INFORMATION TO USERS

This material was produced from a microfilm copy of the original document. While the most advanced technological means to photograph and reproduce this document have been used, the quality is heavily dependent upon the quality of the original submitted.

The following explanation of techniques is provided to help you understand markings or patterns which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting thru an image and duplicating adjacent pages to insure you complete continuity.

2. When an image on the film is obliterated with a large round black mark, it is an indication that the photographer suspected that the copy may have moved during exposure and thus cause a blurred image. You will find a good image of the page in the adjacent frame.

3. When a map, drawing or chart, etc., was part of the material being photographed the photographer followed a definite method in "sectioning" the material. It is customary to begin photoing at the upper left hand corner of a large sheet and to continue photoing from left to right in equal sections with a small overlap. If necessary, sectioning is continued again — beginning below the first row and continuing on until complete.

4. The majority of users indicate that the textual content is of greatest value, however, a somewhat higher quality reproduction could be made from "photographs" if essential to the understanding of the dissertation. Silver prints of "photographs" may be ordered at additional charge by writing the Order Department, giving the catalog number, title, author and specific pages you wish reproduced.

5. PLEASE NOTE: Some pages may have indistinct print. Filmed as received.

University Microfilms International
300 North Zeab Road
Ann Arbor, Michigan 48106 USA
St. John's Road, Tyler's Green
High Wycombe, Bucks, England HP10 8HR
NOCEDAL, JORGE MONCADA
ON THE METHOD OF CONJUGATE GRADIENTS FOR
FUNCTION MINIMIZATION.

RICE UNIVERSITY, PH.D., 1978
RICE UNIVERSITY

ON THE METHOD OF CONJUGATE GRADIENTS FOR
FUNCTION MINIMIZATION

by

JORGE NOCEDAL

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Thesis Director's Signature:

Houston, Texas
MAY, 1978
ON THE METHOD OF CONJUGATE GRADIENTS FOR
FUNCTION MINIMIZATION

by

Jorge Nocedal

ABSTRACT

Two families of algorithms that combine conjugate gradient steps and quasi-Newton updates are studied. The first consists in interleaving the two iterations, the second is a conjugate gradient method with respect to a metric that is constructed by quasi-Newton corrections. It is shown that the two families can be implemented so as to have quadratic termination and that for sufficiently accurate line searches they are N-step quadratically convergent.

Conjugate gradient algorithms that do not require line searches are analyzed. Some results concerning their use on combined methods are given.

Several particular algorithms are tested through a set of numerical examples.
ACKNOWLEDGEMENTS

My greatest gratitude is to Professor Richard Tapia for his guidance and encouragement during my years of graduate school. Dr. Larry Nazareth's help has been invaluable. In our numerous discussions I learned a great deal about the subject of this dissertation.

I am grateful to my fellow students Petter Bjørstad, from whom I learned much, and Richard Byrd who has been a constant source of intellectual stimulation and friendly advice. I am most grateful to Professor Gene Golub for his generosity and help in many ways.

This work was supported by the Universidad Nacional Autónoma de México and Banco de México. I would like to thank the Systems Optimization Laboratory, Stanford University, for providing me with computer facilities and Evelyn T. Morris for her excellent typing of the manuscript.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>II. CONJUGATE GRADIENT WITH ARBITRARY METRIC</td>
<td>4</td>
</tr>
<tr>
<td>III. RESTARTING</td>
<td>7</td>
</tr>
<tr>
<td>IV. CONJUGATE DIRECTION METHODS WITHOUT LINE SEARCHES</td>
<td>13</td>
</tr>
<tr>
<td>A. The Three-Term Recurrence Relation Method</td>
<td>16</td>
</tr>
<tr>
<td>B. The Multistep Conjugate Direction Method</td>
<td>17</td>
</tr>
<tr>
<td>C. The Gradient Prediction Method</td>
<td>22</td>
</tr>
<tr>
<td>D. Memoryless Quasi-Newton Methods</td>
<td>24</td>
</tr>
<tr>
<td>V. COMBINED METHODS</td>
<td>26</td>
</tr>
<tr>
<td>A. The Interleaved QN-CG Method</td>
<td>26</td>
</tr>
<tr>
<td>B. The Variable Storage Conjugate Gradient Method</td>
<td>42</td>
</tr>
<tr>
<td>VI. LOCAL CONVERGENCE</td>
<td>46</td>
</tr>
<tr>
<td>VII. RATES OF CONVERGENCE</td>
<td>61</td>
</tr>
<tr>
<td>VIII. AN APPLICATION TO MULTIPLIER METHODS</td>
<td>67</td>
</tr>
<tr>
<td>IX. NUMERICAL EXAMPLES</td>
<td>70</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>73</td>
</tr>
</tbody>
</table>
CHAPTER I

1. **Introduction**

Two of the most important classes of algorithms for function minimization are conjugate gradient methods and quasi-Newton methods. It is well known that quasi-Newton methods converge faster and do not require very accurate line searches. On the other hand, conjugate gradient methods do not require storage of a matrix and have proved themselves useful in large problems where the matrix of second derivatives is not sparse or where the structure of this matrix is unknown. We are interested in improving the speed of convergence of conjugate gradient methods by combining them with quasi-Newton methods. We will be studying algorithms that will use $O(k \cdot n)$ storage locations, where $1 < k \leq n$. In this class one extreme is the conjugate gradient and the other the quasi-Newton algorithms. Two such methods have been studied in [14] and [19].

Conjugate gradient algorithms that do not require line searches have been recently introduced in [1], [2], [3] and [4]. They will be analyzed and shown to retain quadratic termination when used in conjunction with quasi-Newton updates. Let $f: \mathbb{R}^n \to \mathbb{R}$ be a differentiable function defined on an open set $D$. We consider the problem of finding $x^* \in D$ such that

$$\min_{x \in D} f(x) = f(x^*) \quad (1.1)$$

We denote the gradient of $f$ at $x_k$ by $g(x_k)$ or $g_k$. Let $y_k = g_{k+1} - g_k$ and $s_k = x_{k+1} - x_k$. 

- 1 -
The quasi-Newton methods we are mainly concerned with are those of Broyden's $\beta$-class [8]:

Given $x_0$ and an nxn matrix $H_0$, we let for $k=1,2,...$

$$x_{k+1} = x_k - \alpha_k H_k g_k$$

where $\alpha_k$ is a steplength and $H_k$ is defined recursively by

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k} + \beta_k w_k w_k^T$$

$$w_k = \frac{\alpha_k}{y_k^T H_k y_k} y_k^T s_k - \frac{\alpha_k}{s_k^T y_k} s_k^T y_k$$

with $\beta_k \geq 0$. The conjugate gradient method originally developed by Hestenes and Stiefel [5] to solve systems of linear equations has been adapted to solve the minimization problem (1) in the following way:

Given $x_0$, let $d_0 = -g(x_0)$

For $k = 1,2,...$

$$d_k = -g(x_k) + \beta_k d_{k-1}$$

$$x_{k+1} = x_k + \alpha_k d_k$$

where $\alpha_k$ minimizes $f$ in the direction $d_k$. The parameter $\beta_k$ gives rise to the following algorithms

$$\beta_k = \begin{cases} \frac{\|g_k\|^2}{\|g_{k-1}\|^2} & \text{Fletcher-Reeves [6]} \\ \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2} & \text{Polak-Ribiere [7]} \\ \frac{g_k^T y_{k-1}}{d_{k-1}^T y_{k-1}} & \text{"Hestenes-Stiefel"} \end{cases}$$ (1.5)
For quadratic functions the three formulas for $\beta$ are the same. Let $f$ be a strictly convex quadratic function defined on $\mathbb{R}^n$

$$f(x) = \frac{1}{2} x^T C x + b^T x$$

(1.6)

where $C$ is a symmetric and positive definite $n \times n$ matrix and $b \in \mathbb{R}^n$. A set of nonzero vectors $\{d_1, \ldots, d_n\}$ is said to be conjugate with respect to $C$ if $d_i^T C d_j = 0$ for $i \neq j$. An algorithm $M$ is said to have the quadratic termination property if the following is true: when $M$ is applied to the quadratic function (1.6) starting from any vector $x_0$, it generates a sequence which converges to the solution $x^*$ in at most $n$-steps. That is, $0 < \ell < n$ such that $x_{\ell} = x^*$. We will denote the space spanned by the vectors $a_1, a_2, \ldots a_n$ by $[a_1, \ldots a_n]$.

