac{\gamma_k^4 g_k^T J^{-1} H_o J^{-T} g_k}{\gamma_{k-1}^2 \sigma_{k-1}} d] \quad (4.63)$$

where we recall $\sigma_{k-1} = d_{k-1}^T g_{k-1}$ and $d = d_{k-1}$.

So finally, by deriving the formula (4.63) for the direction d_k, we have completed the description of the algorithm. We first note that the horizon vector is computed only once, at the beginning of the first iteration in contrast to Davidon's $0(n)$ algorithm where it is computed at each iteration. As we mentioned at the beginning of this section, the GN algorithm is a special case of this algorithm. In particular we can get it by just choosing $J = I$. Then the search direction d_k is reduced to the following form

$$\bar{d}_k = \bar{n}_k[-\gamma_k G_k^{-1} H_o g_k + \frac{\gamma_k^4 g_k^T H_o g_k}{\gamma_{k-1}^2 \sigma_{k-1}} \bar{d}] , \quad (4.64)$$

where $$\bar{n}_k \equiv \frac{1}{1 + \gamma_k a_o^T z_k} . \quad (4.65)$$

Having derived the directions for the $0(n)$ and GN algorithms we can now observe that they generate the same sequence of points x_k when they are applied to a normal conic function and ELS are used. Since it is assumed that both use ELS it is enough to show that the directions are the same, i.e. that $d_i = \bar{d}_i$ or $d_i = \xi_i \bar{d}_i$ for $i = 0,1,2,\ldots$, and for some scalar ξ_i . By $\bar{d}_i$ we mean the search directions of the GN algorithm. The proof is by induction. First, for $i = o$ it is true since $d_o = \bar{d}_o = -H_o g_o$. By assuming now that $d_{k-1} = \bar{d}_{k-1}$ we can see by comparing (4.39) and (4.64) that $d_k = \frac{1}{\bar{n}_k} \bar{d}_k$ which completes the induction and our proof.

To conclude this section, we will make some important observations which give a Theorem which is a generalization of the Expanding Subspace Theorem (EST) [see (3.1)]. First we observe, by using (4.56), that $h_k = 0$ implies $g_k = 0$ also. Therefore according to the previous discussion, the GN algorithm and hence the $\dot{O}(n)$ algorithm are terminated in at most n steps when they are applied to normal conic function and ELS are used. We will require a definition in order to state our theorem.

<u>Definition</u>: Two lines in the domain of the normal conic function f are conjugate if and only if they are images of orthogonal ones in the domain of the quadratic $\bar{q}$ under a collinear scaling T. The previous discussion now establishes the following Theorem.

<u>Theorem 4.3</u>: Let d_i, $i = 1,2,\ldots,n$ be a set of conjugate lines in $\mathbb{R}^n$ and let B_k be the subspace of $\mathbb{R}^n$ spanned by $d_1, d_2, \ldots, d_k$. Then for $x_0 \in \mathbb{R}^n$ the sequence $\{x_k\}$ generated according to

$$x_k = x_{k-1} + t_k d_{k-1}$$

has the property that x_k minimizes the normal conic function f on the linear variety $x_0 + B_k$, provided all line searches are exact.

4.4 Davidon's $O(n^2)$ Algorithm

Davidon [6] also proposes another algorithm based on conics. This one can be considered as a generalization of the QN method and particularly of the BFGS algorithm [section (3.2)]. It uses $O(n^2)$ operations per iteration to update approximations B_k to the conjugacy matrix A_k; for that reason Davidon called it "the $O(n^2)$ algorithm".

The derivation of the $O(n^2)$ algorithm is based on the same idea as the $O(n)$ algorithm (4.2). The only distinction between them is at step 9, where in $O(n^2)$ the search direction d_k is computed in order to satisfy the equation $B_k d_k = -g_k$ at each iteration. In fact in the $O(n^2)$ algorithm, we have the new

Step 9. Set

$$\bar{B}_k \equiv L_k^T B_{k-1} L_k \ , \tag{4.66}$$

$$B_k = \bar{B}_k - \frac{\bar{B}_k d d^T \bar{B}_k}{d^T \bar{B}_k d} + \frac{b_k b_k^T}{2p} \bar{t}^2 \tag{4.67}$$

and $$d_k = - B_k^{-1} g_k \ . \tag{4.68}$$

Note that by d we again mean d_{k-1}. For each i, we assume $H_i \equiv B_i^{-1}$.