For the iteration

$$x_{k+1} = x_k + \alpha_k d_k$$

we will say that $\alpha_k$ is an exact minimizer if

$$f(x_k + \alpha_k d_k) < f(x_k + t d_k) \quad \text{for all } t \in [0, \infty).$$

- 3 -
CHAPTER II

CONJUGATE GRADIENT WITH ARBITRARY METRIC

Let $f$ be the strictly convex quadratic function (1.6)

$$f(x) = \frac{1}{2} x^T C x + b^T x$$

and let $H$ be symmetric and positive definite. $H$ can be factorized in various ways, for example as $LL^T = H$, where $L$ is lower triangular and nonsingular. Let $Z = L^{-1} x$ and define

$$h(Z) = f(LZ) = f(x) = \frac{1}{2} Z^T L^T C L z + b^T L z.$$

Then

$$h'(Z) = L^T C L z + L^T b = L^T g(x).$$

So that if $h'(Z^*) = 0$ then $g(LZ) = 0$. Consider applying the conjugate gradient method, say Fletcher-Reeves, to $h$.

$$Z_{k+1} = Z_k + \alpha_k d_k^y \quad \therefore \quad x_{k+1} = x_k + \alpha_k L d_k^y.$$

Define $L d_k^y \overset{\Delta}{=} d_k^x$. If $\alpha_k$ minimizes $h$ in the direction $d_k^y$ then it also minimizes $f$ in that direction. Using (2.1) and (2.2) we see that the conjugate gradient iterates in the $x$-variable are:

$$d_0^x = L d_0^y = - LL^T g_0 = - H g_0$$

$$\beta_k^y = g_k^T H g_k / g_{k-1}^T H g_{k-1}$$

$$d_k^x = - H g_k + \beta_k^y d_{k-1}^x$$

We call (2.3) the \textit{conjugate gradient method with metric $H$}. Whereas for the ordinary conjugate gradient method we have

$$g_i^T g_j = 0 \quad \text{for} \quad i \neq j \quad i, j \in \{1, 2, \ldots n\}$$
for the iteration (2.3) we have

\[ g_i^T H g_j = 0 \quad \text{for} \ i \neq j \]  

(2.4)

Now let \( \{ \lambda_1, \ldots, \lambda_n \} \) be the eigenvalues of \( C \). Luenberger [9] showed that for the conjugate gradient method

\[ E(x_m) \leq \frac{(\lambda_{n-m} - \lambda_1)^2}{(\lambda_{n-m} + \lambda_1)^2} \cdot E(x_0) \quad 0 < m < n \]  

(2.5)

where \( E(x) = \frac{1}{2}(x - x^*)^T C(x^* - x) \). For the function \( h \) defined in (2.1) we have the estimate

\[ T(Z_m) \leq \frac{(\sigma_{n-m} - \sigma_1)^2}{(\sigma_{n-m} + \sigma_1)^2} T(Z_0) \]  

(2.6)

where \( \{ \sigma_1, \ldots, \sigma_n \} \) are the eigenvalues of the matrix \( L^T C L \) and

\[ T(Z) \triangleq \frac{1}{2}(Z - Z^*)^T L^T C L (Z - Z^*) \]. Note that \( T(Z) = E(x) \), so that the conjugate gradient method with metric \( H \) satisfies the estimate (2.6).

If \( H \) resembles \( C^{-1} \) we can expect that \( L^T C L \) "approaches" the identity and so the function values will decrease faster. Suppose that \( H \) was obtained from a quasi-Newton iteration (1.2) – (1.3). It is well known that

\[ H y_j = s_j \quad j = 1, 2, \ldots, m \]  

(2.7)

where \( m \) is the number of iterations performed. Hence

\[ H C s_j = L L^T C s_j = s_j \quad j = 1, 2, \ldots, m \]  

(2.8)

\[ L^T C L (L^{-1} s_j) = L^{-1} s_j \quad j = 1, 2, \ldots, m \]  

(2.9)

The vectors \( \{ L^{-1} s_j \} \) are linearly independent so that \( L^T C L \) has \( m \) unit eigenvalues. The conjugate gradient with metric \( H \) would find the solution
in $n-m$ steps. Although this result is only true for quadratics it
should motivate the use of the conjugate gradient iteration (2.3) where
the metric was obtained by using quasi-Newton updates. Note that the
above argument also holds for Davidon's recent methods. [10], [11].

The conjugate gradient method with arbitrary metric has been used
for solving positive definite and symmetric systems of linear equations,
see [36] and [37]. In some cases the metric can be chosen so as to
produce a dramatic increase in the efficiency of the CG method.
CHAPTER III
RESTARTING

Conjugate gradient methods are usually implemented with restarts. Every n steps the search direction is set equal to the negative gradient. This greatly improves the numerical performance of the algorithm. In fact, Cohen [12] showed that several restarted conjugate gradient methods have n-step Q-quadratic convergence. If the methods are not restarted the convergence is only linear.

Beale [13] proposed a method that uses a restart direction different than the negative gradient. It has been studied by several authors who thought that one could increase the immediate reduction in the function value by using another starting direction. Crowder and Wolfe [16] concluded that Beale's method is not efficient; whereas Powell [17] came to the opposite conclusion.

Let \( d_o \neq 0 \) be an arbitrary vector, Beale's method is given by

\[
x_{k+1} = x_k + \alpha_k d_k
\]

where \( \alpha_k \) is the exact minimizer

\[
d_k = -g_k + \beta_k d_{k-1} + \gamma_k d_o \quad \text{for } k = 1, 2, n-1.
\]

\[
x_{k+1} = x_k + \alpha_k d_k
\]

\( \alpha_k \) is the exact minimizer

\[
\beta_k = g_k^T y_{k-1} / d_{k-1}^T y_{k-1}
\]

(3.1)

\[
\gamma_k = \begin{cases} 
    g_k^T y_o / d_o^T y_o & \text{if } k > 1 \\
    0 & \text{if } k = 1
\end{cases}
\]

Equations (3.1) define a cycle of n-steps. When the cycle is finished a new direction \( d_o \) is defined and a new cycle is started. Note that
If \( d_o \) is the negative gradient we obtain the usual conjugate gradient method with restarts.

It is easy to show that when applied to the quadratic function (1.6) Beale's method generates conjugate directions and therefore finds the solution in at most \( n \)-iterations, regardless of the choice of \( d_o \neq 0 \).

Also we have that

\[
d_1 = g_1 + \beta_1 d_o = g_o + \alpha_0 C d_o + \beta_1 d_o \implies d_1 \in [g_o, d_o, C d_o]
\]

\[
d_2 = g_2 + \beta_2 d_1 + \gamma_2 d_o = g_o + \gamma_o (\alpha_0 d_o + \alpha_1 d_1) + \beta_2 d_1 + \gamma_2 d_o
\]

\[= g_o + \alpha_0 C d_o + \alpha_1 g_o + \alpha_0 \alpha_1 C^2 d_o + \alpha_1 \beta_1 C d_o + \beta_2 g_o + \beta_2 \alpha_0 C d_o + \beta_2 \beta_1 d_o + \gamma_2 d_o
\]

\[\implies d_2 \in [g_o, C g_o, d_o, C d_o, C^2 d_o]
\]

\[\vdots
\]

\[\implies d_k \in [g_o, C g_o, \ldots, C^{k-1} g_o, C d_o, \ldots, C^k d_o].
\]

Therefore \([d_o, \ldots, d_k] \in [g_o, C g_o, \ldots, C^{k-1} g_o, C d_o, \ldots, C^k d_o]\] (3.2)

Let \( E(x_k) \) be defined as in (2.5) and suppose that

\[
0 < a \leq \lambda_1 \leq \lambda_2 \ldots \leq \lambda_{n-k} \leq b
\]

(3.3)

\[
b < \lambda_{n-k+1} \leq \lambda_{n-k+2} \leq \ldots \leq \lambda_n
\]

where the \( \lambda \)'s are the eigenvalue of \( C \).
Theorem 3.4  For any \( x_0 \in \mathbb{R}^n \), let \( x_1, x_2 \ldots \) be generated by Beale's method when applied to the quadratic function (1.6). Assume that \( d_o \) is such that
\[
[d_o, \ldots, d_k] = [g_0, Cg_0, \ldots, C^{k-1}g_0, d_o, Cd_o, \ldots, C^kd_o]
\]
for \( 1 \leq k \leq n \), then
\[
E(x_{k+1}) \leq \frac{(b-a)^2}{(b+a)}[E(x_0) + d_o^T C(x_o - x^* + \frac{1}{2}d_o)]
\]

Proof:  Beale's method is a conjugate direction method, therefore by Theorem 4.3 it minimizes the function \( f \) (also the function \( E \)) over the space \([d_o, \ldots, d_k]\). Consider the iteration
\[
x_{k+1} = x_o + p_{k-1}(C)g_o + s_k(C)d_o
\]
(3.5)

where \( p_{k-1} \) and \( s_k \) are polynomials in \( C \). Then the polynomials that minimize \( E(x_{k+1}) \) are those obtained by Beale's method. Now
\[
(x_{k+1} - x^*) = (x_o - x^*) + p_{k-1}(C)C(x_o - x^*) + s_k(C)d_o
\]
(3.6)
\[
= [I + p_{k-1}(C)C](x_o - x^*) + s_k(C)d_o
\]

Choose
\[
s_k = (I + p_{k-1}(C)C).
\]
(3.7)

We have
\[
E(x_{k+1}) \leq \frac{1}{2}[(x_o - x^*) + d_o]_o^T[I + p_{k-1}(C)C][I + p_{k-1}(C)C][(x_o - x^*) + d_o]
\]
\[
= \frac{1}{2}(x_o - x^*)_o^T C[I + p_{k-1}(C)C]^2(x_o - x^*) + d_o^T C[I + p_{k-1}(C)C]^2(x_o - x^*)
\]
\[
+ \frac{1}{2}d_o^T C[I + p_{k-1}(C)C]^2 d_o
\]
(3.8)
Let \( \{ e_i \} \) be an orthonormal set of eigenvalues of \( C \) and let

\[
(x_o - x^*) = \sum_{i=1}^{n} \xi_i e_i \quad \text{and} \quad p_{k-1}(C) = \alpha_0 I + \alpha_1 C + \ldots + \alpha_{k-1} C^{k-1} \tag{3.9}
\]

Then

\[
p_{k-1}(C)(x_o - x^*) = \alpha_0 + \sum_{i=1}^{n} \xi_i e_i + \ldots + \alpha_{k-1} \sum_{i=1}^{n} \xi_i e_i \lambda_i^{k-1} \tag{3.10}
\]

\[
[I + Cp_{k-1}(C)](x_o - x^*) = \sum_{i=1}^{n} \xi_i e_i + \alpha_0 \sum_{i=1}^{n} \xi_i e_i \lambda_i + \ldots + \alpha_{k-1} \sum_{i=1}^{n} \xi_i e_i \lambda_i^{k-1} \tag{3.11}
\]

\[
= \sum_{i=1}^{n} \xi_i e_i (1 + \alpha_0 \lambda_i + \ldots + \alpha_{k-1} \lambda_i^{k-1})
\]

\[
p_{k-1}(C)[I + Cp_{k-1}(C)](x_o - x^*) = \alpha_0 \sum_{i=1}^{n} \xi_i e_i (1 + \alpha_0 \lambda_i + \ldots + \alpha_{k-1} \lambda_i^{k-1}) + \ldots + \alpha_{k-1} \sum_{i=1}^{n} \xi_i e_i \lambda_i^{k-1} (1 + \alpha_0 \lambda_i + \ldots + \alpha_{k-1} \lambda_i^{k-1}) \tag{3.12}
\]

\[
[I + Cp_{k-1}(C)]^2(x_o - x^*) = \sum_{i=1}^{n} \xi_i e_i + \alpha_0 \sum_{i=1}^{n} \lambda_i \xi_i e_i (1 + \ldots + \alpha_{k-1} \lambda_i^{k-1}) + \ldots + \alpha_{k-1} \sum_{i=1}^{n} \xi_i e_i \lambda_i^{k-1} (1 + \ldots + \alpha_{k-1} \lambda_i^{k-1}) \tag{3.13}
\]

\[
C[I + Cp_{k-1}(C)]^2(x_o - x^*) = \sum_{i=1}^{n} \xi_i e_i \lambda_i (1 + \alpha_0 \lambda_i + \ldots + \alpha_{k-1} \lambda_i^{k-1}) (1 + \alpha_0 \lambda_i + \ldots + \alpha_{k-1} \lambda_i^{k-1})
\]

\[
= \sum_{i=1}^{n} \xi_i e_i \lambda_i [1 + \lambda_i p_{k-1}(\lambda_i)]^2 \tag{3.14}
\]

\[
(x_o^T - x^*) C[I + Cp_{k-1}(C)]^2(x_o - x^*) = \sum_{i=1}^{n} \xi_i^2 \lambda_i [1 + \lambda_i p_{k-1}(\lambda_i)]^2
\]