In the same manner as in Section 3.2 for the BFGS algorithm, we can prove that the formula (4.67) preserves positive definiteness in the matrices B_k, provided the quantity p is positive and the matrix B_o is positive definite. As we mentioned previously, the quantity p is always positive and the matrix B_o will be chosen to be positive definite. Therefore the formula (4.67) generates positive definite matrices B_k for any k. We can also observe that the $O(n^2)$

algorithm shares all of the orthogonality properties with the $O(n)$ algorithm which are given through the Theorem 4.2. The proofs for these relations are executed in the same manner, provided that the formula for the direction d_k is taken into consideration, since this is the only difference between the $O(n)$ and $O(n^2)$ algorithms. For example, to prove the relation R10 [Section 4.2], we also use (4.10) and (2.19) to get

$$\begin{aligned}(G_{k-1}d_{k-1})^T A_o (G_k d_k) &= r_+ (G_k d_{k-1})^T A_o (G_k d_k) \\ &= r_+ d_{k-1}^T (G_k^T A_o G_k) d_k \\ &= r_+ d_{k-1}^T A_k d_k \ .\end{aligned}$$

Then, by multiplying both sides of (4.67) by d_k and d_{k-1} and using (4.68) and (4.33a) we obtain

$$\begin{aligned}-d_{k-1}^T g_k &= (b_k^T d_{k-1})(b_k^T d_k)\,\bar{t}^2/2p \\ &= (b_k^T d_{k-1})(d_{k-1}^T A_k d_k)\,\gamma_k^2 \bar{t}^2/2p\,\gamma_{k-1}\bar{\gamma} \ .\end{aligned}$$

Since ELS are used we get $d_{k-1}^T A_k d_k = 0$. Therefore R10 holds for the $O(n^2)$ algorithm as well.

As we mentioned at the beginning of this thesis, we will finally prove in this section that both, $O(n)$ and $O(n^2)$ algorithms determine the same sequence of points, when they are applied to normal conic funtions and ELS are used. But first, we must examine some relations satisfied by the matrices B . Note that by M_k we denote the second update term, i.e.

$$M_k \equiv \frac{\bar{B}_k d_{k-1} d_{k-1}^T \bar{B}_k}{d_{k-1}^T \bar{B}_k d_{k-1}} \ . \qquad (4.69)$$

Then the following hold:

$$\bar{B}_k d_{k-1} = -\frac{1}{r_+^2} L_k^T g_{k-1} , \tag{4.70}$$

$$M_k = -\frac{L_k^T g_{k-1} g_{k-1}^T L_k}{d_{k-1}^T g_{k-1}} , \tag{4.71}$$

$$B_k d_{k-1} = \frac{b_k b_k^T d_{k-1}}{2p} \bar{t}^2 , \tag{4.72}$$

$$M_k c_k = 0 \; . \tag{4.73}$$

The first equation (4.70) follows by using (4.66), (4.10), (4.5) and the basic relation $B_i d_i = -g_i$ for $i = k-1$. In particular

$$\bar{B}_k d_{k-1} = L_k^T B_{k-1} L_k d_{k-1} = L_k^T B_{k-1} G_{k-1}^{-1} G_k d_{k-1}$$

$$= \frac{1}{r_+^2} L_k^T B_{k-1} G_{k-1}^{-1} G_{k-1} d_{k-1} = \frac{1}{r_+^2} L_k^T B_{k-1} d_{k-1} = -\frac{1}{r_+^2} L_k g_{k-1} \; .$$

To prove relation (4.71) we use basically (4.70), (4.5) and (4.10):

$$M_k = \frac{\frac{1}{r_+^4} L_k^T g_{k-1} g_{k-1}^T L_k}{-\frac{1}{r_+^2} d_{k-1}^T L_k^T g_{k-1}} = -\frac{L_k^T g_{k-1} g_{k-1}^T L_k}{r_+^2 d_{k-1}^T G_k^T G_{k-1}^{-T} g_{k-1}}$$

$$= -\frac{L_k^T g_{k-1} g_{k-1}^T L_k}{g_{k-1}^T G_{k-1}^T G_{k-1}^{-T} g_{k-1}} = -\frac{L_k^T g_{k-1} g_{k-1}^T L_k}{d_{k-1}^T g_{k-1}}$$