Let \( d_o = \sum_{i=1}^{n} \beta_i e_i \), then

\[
-10-
\]
\[
\mathbf{d}_0^T \mathbf{C}[\mathbf{I} + \mathbf{Cp}_{k-1}(\mathbf{C})]^{-1}(\mathbf{x}_0 - \mathbf{x}_*^k) = \sum \xi_\lambda \beta_\lambda \lambda_\lambda [1 + \lambda_\lambda \mathbf{p}_{k-1}(\mathbf{C})]^2 \tag{3.15}
\]
\[
\mathbf{d}_0^T \mathbf{C}[\mathbf{I} + \mathbf{Cp}_{k-1}(\mathbf{C})]^{-1} \mathbf{d}_0 = \sum \xi_\beta \lambda_\beta [1 + \lambda_\beta \mathbf{p}_{k-1}(\mathbf{C})]^2 \tag{3.16}
\]
Therefore, for any \( \mathbf{p}_{k-1} \)
\[
E(\mathbf{x}_{k+1}) \leq \sum \left( \frac{1}{2} \xi_\lambda^2 + \xi_\beta \lambda_\lambda + \frac{1}{2} \beta_\lambda \right) \lambda_\lambda [1 + \lambda_\lambda \mathbf{p}_{k-1}(\mathbf{C})]^2 \]
\[
\leq \max_{\lambda_\lambda} [1 + \lambda_\lambda \mathbf{p}_{k-1}(\mathbf{C})]^2 \frac{1}{2} \sum (\xi_\lambda + \beta_\lambda)^2 \lambda_\lambda \tag{3.17}
\]
or alternatively
\[
E(\mathbf{x}_{k+1}) \leq \max_{\lambda_\lambda} [1 + \lambda_\lambda \mathbf{p}_{k-1}(\mathbf{C})]^2 \left[ E(\mathbf{x}_0) + \mathbf{d}_0^T \mathbf{C}(\mathbf{x}_0 - \mathbf{x}_*^k + \frac{1}{2} \mathbf{d}_0) \right] \tag{3.18}
\]
The rest of the proof follows as in Luenberger [9]. ///

Observe that Eq. (3.2)
\[
[d_0, \ldots, d_k] \subset [g_0, \mathbf{C}_0, \ldots, \mathbf{C}^{k-1}g_0, \mathbf{d}_0, \mathbf{C}_0d_0, \ldots, \mathbf{C}^kd_0] \tag{3.2}
\]
only shows that the first space is contained in the other. In order to use theorem 3.4 we have to see for what choices of \( d_0 \) equality holds.
Clearly \( d_0 = g_0 \) will do, but this gives the conjugate gradient method. Another possibility is that \( d_0 \) be an eigenvector of \( \mathbf{C} \). However, we don't know these eigenvectors. Nevertheless this suggest that we consider Beale's method with arbitrary metric \( \mathbf{H} \), where the directions are defined by
\[
\mathbf{d}_k = -\mathbf{H}g_k + \beta_k \mathbf{d}_{k-1} + \gamma_k \mathbf{d}_0
\]
\[
\beta_k = g_k^T \mathbf{H}y_{k-1} / \mathbf{d}_k^T y_{k-1} \tag{3.21}
\]
\[
\gamma_k = \begin{cases} 
T_k^\top H y_0/d_o T_y_0 & \text{if } k > 1 \\
0 & \text{if } k = 1
\end{cases}
\]  
(3.19)

In this case we have that

\[\begin{bmatrix} d_o, \ldots, d_k \end{bmatrix} \in [H g_o, (H C) H g_o, \ldots, (H C)^{k-1} H g_o, d_o, H C d_o, \ldots, (H C)^k d_o] . \]  
(3.20)

If \( H \) was obtained by a quasi-Newton update formula using \( d_o \), then

\[H y_0 = H C d_o = d_o \]  
(3.12)

so that \( d_o \) is an eigenvalue of \( H C \) and (3.20) will hold with equality. Theorem 3.4 will hold (except that the \( \lambda \) now are the eigenvalues of a different matrix, see (2.6)). Beale's method then takes advantage of the eigenvalue distribution of \( C \), which is a very desirable property.

In short one would implement the method as follows:

(a) Choose any \( d_o \neq 0 \).
(b) Find \( H \) satisfying \( H y_0 = d_o \), using some quasi-Newton update formula.
(c) Continue with Beale's with metric \( H \), given by (3.19).

Notice that if (3.21) holds then \( \beta_1 = 0 \) and therefore \( d_1 = -H g_1 \).

So we obtain the conjugate gradient method with metric \( H \). For inexact line searches or general nonlinear functions \( \beta_1 \neq 0 \) and a different algorithm will be obtained.
CHAPTER IV

CONJUGATE DIRECTION METHODS WITHOUT LINE SEARCHES

Consider the quadratic function (1.6) and let \( d_1, d_2, \ldots, d_r \) be conjugate with respect to \( C \). We construct two sequences \( \{\overline{x}_i\} \) and \( \{x_i\} \) obtained by minimizing along the directions \( d_i \) and by doing direct prediction, respectively. That is, given \( \overline{x}_0 = x_0 \), for \( i = 0, 1, \ldots \) let

\[
\overline{x}_{i+1} = \overline{x}_i + \alpha_i d_i \quad \text{where } \alpha_i \text{ is the exact minimizer of } f \quad \text{along } d_i
\]

(4.1)

\[
x_{i+1} = x_i + d_i.
\]

(4.2)

The following result is well known [9]:

**Theorem 4.3** There exists an \( \ell \), \( 1 \leq \ell \leq n \) such that \( \overline{x}_\ell = x^* \),

where \( g(x^*) = 0 \), and \( g(\overline{x}_i)^T d_j = 0 \) for \( j = 1, 2, \ldots, i-1 \), \( i = 1, 2, \ldots, \ell \).

Therefore \( \overline{x}_i \) minimizes \( f \) over \([d_1, d_2, \ldots, d_{i-1}]\).

From (4.1) and (4.2) we have for any \( 1 \leq i < \ell \)

\[
\overline{x}_{i+1} - x_{i+1} = \sum_{j=1}^{i} (\alpha_i - 1) d_j = \sum_{j=1}^{i} c_j d_j
\]

(4.4)

\[
\overline{g}_{i+1} - g_{i+1} = \sum_{j=1}^{i} c_j C d_j = \sum_{j=1}^{i} c_j y_j \quad \text{where } \overline{g}_{i+1} = g(x_{i+1})
\]

(4.5)

\[
\overline{g}_{i+1}^T d_i - g_{i+1}^T d_i = -g_{i+1}^T d_i = c_i y_i
\]

\[
c_i = g_{i+1}^T d_i / d_i^T y_i
\]

(4.6)
Hence, at any stage of the algorithm without line searches we can compute the point that would have resulted if exact line searches had been used

$\overline{x}_{i+1} = x_{i+1} - \sum_{j=1}^{i} \frac{g_{j+1}^T d_j}{d_j^T y_j}.$  \hspace{1cm} (4.7)

We can also compute the gradient

$\overline{g}_{i+1} = g_{i+1} - \sum_{j=1}^{i} \frac{g_{j+1}^T d_j}{d_j^T y_j} y_j.$  \hspace{1cm} (4.8)

Starting from any $x_0$, if we obtain $\overline{x}_n$ by the iteration (4.2) and then compute $\overline{x}_n$ using (4.7) we will have that $\overline{x}_n = x^*$. The right most term of (4.7) can be updated at every iteration and the correction step should be applied at the end of the cycle only, in order not to introduce additional function evaluations.

Some of the algorithms that we will consider generate directions $d_i$ such that $d_i^T C d_j = 0$ \hspace{0.5cm} i \neq j$. When used in iteration (4.2) the following will hold: if $d_k = 0$ for some $k$ then $g_k \in \{g_0, g_1, \ldots, g_{k-1}\}$. That these algorithms have quadratic termination will follow from the following lemma.