Relation (4.72) follows directly by multiplying both sides of (4.67) by d_{k-1} . By using the orthogonality results of the Theorem 4.2 with (4.71), (4.32c), (4.5), (4.17a) we obtain (4.73). In particular

$$M_k c_k = -\frac{\gamma_k L_k^T g_{k-1} g_{k-1}^T L_k G_k^{-1} H_o g_k}{d_{k-1}^T g_{k-1}}$$

$$= - \frac{\gamma_k L_k^T g_{k-1} g_{k-1}^T G_{k-1}^{-1} H_o g_k}{d_{k-1}^T g_{k-1}}$$

$$= - \frac{\gamma_k \gamma_{k-1} L_k^T g_{k-1} g_{k-1}^T H_o g_k}{d_{k-1}^T g_{k-1}} = 0 \; .$$

Next, we wish to present our main theorem. But before that some simple observations will be given to initiate the induction used in applying the Theorem. Temporarily, we denote by $\hat{d}_i$ the $O(n^2)$ search directions and by d_i those for the $O(n)$ algorithm. Then

$\overline{01}$: $\hat{d}_o = d_o$;

$\overline{02}$: $B_o G_o^{-1} H_o v = \tau v$ for all $H_o v \perp g_o$,

for some scalar τ . Relation $\overline{01}$ is true since for both algorithms we have made the same initial choice $-H_o g_o$. The second equation $\overline{02}$ follows by using the fact that $B_o \equiv H_o^{-1}$ and $G_o^{-1} = I$. Now we can state our main Theorem.

<u>Theorem 4.4</u>: Suppose $k > 1$ and $x_i \neq x^*$ for $i < k$, where x^* is the global minimum of f . Assume

$\overline{A1}$: $\hat{d}_i = \bar{\tau}_i d_i$ for all $i = 1, \ldots , k-1$;

$\overline{A2}$: $B_i G_i^{-1} H_o v = \tau_i v$ for all $H_o v \perp g_i$, $i = 0, 1, \ldots , k-1$,

where τ_i and $\bar{\tau}_i$ are appropriate constants, and that we have just reached x_k by an ELS along $\hat{d}_{k-1}$ $(= d_{k-1})$. Then either $x_k = x^*$ or $\overline{A1}$ and $\overline{A2}$ hold with $k-1$ replaced by k .

<u>Proof</u>: First of all, we can see that it is sufficient to show that

$\overline{S1}$: $\hat{d}_k = \bar{\tau}_k d_k$

$\overline{S2}$: $B_k G_k^{-1} H_o v = \tau_k v$ for all $H_o v \perp g_i$, $i = 0, 1, \ldots , k$.

To prove $(\bar{S1})$ it is obviously equivalent to show that $B_k d_k = 1/\bar{\tau}_k g_k$, since $B_k \hat{d}_k = g_k$. In order to prove that, first we use (4.32b), (4.67), (4.72) and (4.73). In fact

$$
\begin{aligned}
B_k d_k &= - B_k c_k + B_k d \frac{b_k^T c_k}{b_k^T d} \\
&= - \bar{B}_k c_k + M_k c_k - \frac{b_k b_k^T c_k}{2p} \bar{t}^2 + \frac{b_k b_k^T d}{2p} \cdot \frac{b_k^T c_k}{b_k^T d} \bar{t}^2 \\
&= - \bar{B}_k c_k .
\end{aligned}
$$

Then by using (4.66), (4,32c) and (4.5) we obtain

$$
\begin{aligned}
B_k d_k &= - \gamma_k L_k^T B_{k-1} L_k G_k^{-1} H_o g_k \\
&= - \gamma_k L_k^T B_{k-1} G_{k-1}^{-1} H_o g_k ,
\end{aligned}
$$

and finally by using the assumptions of the Theorem and (4.18) we get

$$
\begin{aligned}
B_k d_k &= \tau_k L_k^T g_k \\
&= \bar{\tau}_k g_k .
\end{aligned}
$$

Next in order to prove $(\bar{\bar{S2}})$ we will state and prove some equations which hold only under the assumptions of the Theorem:

$\bar{R1}$: $$M_k G_k^{-1} H_o v = 0 \quad ;$$

$\bar{R2}$: $$b_k^T G_k^{-1} H_o v = 0 \quad ;$$

$\bar{R3}$: $$L_k^T B_{k-1} G_{k-1}^{-1} H_o v = \tau v \quad .$$

The proof for $\bar{R1}$ follows by using (4.71), (4.5), (4.17a) and finally the assumptions of the Theorem, as

$$M_k G_k^{-1} H_o v = -\frac{L_k^T g_{k-1} g_{k-1}^T L_k G_k^{-1} H_o v}{d_{k-1}^T g_{k-1}}$$

$$= -\frac{L_k^T g_{k-1} g_{k-1}^T G_{k-1}^{-1} H_o v}{d_{k-1}^T g_{k-1}}$$

$$= -\frac{\gamma_{k-1} L_k^T g_{k-1} (g_{k-1}^T H_o v)}{d_{k-1}^T g_{k-1}}$$

$$= 0 .$$

To prove $\overline{R2}$, use (4.33a) and (2.19) to obtain

$$b_k^T G_k^{-1} H_o v = \tau d_{k-1}^T A_k G_k^{-1} H_o v$$

$$= \tau d_{k-1}^T G_k^T A_k G_k G_k^{-1} H_o v$$

$$= \tau d_{k-1}^T G_k^T A_k H_o v ,$$

where $\tau = \gamma_k^2/\gamma_{k-1}\bar{r}_{k-1}$. Then by using (4.14), (4.17a) and the assumption of the Theorem we get

$$b_k^T G_k^{-1} H_o v = \tau (A_o G_k d_{k-1})^T H_o v$$

$$= \tau' (G_k^{-T} g_k - G_{k-1}^{-T} g_{k-1})^T H_o v$$

$$= \tau' (\gamma_k g_k - \gamma_{k-1} g_{k-1})^T H_o v$$

$$= \tau' \gamma_k (g_k^T H_o v) - \tau' \gamma_{k-1} (g_{k-1}^T H_o v)$$

$$= 0 ,$$

where $\tau' \equiv \tau/t_k r_+$. The last relation ($\overline{R3}$) follows by using the orthogonality results of Theorem 4.2, the assumptions of the Theorem and the basic relation that $B_i d_i = -g_i$ for $k = k-1$. In fact

$$L_k^T B_{k-1} G_{k-1}^{-1} H_o v = \frac{\gamma_k}{r_+}[I + t_k a_k d_{k-1}^T] B_{k-1}[I - s_{k-1} a_o^T] H_o v$$

$$= \frac{\gamma_k}{r_+}[B_{k-1} + t_k a_k d_{k-1}^T B_{k-1}][H_o v - s_{k-1} a_o^T H_o v]$$

$$= \frac{\gamma_k}{r_+}[B_{k-1} - t_k a_k g_{k-1}^T][H_o v - s_{k-1} a_o^T H_o v]$$

$$= \frac{\gamma_k}{r_+}[B_{k-1} H_o v - t_k a_k (g_{k-1}^T H_o v) - B_{k-1} s_{k-1} a_o^T H_o v$$

$$+ t_k a_k (g_{k-1}^T s_{k-1}) a_o^T H_o v]$$

$$= \frac{\gamma_k}{r_+}[B_{k-1} H_o v - B_{k-1} s_{k-1} a_o^T H_o v]$$

$$= \frac{\gamma_k}{r_+} B_{k-1}[I - s_{k-1} a_o^T] H_o v$$

$$= \frac{1}{r_+} B_{k-1} G_{k-1}^{-1} H_o v$$

$$= \tau v \ .$$

Now by using $(\overline{R1})$, $(\overline{R2})$, $(\overline{R3})$ and (4.66) we can easily obtain $(\overline{\overline{S2}})$. In fact

$$B_k G_k^{-1} H_o v = \bar{B}_k G_k^{-1} H_o v - M_k G_k^{-1} H_o v + \frac{b_k (b_k^T G_k^{-1} H_o v)}{2p} \cdot \bar{t}^2$$

$$= \bar{B}_k G_k^{-1} H_o v$$

$$= L_k^T B_{k-1} L_k G_k^{-1} H_o v$$

$$= \tau_k v \ .$$

This concludes the proof of the Theorem 4.4.

From an inductive application of this Theorem, we can conclude that

the $0(n)$ and $0(n^2)$ algorithms produce the same search directions if ELS are used and they are applied to normal conic function. As a consequence of this they also produce the same sequence of points x_k .