**Lemma 4.9** Let $d_0, \ldots, d_t$ be linearly independent directions. Given $x_0$, let $x_1, \ldots, x_{t+1}$ be generated by iteration (4.2). Suppose that $g_{t+1}$ is linearly dependent upon $g_0, \ldots, g_t$. Then the solution $x^*$ lies in the subspace

$$v = \left\{ z : z = x_0 + \sum_{k=0}^{t} d_k \alpha_k \hspace{0.5cm} , \hspace{0.5cm} \alpha_k \in \mathbb{R} \right\}.$$  

- 14 -
Proof: Let \[ g_{t+1} = \sum_{k=0}^{t} \mu_k g_k \] 
(4.10)

For the quadratic (1.4) \[ g_k = g_o + \sum_{i=0}^{k-1} C d_i \] 
(4.11)

Then

\[
g_o + \sum_{i=0}^{t} C d_i = \sum_{k=0}^{t} \mu_k \left( g_o + \sum_{i=0}^{k-1} C d_i \right)
\]

\[
C^{-1} g_o - \sum_{i=0}^{t} d_i = \sum_{k=0}^{t} \mu_k C^{-1} g_o + \sum_{k=0}^{t} \sum_{i=0}^{k-1} \mu_k d_i
\]

But \( x^* - x_o = -C^{-1} g_o \), therefore

\[
\left(1 - \sum_{k=0}^{t} \mu_k \right) (x^* - x_o) = \sum_{i=0}^{t} d_i - \sum_{k=0}^{t} \sum_{i=0}^{k-1} \mu_k d_i
\]

\[
= \sum_{i=0}^{t} \left(1 - \sum_{k=1}^{t} \mu_k \right) d_i
\]

The coefficient of \( d_i \) is 1, therefore by linear independence

\[ 1 - \sum_{k=0}^{t} \mu_k \neq 0 \]. Hence

\[ x^* - x_o = \sum \delta_i d_i \] where \( \delta_i = \frac{1 - \sum_{k=1}^{t} \mu_k}{1 - \sum_{k=0}^{t} \mu_k} \) ///

Now we have to see how to generate the directions \( d_i \). First we consider Nazareth's method [2].
A. The Three-Term Recurrence Relation Method

We will call it the TTR method. It is given by:

\[ d_{-1} = 0 , \quad d_0 = -g_0 \]

\[ d_{j+1} = -y_j + \frac{y_j^T y_j}{y_j^T d_j} d_j + \frac{y_j^T y_{j-1}}{y_{j-1}^T d_{j-1}} d_{j-1} \]

\[ 0 \leq j \leq n-1 \]

\[ x_{j+2} = x_{j+1} + d_{j+1} \quad . \tag{4.12} \]

It is easy to see that with exact line searches and applied to the quadratic function (1.6) it is identical to the conjugate gradient method. One can also show that either

\[ d_j^T C d_i = 0 \quad i = 1,2,\ldots,j \quad \text{or} \quad d_{j+1} = 0 \quad . \]

The second case implies that \( g_{j+1} \in [g_0,\ldots,g_j] \). From Lemma 4.9 we conclude that if \( d_{j+1} = 0 \), then doing the correction step (4.7) from \( x_{j+1} \) one obtains \( x^* \). Therefore the TTR method has the quadratic termination property. Unlike the conjugate gradient method it can be started with any \( d_0 \neq -g_0 \) and we still have that \( d_i^T C d_j = 0 \quad \text{for} \quad i \neq j \). However, we cannot guarantee that for an arbitrary \( d_0 \) the directions generated are linearly independent; not even for quadratic functions as the following example shows.

**Example 4.13** Let \( f(x) = \frac{1}{2}(x^2 + 2y^2 + z^2) \), \( g(x)^T = (x,2y,z) \), \( x_0^T = (1,3,4) \) and \( d_0^T = (1,0,0) \). We have
$g_o^T = (1,6,4)$, $x_1^T = (2,3,4)$, $g_1^T = (2,6,4)$, $d_1 = 0$.

The solution $x^* = (0,0,0)$ is not in the span of $d_1$ and $d_2$.

We can define the TTR method with arbitrary metric. Following the development of section 2 we obtain

$$d_{j+1} = -Hy_j + \frac{y_j^THy_j}{y_j^Td_j}d_j + \frac{y_j^THy_{j-1}}{y_{j-1}^Td_{j-1}}d_{j-1} \quad 0 \leq j \leq n-1$$

$$x_{j+2} = x_{j+1} + d_{j+1}.$$

The directions satisfy $d_k \in [Hg_o, \ldots, Hg_{k-1}]$ and $d_1^T C d_j = 0$ for $i \neq j$.

Note that even for $d_o = -g_o$ the TTR does not necessarily generate descent directions for nonquadratic functions. Gill and Murray [34] have done computational experimentation with the TTR and obtained good results by doing a CG step whenever a non-descent direction was produced.

B. The Multistep Conjugate Direction Method [3]

We shall refer to it as the multistep method. It can be derived in a very natural way. We first ask that $d_k \in [g_1, \ldots, g_k]$ which we can write as

$$-G = DR$$

where $G \equiv (g_1, \ldots, g_k)$, $D \equiv (d_1, \ldots, d_k)$ and $R = (r_{ij})$ is an $k \times k$ upper triangular matrix. We may assume that $r_{ii} = 1$, $i = 1, \ldots, k$.

Then we demand conjugacy:

$$D^T C D = \bar{\alpha}$$

(4.15)
where \( \overline{\alpha} \) is a \((k \times k)\), nonsingular diagonal matrix. The algorithm will be

\[
x_{i+1} = x_i + \lambda_i d_i
\]

(4.16)

where \( \lambda_i \neq 0 \) is a certain steplength, possibly \( \lambda_i = 1 \) for all \( i \).

The quadratic function (1.6) satisfies

\[
g_{i+1} - g_i = C \lambda_i d_i \quad i = 1, 2, \ldots, k.
\]

(4.17)

Let \( t \) be the smallest integer such that \( g_{t+1} \in [g_1, \ldots, g_t] \). Let

\[
g_{t+1} = \sum_{j=1}^{t} \mu_j g_j.
\]

Then we may write (4.17) as

\[
Y \equiv A D \Lambda = GH
\]

(4.18)

where \( \Lambda = (\lambda_i) \) is a diagonal matrix and \( H \) is the \((t \times t)\) upper Hessenberg matrix:

\[
H = \begin{bmatrix}
-1 & & & \\
1 & -1 & & \\
& 1 & \ddots & \\
& & \ddots & -1 \mu_{t-1} \\
& & & 1 \mu_t
\end{bmatrix}.
\]

(4.19)

Therefore assuming: (i) that the directions generated are in the span of the previous gradients (4.14), (ii) that the directions are conjugate, and (iii) that the algorithm (4.16) is applied to a quadratic function (4.18) we obtain the system

\[
\begin{cases}
-G = DR \\
Y \equiv C D \Lambda = GH \\
D^T C D = \overline{\alpha}
\end{cases}
\]

(4.20)
where all matrices are of order \( (t \times t) \). From the way we defined \( t \) we know that \( \bar{\alpha} \), \( D \) and \( G \) are nonsingular. Therefore

\[
CDA = -DRH ,
\]

\[
D^TCD\Lambda = -(D^TD)(RH) ,
\]

\[
(D^TD)^{-1} = (RH)(\bar{\alpha}\Lambda)^{-1} .
\]

Now, \((D^TD)^{-1}\) is symmetric and \((RH)(\bar{\alpha}\Lambda)^{-1}\) is upper Hessenberg, hence \((RH)(\bar{\alpha}\Lambda)^{-1}\) is tridiagonal. We conclude that \(RH\) is tridiagonal. Using (4.21) and (4.22)

\[
Y^TY = (RH)^T(D^TD)(RH) = -\bar{\alpha}\Lambda
\]

so that \( Y^TY \) is tridiagonal.

From the fact that \(RH\) is tridiagonal it follows from (4.14) that

\[
(RH)_{ij} = r_{ij} - r_{i,j+1} = 0 \quad \text{for} \quad i+1 < j < t .
\]

Therefore \( r_{i,j} = r_{i,k} \) for \( i+1 < j < k \leq t \). The matrix \( R \) is then of the form

\[
R = \begin{bmatrix}
1 & r_{12} & \alpha & \alpha & \ldots & \alpha \\
1 & r_{23} & \beta & \beta & \ldots & \beta \\
1 & r_{34} & \gamma & \ldots & \gamma \\
\vdots & \vdots & \ddots & \ddots & \ddots \\
\vdots & \vdots & \ddots & \ddots & \ddots \\
1 & \ldots & \ldots & \ldots & \ldots & 1
\end{bmatrix}
\]

(4.25)
Using (4.14) and (4.15)

\[ g_k = a d_1 + \beta_2 + \cdots + r_{k-1,k} d_{k-1} + d_k \]

\[ \alpha = g_k^T y_1 / y_1^T d_1 \quad \beta = g_k^T y_2 / y_2^T d_2, \ldots, r_{k-1,k} = g_k^T y_{k-1} / d_{k-1} y_{k-1} \]

As \( Y^T Y \) is tridiagonal

\[ y_i^T y_{i+j} = 0 \quad \text{for} \quad j \geq 2 \]  

hence

\[ y_i^T y_{i+2} = y_i^T (g_{i+3} - g_{i+2}) = 0 \]

\[ y_i^T y_{i+3} = y_i^T (g_{i+4} - g_{i+3}) = 0 \]  

\[ \ldots \]

and so

\[ y_i^T g_{i+2} = y_i^T g_{i+3} = y_i^T g_{i+4} = \ldots \]

Equations (4.24) reduce to

\[ \alpha = g_3^T y_1 / y_1^T d_1 \quad \beta = g_4^T y_2 / y_2^T d_2, \ldots, r_{k-1,k} = g_k^T y_{k-1} / d_{k-1} y_{k-1} \]

(4.26)

The multistep conjugate direction method is

\[ d_1 = -g_1 \]

\[ p_{j+1} = -g_{j+1} + y_j^T g_{j+1} / y_j^T d_j \]

\[ d_{j+1} = p_{j+1} + c_j \]  

(4.27)
\[ c_j = \begin{cases} 
  c_{j-1} + \frac{y_{j-1}^T g_{j+1}}{y_{j-1}^T d_{j-1}} d_{j-1} & j > 1 \\
  c_1 = 0 
\end{cases} \]

When implemented without line searches and using the correction step (4.7), the algorithm will have quadratic termination. To see this note that \( d_{k=0} \implies g_k \in \{g_1, \ldots, g_{k-1}\} \) so that Lemma 4.9 can be applied. Observe also that for quadratic functions and exact line searches the algorithm reduces to the conjugate gradient method.

While developing his algorithm (see next section) Dixon [1] considered directions of the form

\[ d_k = -g_k + \sum_{i=1}^{k=1} b_i^k d_i \quad d_1 = -g_1 \]

For conjugacy to hold, we must have

\[ b_j^k = g_k^T y_j / y_j^T d_j \quad (4.28) \]

Furthermore in [20], Dixon shows that this iteration has n-step quadratic convergence. Notice that to compute \( d_{k+1} \) it is necessary to have \( d_1, \ldots, d_{k-1} \). So the algorithm use \( O(n^2) \) storage. By our previous observation, Eq. (4.25), one need not compute the terms \( b_j d_j \) anew every iteration, since

\[ b_j^k = b_j^{k-1} \quad \text{for } j = 1, \ldots, k-3 \]

The multistep algorithm can therefore be obtained from iteration (4.28) and the properties (4.25); and shows that only 3 vectors have to be stored
at the k-th step: \( d_k \), \( d_{k-1} \) and \( c_{k-1} \).

The generation of conjugate directions can be seen as a Gram-Schmidt process. The conjugate gradient method has the property that \( g_j \) is conjugate to \( d_1, d_2, \ldots, d_{j-2} \). As \( d_j \) is in the span of \( g_j \) and \( d_{j-1} \) it is conjugate to \( d_1, \ldots, d_{j-2} \) regardless of the choice of coefficients. The computation of only one coefficient insures that \( d_j \) is conjugate to \( d_{j-1} \). So the Gram-Schmidt orthogonalization of \( d_j \) with respect to the previous \( j-1 \) directions involves only one of these directions, namely \( d_{j-1} \). The coefficients of the others are zero. For inexact line searches the Multistep Method is the Gram-Schmidt orthogonalization of \( d_j \) using only \( d_{j-1} \) and \( d_{j-2} \). The coefficients of the previous \( d \)'s have already been computed in the past iteration.

When applied to non-quadratic functions the algorithm (4.27) does not necessarily generate descent directions. In a practical implementation of the algorithm we would set \( c_j = 0 \) if this occurs, obtaining a CG step.

C. The Gradient Prediction Method

This was the first conjugate direction method without line searches to be proposed [1]. We will call it the Gradient Prediction method or GP method. The idea is to generate without line searches directions parallel to the CG directions, for quadratic functions. Let \( \overline{d}_j \) be the directions generated by the CG method using exact line searches and by \( d_j \) the directions generated by an algorithm that doesn't use line searches. Let \( \overline{d}_0 = d_0 = -g_0 \). From Eqs. (4.7) and (4.8) we know that

\[
\overline{g}_1 = g_1 - \frac{g_1^T d_0}{d_0^T y_0} d_0
\]
so we define
\[ d_1 = g_1 + \beta(g_1, g_0)d_0 \]
where \( \beta(x, y) = \| x \|^2 / \| y \|^2 \) is the Fletcher-Reeves parameter (1.5). Therefore \( d_1 = \bar{d}_1 \). In general
\[ \bar{g}_{i+1} = \bar{g}_{i+1} - \sum_{j=1}^{i} \frac{g_{j+1}^T d_j}{d_j^T y_j} \]
\[ \bar{d}_{i+1} = -\bar{g}_{i+1} + \beta(\bar{g}_{i+1}, \bar{g}_i) d_i \] (4.30)

The algorithm is
\[ x_{i+1} = x_i + d_i \]
where \( d_i \) is given by (4.30). At the end of a cycle (say n-steps) one applies the correction (4.7)
\[ \bar{x}_{i+1} = x_{i+1} - \sum_{j=1}^{i} \frac{g_{j+1}^T d_j}{d_j^T y_j} d_j \]

The GP method moves along directions parallel to the CG directions therefore it is a descent algorithm for quadratic functions and has the quadratic termination property. More than just one algorithm the GP is a general approach to conjugate gradient methods without line searches. Formula (4.8) could be used for example to produce a method that generates parallel directions to Beale's method. Clearly the corrections (4.30) could be updated from one iteration to the other.
\[ w_{k+1} = w_k + \frac{g_{k+1}^T y_k}{d_k^T y_k} \]
\[ \bar{g}_{k+1} = g_{k+1} - w_{k+1} \] (4.31)
Notice that for non-quadratic functions and exact line searches the GP and the CG are identical.