CHAPTER V

MULTIPLE UPDATES: THE VSOn ALGORITHM

5.1 The Relation between the $O(n)$ and $O(n^2)$ Algorithm

The close relationship that exists between the CG algorithm and the BFGS form of the QN algorithm has been established in Section 3.3. It is the purpose of this section to extend this relationship to the algorithms based on conics and to derive similar results for the $O(n)$ and $O(n^2)$ algorithms. Specifically, we will demonstrate how the $O(n)$ algorithm, under an appropriate choice of the updating matrix B, can be written in an $O(n^2)$-like manner.

In order to do that, we will derive an alternate update formula to (4.67), namely one which directly updates B_k^{-1}, since we need it to determine the search direction $d_k = -B_k^{-1} g_k$. So first we will note another relation, which we need to reduce the form (4.67) and to make it similar to (3.14), namely

$$\frac{2p}{\bar{t}^2} = \frac{\gamma_k^4}{\gamma_{k-1}^2 \bar{\gamma}^2} \, d^T A_k d \, , \tag{5.1}$$

where we recall that $d \equiv d_{k-1}$. This relation follows by first using the definition (2.24) of $p \equiv p_{ij}$, for $x_j = \bar{x}$ and $x_i = x_{k-1}$:

$$2p = \left(\frac{\bar{\gamma}}{\gamma_{k-1}} \bar{g} - \frac{\gamma_{k-1}}{\bar{\gamma}} g_{k-1} \right)^T (\bar{x} - x_{k-1})$$

$$= \left[\frac{\bar{\gamma}^2 \bar{g}^T - \gamma_{k-1}^2 g_{k-1}^T}{\gamma_{k-1} \bar{\gamma}} \right] \bar{t} d$$

$$= \frac{\bar{t}}{\gamma_{k-1}\bar{\gamma}} (\bar{\gamma}^2\bar{\sigma} - \gamma_{k-1}^2\sigma_{k-1}) ,$$

and then by combining this with (4.30) we obtain (5.1). Next, by substituting (5.1) into (4.67) and by using (4.33a) twice we get the following formula:

$$B_k = \bar{B}_k - \frac{\bar{B}_k dd^T \bar{B}_k}{d^T \bar{B}_k d} + \frac{A_k dd^T A_k}{d^T A_k d} , \tag{5.2}$$

which is much closer in form to the BFGS update formula (see (3.14)).

Now, by using (3.13b), but with the substitutions of $\bar{B}_k$ for B, A_k for Q and d for s, we get an update formula for B_k^{-1}:

$$B_k^{-1} = \bar{B}_k^{-1} - \frac{\bar{B}_k^{-1} A_k dd^T + dd^T A_k \bar{B}_k^{-1}}{d^T A_k d}$$

$$+ \left(1 + \frac{d^T A_k \bar{B}_k^{-1} A_k d}{d^T A_k d}\right) \frac{dd^T}{d^T A_k d} . \tag{5.3}$$

This can be easily verified by checking that $B_k B_k^{-1} = I$.

For our purpose now, we recall the direction $d_k = -B_k^{-1} g_k$. We wish to relate this to the direction of the $0(n)$ algorithm (4.32c). In order to do that we have to compute $B_k^{-1} g_k$ by using (5.3). Note that, in this computation, there are a number of terms containing the form $d^T g_k$, which disappear since ELS are used. In fact

$$d_k = -B_k^{-1} g_k = -\bar{B}_k^{-1} g_k + \frac{d^T A_k \bar{B}_k^{-1} g_k}{d^T A_k d} d , \tag{5.4}$$

where we recall that $\bar{B}_k = L_k^T B_{k-1} L_k$. By comparing (5.4) and the conjugate direction (4.32b) for the $0(n)$ algorithm, which we recall is

$$d_k = -\gamma_k G_k^{-1} B_o^{-1} g_k + \frac{b_k^T c_k}{b_k^T c_k} d ,$$

we observe that the direction d_k for the $O(n^2)$ algorithm will be identical to the direction d_k for the $O(n)$ algorithm if the following equations hold:

$$L_k^{-1} B_{k-1}^{-1} L_k^{-T} \equiv \gamma_k G_k^{-1} B_o^{-1} , \tag{5.5}$$

$$\frac{d^T A_k \bar{B}_k^{-1} g_k}{d^T A_k d} = \frac{b_k^T c_k}{b_k^T d} . \tag{5.6}$$

The first equation (5.5) gives us what must be the choice of the matrix B_{k-1}, by solving (5.5) in terms of B_{k-1}^{-1}. In fact we must choose

$$B_{k-1} \equiv \frac{1}{\gamma_k} L_k^{-T} B_o G_k L_k^{-1} . \tag{5.7}$$

This gives us

$$\bar{B}_k^{-1} g_k = \gamma_k G_k^{-1} B_o^{-1} g_k = c_k , \tag{5.8}$$

since c_k is given by (4.32b) for $H_o = B_o^{-1}$. With this and (4.33a) relating b_k to $A_k d$ we can easily obtain (5.6).