D. Memoryless Quasi-Newton Methods

In [4] Shanno showed that one can generate a class of conjugate gradient algorithms from Broyden's $\beta$ class. Let us write the latter as

$$
\bar{H} = H - \frac{Hy^T}{y^T y} + \frac{ss^T}{s^T y} + \beta \left[ Hy - \frac{y^T y}{s^T y} s \right] \left[ Hy - \frac{y^T y}{s^T y} s \right]^T.
$$

The family of algorithms is defined by setting $H = I$ in (4.33) and defining

$$
d_k^\beta = -\bar{H}g_k.
$$

Now, if exact line searches are performed $s^T g = 0$, therefore Broyden's class gives the directions

$$
d = -Hg + \frac{Hy^T Hg}{y^T y} + \beta \left[ Hy - \frac{y^T y}{s^T y} s \right] y^T Hg.
$$

If $\beta = 0$ (DFP) one has

$$
d = -Hg + \frac{Hy^T Hg}{y^T y} s
$$

If $\beta = 1/y^T H y$ (BFGS) one has

$$
d = -Hg + \frac{y^T H g}{s^T y} s
$$

which is the conjugate gradient method with metric $H$. Shanno studies in [4] the algorithm given by (4.33) when $\beta = 1/y^T H y$ and $H = I$ for all
iterations. He scales his algorithm after observing that conjugate
gradient methods (with metric $H = I$) do not give reasonable steplengths.
We only note here that this scaling can be done quite naturally if some
$H$ is available. This will be discussed in Chapter 5, where an interest-
ing observation regarding the direction (4.36) will be given.

Two important properties of Shanno's method are that it generates
descent directions but does not have quadratic termination.
CHAPTER V
COMBINED METHODS

In this section we analyze two methods that use both quasi-Newton updates and conjugate gradient steps. They are the interleaved QN-CG method and the Variable Storage CG method. Both are designed for the case where less than \( n^2 \) locations available to store the QN matrix. The first performs QN steps and CG steps sequentially. The second does only CG steps, but with respect to an ever changing metric. This metric is defined through quasi-Newton updates.

A. The Interleaved QN-CG Method

It consists in doing QN and CG steps intermittently. We do the CG iterations with the metric defined in the previous QN cycle. The matrix is not updated during the CG iterations. The QN updates are defined by rank two corrections and they will not be stored in matrix form, but will be kept individually. We retain as many corrections as our storage capacity allows us. In Chapter 9 we discuss how long the QN and CG cycles should be and how to restart. In what follows it is only important to keep in mind that during each QN iteration the matrix is updated and during each CG iteration it is held fixed.

An interleaved QN-CG has been studied by Buckley. In [14] he states a theorem (see theorem 5.7 below), which we generalize slightly to show that it does not depend on the use of conjugate gradient iterations.

Consider the general iteration

\[
\begin{align*}
  d_j &= -Hg_j + \sigma_j^j d_{j-1} \\
  x_{j+1} &= x_j + \alpha_j^j d_j \\
  d_0 &= -Hg_0
\end{align*}
\]

\[ j = 1,2,\ldots \]  

(5.1)
where $H$ is a symmetric and positive definite matrix and $\alpha_j$ is a steplenent. Now consider the interleaved method that uses iteration (5.1) and $Q^N$ steps from Broyden's $\beta$ class (1.4). Let the points generated by this method be

$$x_0 = x_{0,1}, x_{0,2}, \ldots, x_{0,F}, x_{1,1}, x_{1,2}, \ldots, x_{1,F}, \ldots, x_{R,F}$$

where the $Q^N$ steps were performed at $x_{i,1}, i = 0, 1, \ldots, H_i$ is the new matrix obtained at $x_{i,1}$ and $d_{i,j}$ the displacement from $x_{ij}$ to $x_{i,j+1}$.

We use the following result of Powell [15] for the $\beta$ class.

**Theorem 5.1** Let $H_k$ be symmetric and positive definite, let $s_k \neq 0$ be in the space spanned by the columns of $H$ and let $y_k$ be such that $s_k^T y_k > 0$. Also assume that vector $w_k$ is nonzero. Then if

$$\beta > \frac{(y_k^T H_k y_k)(s_k^T y_k)^2}{(y_k^T H_k y_k)(s_k^T H_k s_k) - (s_k^T y_k)^2} \equiv \beta^*$$

we will have that $\text{rank } (H_{k+1}) = \text{rank } (H_k)$ and $H_{k+1}$ will be positive definite. $H^+$ is the generalized inverse of $H$. 

- 27 -
This result is important because it will ensure decent. It can be shown that

$$\beta \equiv t \beta_{\text{BFGS}} + (1-t) \beta_{\text{DFP}} > \beta^* \quad \text{for } t \in [0,1]$$  \hspace{1cm} (5.3)

where \( \beta_{\text{BFGS}} = \frac{y_k^T H y_k}{2} \alpha_k \) gives rise to the BFGS method and \( \beta_{\text{DFP}} = 0 \) to the DFP method.

Now we shall see that in the interleaved method one obtains positive definite matrices for any \( \beta \) satisfying (5.3).

**Lemma 1** Let \( H_i \) be symmetric and positive definite. Consider the interleaved method that uses QN updates and the general iteration (5.1). If sufficiently accurate line searches are performed (see below) one can choose \( \lambda \in \{1, 1, 2, \ldots, i, i, F-1\} \) such that the following is true: If \( s_{i, \lambda} \) and \( y_{i, \lambda} \) are used in (1.3) to obtain \( H_{i+1} \), and if \( w_i \neq 0 \), \( \beta > \beta^* \), then \( H_{i+1} \) will be positive definite.

**Proof:**

$$d_{i, \lambda} = -H_i s_{i, \lambda} + \sigma_{i, \lambda} d_{i, \lambda-1} = \sum_{j=1}^{\lambda} H_i g_{i, j} \tau_j$$  \hspace{1cm} (5.4)

for some \( \tau_j \) appropriately chosen. Therefore \( d_{i, \lambda} \) is in the column space of \( H_i \). Let us drop the first index, i.e. \( d_{i, \lambda} \equiv d_{\lambda} \). Now

$$d_{\perp} = -H g_{\perp}, \quad d_{\perp}^T g_{\perp} < 0$$

$$d_{\perp}^T g_{\perp} = -g_{\perp}^T H g_{\perp} + \tau \tau_{\lambda} d_{\lambda-1}^T g_{\perp}$$  \hspace{1cm} (5.5)
Let the accuracy of the search be such that

\[ |d_j^T g_{j+1}| < \min \left\{ \frac{1}{\tau_j} g_{j+1}^T H g_{j+1}, \ |d_j^T g_j| \right\} \quad \text{for all } j \quad (5.6) \]

Then from (5.5) \( d_k^T g_k < 0 \) and

\[ \alpha_k d_k^T \gamma_k = \alpha_k \left[ d_k^T g_{k+1} - d_k^T g_k \right] > 0 . \]

Therefore \( \delta_k^T \gamma_k > 0 \). We assume that \( \alpha_j > 0 \) for all \( j \). Applying Theorem 5.1 we conclude the proof. //.

The lemma tells us that we can use any of the general iteration steps to do the next quasi-Newton update and retain positive definiteness. In order to obtain termination for a quadratic we will have to use the last step.

Algorithm 1: Consider the interleaved method that uses the general iteration (5.1) and which satisfies the following:

(a) in the previous iteration to any QN step an exact line search is employed.