Thus by substituting (5.6) and (5.8) into (5.4), the search direction d_k for the $O(n^2)$ algorithm becomes identical to the search direction determined by the $O(n)$ algorithm. Hence, clearly, we see that under the appropriate choice (5.7) the $O(n)$ algorithm can be interpreted as an $O(n^2)$ algorithm, in which a fixed preconditioning matrix B_o is updated at each step.

5.2 The VSOn Algorithm

As we mentioned at the beginning of this thesis, one main objective is to derive an algorithm which is a combination of the conjugate direction and variable metric algorithms based on conics. Specifically, by combining the $O(n)$ and $O(n^2)$ algorithms, we obtain a new algorithm which we will refer to as a variable storage $O(n)$ (VSOn) algorithm, since it can also use, as the VSCG algorithm, a variable amount of storage. So this algorithm can be viewed as an extension of the VSCG algorithm to the algorithms based on conics.

We will need some notation in order to describe and investigate the VSO_n algorithm. First, according to the update formula (5.3), by replacing B_i^{-1} by H_i, we may denote the update by $U(\bar{H}_i, i)$ where

$$U(H, i) \equiv H - \frac{HA_i dd^T + dd^T A_i H}{d^T A_i d} + \left(1 + \frac{d^T A_i H A_i d}{d^T A_i d}\right)\frac{dd^T}{d^T A_i d} . \qquad (5.9)$$

We have $\bar{H}_i \equiv L_i^{-1} H_{i-1} L_i^{-T}$ and $d \equiv d_{i-1}$. Note that the matrix L_i^{-1} can be easily given in the following form:

$$L_i^{-1} = \frac{\gamma_i}{\gamma_{i-1}} [I - \frac{t_i d a_o^T}{\gamma_{i-1}}] . \qquad (5.10)$$

By using (4.33a), (4.33b) and (5.10) we can get the following equivalent update formula to (5.9).

$$U(H, i) \equiv H - \frac{H b_i d^T + d b_i^T H}{b_i^T d} + \left[\frac{\gamma_{i-1}\bar{\gamma}}{\gamma_i^2} + \frac{b_i^T H b_i}{b_i^T d}\right] \frac{dd^T}{b_i^T d} , \qquad (5.11)$$

which is again closer in form to the BFGS update (3.13a).

We will now describe a single iteration of the VSOn algorithm. First the common steps from one to eight of the $O(n)$ and $O(n^2)$

algorithms are repeated for the VSOn algorithm. The only distinction will be at step 9, as we will see. Suppose that we begin at x_o and a positive definite matrix H_o is given. Then the VSOn algorithm consists of two parts, as follows.

<u>$O(n^2)$ - part</u>: Iterate (together with step 1 - step 8) for $i = 1,2,\ldots,m$:

$$H_i = U(\bar{H}_i, i) \quad , \tag{5.12a}$$

$$d_i = -H_i g_i \quad . \tag{5.12b}$$

Store some appropriate vectors at each step [we will see later which exactly], instead of thw whole matrix H_i.

<u>$O(n)$ - part</u>: From the point x_{m+1} reached by the $O(n^2)$ - part, and using the fixed matrix H_m as a preconditioner, iterate for $i = m+1\, , m+2\, , \ldots ,$

$$H_{i-1} \equiv \gamma_i L_i G_i^{-1} H_m L_i^T \quad , \tag{5.13a}$$

$$H_i = U(H_{i-1}, i) \quad , \tag{5.13b}$$

$$d_i = -H_i g_i \quad . \tag{5.13c}$$

As can be seen, the two parts differ only in the definition of H_i. In the first one we update H_i from the previous matrix H_{i-1}, until some storage limit is reached; subsequently, the $O(n)$ - part is the implementation of the $O(n)$ algorithm in the form described in Section 5.1.

We will now show how $H_1, \ldots, H_m$ can be stored. First, we observe that the matrices H_i are not themselves needed, but only products of the form $H_i v$ for $v \in \mathbb{R}^n$ are required. Therefore consider the equation derived from (5.12a) or (5.13b):

$$H_i v = U(H,i)v = Hv - \frac{Hb_i(d^T v) + d(b_i^T Hv)}{b_i^T d}$$

$$+ \left[\frac{\gamma_{i-1}\bar{\gamma}}{\gamma_i^2} + \frac{b_i^T H b_i}{b_i^T d} \right] \frac{d^T v}{b_i^T d} d \,, \qquad (5.14)$$

where H is either $\bar{H}_i$ or H_{i-1} according to whether H_i is defined by (5.12a) or (5.13b); i.e. H is defined either as:

$$H \equiv L_i^{-1} H_{i-1} L_i^{-T} \quad , \qquad (5.15)$$

or

$$H \equiv \gamma_i L_i G_i^{-1} H_m L_i^T \quad . \qquad (5.16)$$

Then it is clear, (by using also the definitions of the matrices L_i, L_i^{-1} and G_i^{-1}) that for each update we have to store the vectors b_i and d_i (2n locations) and the scalars $b_i^T d$, γ_i, $\bar{\gamma}$, t_i and $b_i^T H b_i$ (5 locations) to be able to compute $H_i v$. Also we have to compute vectors of the form Hv, so according to (5.15) or (5.16) we must first compute $L_i^{-T} v$ or $L_i^T v$, and then we apply the same technique recursively to get H_{i-1} or $H_m v$. But in order to compute $L_i^{-T} v$ or $L_i^T v$, we need also to store the horizon vector a_o and the initial direction d_o. Hence if we choose $H_o = I$, the total storage needed for $H_1, \ldots, H_m$ is $m(2n + 5) + 2n$. Provided $m << n$ this represents a substantial storage saving.

In conclusion, we have described a conic algorithm which allows the use of a variable amount of storage. And since more storage should mean better performance, we hope that it will improve these based on conics.

5.3 Conclusions

We have described and investigated a new class of algorithms for the unconstrained minimization problem (1.1). These are based on conics, and were first proposed by Davidon. We have described the basic members of this class, namely those introduced by Davidon and by Gourgeon and Nocedal and have demonstrated the relation between them. The role of the "reference point" in these algorithms has also been clarified. Finally we have introduced a new algorithm which combines them, and has variable storage requirements. Its derivation was based on the same idea as the derivation of the VSCG algorithm for quadratics introduced by Buckley and LeNir. The conic algorithms, which are the main ones investigated in this thesis, are designed to make use of more information for general functions, and so we hope that these will have better performance than the ones based on quadratics.

REFERENCES

[1] Axelsson, O., "On preconditioning and convergence acceleration in sparse matrix problems", CERN Data Handling Division Report 74-10, (1974).

[2] Broyden, C.G., "Quasi-Newton methods and their application to function minimization", Mathematics of Computation, 21(1967), pp. 368-381.

[3] Buckley, A.G., "Conjugate gradients, finite termination and linear constraints", MIS Technical Report No. 82-3, The University of Arizona (1982).

[4] Buckley, A.G. and LeNir, A., "QN-like variable storage conjugate gradients", to appear, Mathematical Programming (1983).

[5] Davidon, W.C., "Variable metric method for minimization", A.E.C. Research and Development Report, ANL-5990 (1959).

[6] Davidon, W.C., "Conjugate directions for conic functions", in Nonlinear Optimization 1981, M.J.D. Powell, ed., Academic Press, London.

[7] Davidon, W.C., "Conic approximations and collinear scalings for optimizers", SIAM J. on Numerical Analysis, 17 (1980).

[8] Fletcher, R., "Practical Methods of Optimization", Volume 1, John Wiley and Sons (1980).

[9] Fletcher, R. and Powell, M.J.D., "A rapidly convergent descent method for minimization", Computer Journal, 7(1963).

[10] Fletcher, R., and Reeves, C.M., "Function minimization by conjugate gradients", Computer Journal, 7(1964).

[11] Gourgeon, H. and Nocedal, J., "A Conic algorithm for optimization", Internal Report, Columbia University (1982).

[12] Hestenes, M.R. and Stiefel, E., "Methods of conjugate gradients for solving linear systems", J. Res. Nat. Bureau Standard, 49, (1959).

[13] Luenberger, D., "Introduction to Linear and Nonlinear Programming", Addison-Wesley (1973).