(b) For all other iterations we do sufficiently accurate line searches (in the sense of lemma 1).

(c) The QN updates use the last displacement vector.

The following theorem says that the intermediate steps do not undo the progress of the QN steps.

Theorem 5.7 If Algorithm 1 is applied to the quadratic function (1.6), starting from any \( x_0 \) then the minimum \( x^* \) will be reached after \( r \) QN steps, where \( 0 < r \leq n \).
Proof: Let \( S_{i,k} = \left\{ v: CH_i v = v, v^T H_i g_{i,j} = 0 \mid j = 1,2,...k \right\} \).

First we show that \( S_{i,k} - S_{i,k+1} \) for \( 1 \leq k < F_i \). \hspace{1cm} (5.8)

Let \( v \in S_{i,k} \)

\[
x_{i,k+1} = x_{i,k} + \sum_{j=1}^{k} \beta_j H_i g_{i,j}
\]

for some constants \( \beta_j \)

\[
g_{i,k+1} = g_{i,k} + \sum_{j=1}^{k} \beta_j C H_i g_{i,j}
\]

\[
v^T H_i g_{i,k+1} = v^T H_i g_{i,k} + \sum_{j=1}^{k} \beta_j g_{i,j}^T H_i C H_i v
\]

\[
= \sum_{j=1}^{k} \beta_j g_{i,j}^T H_i v = 0
\]

Therefore \( S_{i,k} \subseteq S_{i,k+1} \).

Next we will show that \( S_{i,F_i-1} \subseteq S_{i+1,1} \). To simplify the notation we will drop the subindex of \( F_i \), i.e., \( x_{i,F} = x_{i,F_i} \).

Let \( v \in S_{i,F-1} \) then

\[
CH_i v = v \hspace{1cm} v^T H_i g_{i,j} = 0 \hspace{1cm} j = 1,2,...F-1
\]

Let us call \( \delta = s_{i,F-1} \) and \( \gamma = y_{i,F-1} \)

\[
H_{i+1} = H_i - \frac{H_i \gamma \gamma^T H_i}{\gamma^T H_i \gamma} \delta \delta^T + \beta w w^T
\]

\[
w = \alpha \left[ \begin{array}{c} \frac{H_i \gamma}{\gamma^T H_i \gamma} - \frac{\delta}{\delta^T \gamma} \\ \frac{H_i \gamma}{\gamma^T H_i \gamma} - \frac{\delta}{\delta^T \gamma} \end{array} \right]
\]
\[
CH_{i+1}v = CH_{i}v - \frac{CH_{i}Y^{T}H_{i}v}{Y^{T}H_{i}Y} + \frac{C\delta^{T}v}{\delta^{T}v} + \bar{\beta}Cw\alpha\begin{bmatrix}
\gamma^{T}H_{i}v & -\delta^{T}v \\
\gamma^{T}H_{i}Y & \delta^{T}Y
\end{bmatrix}
\]

\[
= v - \frac{CH_{i}\delta^{T}CH_{i}v}{Y^{T}H_{i}Y} + \frac{C\delta^{T}v}{\delta^{T}v} + \bar{\beta}Cw\alpha\begin{bmatrix}
\delta^{T}CH_{i}v & -\delta^{T}v \\
\gamma^{T}H_{i}Y & \delta^{T}Y
\end{bmatrix}
\]

(5.9)

Now \(\delta^{T}CH_{i}v = \delta^{T}v\)

\[
\delta = \sum_{j=1}^{F-1} \beta_{j}H_{i}g_{i,j}
\]

therefore \(\delta^{T}v = 0\).

Using this in (5.9) we obtain

\[
CH_{i+1}v = v
\]

(5.10)

We only need to show that \(g_{i+1}^{T}H_{i}v = 0\)

\[
x_{i+1,1} = x_{i,F-1} + \sum_{j=1}^{F-1} \beta_{j}H_{i}g_{i,j}
\]

\[
g_{i+1,1} = g_{i,F-1} + \sum_{j=1}^{F-1} \beta_{j}CH_{i}g_{i,j}
\]

\[
g_{i+1}^{T}H_{i+1}v = g_{i,F-1}^{T}H_{i+1}v + \sum_{j=1}^{F-1} \beta_{j}g_{i,j}^{T}H_{i}CH_{i+1}v
\]

\[
= g_{i,F-1}^{T}H_{i+1}v + \sum_{j=1}^{F-1} \beta_{j}g_{i,j}^{T}H_{i}v
\]

(from 5.10)

\[
= g_{i,F-1}^{T}H_{i+1}v
\]

\[
= g_{i,F-1}^{T}C^{-1}CH_{i+1}v
\]

\[
= g_{i,F-1}^{T}C^{-1}CH_{i}v = 0
\]

(5.11)
Therefore $S_{i,F-1} \subseteq S_{i+1,1}$. Note that we haven't used the exact line search hypothesis.

Finally, we will show that if $x_{i,F}$ was obtained by doing an exact line search from $x_{i,F-1}$ then $S_{i,F-1} \subseteq S_{i+1,1}$. From the quasi-Newton equation, $\delta = H_{i+1}^{\gamma}$, therefore

$$CH_{i+1}^{\gamma} = C\delta = \gamma$$  \hspace{1cm} (5.12)

$$\delta^T g_{i+1,1} = 0 \quad \text{(exact search)}, \quad \text{then}$$

$$(H_{i+1}^{-1}\delta)^T H_{i+1} g_{i+1,1} = \gamma^T H_{i+1} g_{i+1,1} = 0 .$$  \hspace{1cm} (5.13)

Hence $\gamma \in S_{i+1,1}$. Assume that $\gamma \in S_{i,F-1}$:

$$g_{i,F-1}^T H_{i}^{\gamma} = 0 \quad CH_{i}^{\gamma} = \gamma$$

Therefore $g_{i,F-1}^T C^{-1}\gamma = g_{i,F-1}^T \delta = 0$.  \hspace{1cm} (5.14)

This is not possible; in the proof of Lemma 1 it is shown that $g_{j}^T d_{j} < 0$ for all $j$. This shows that $S_{i,F-1} \subseteq S_{i+1,1}$. So, the linear spaces $S_{i,k}$ are nondecreasing and their dimension increases by at least 1 after the QN-step. //

It is clear from the proof that we could do a QN after a step with inexact line searches and this would not destroy the termination property. Theorem 5.7 can be paraphrased as follows: after most $n$ QN steps that were preceded by exact-line search steps we will reach the minimum $x^*$. 

- 32 -
The BFGS update formula is

$$H_{k+1} = H_k + \frac{1}{s_k y_k} \left( 1 + \frac{y_k^T H_k y_k}{s_k y_k} - H_k y_k \right) s_k^T s_k^T - \frac{s_k y_k^T H_k}{s_k y_k}$$  \hspace{1cm} (5.15)

which we will write as

$$H_{k+1} = H_k + a_k s_k^T + b_k y_k^T H_k$$  \hspace{1cm} (5.16)

where the vectors $a_k$ and $b_k$ are found by comparing with (5.15).

Theorem (5.7) is a weak result; we shall now show that these iterations can be chosen so that (n-step) quadratic termination is obtained.

Algorithm 2 Consider the Interleaved method that uses BFGS steps and conjugate gradient iterations with respect to the newest $H'_{BFGS}$ matrix. An exact line search is performed at each step.

**Theorem 5.17** If Algorithm 2 is applied to the quadratic function (1.6) starting from any $x_0 \in \mathbb{R}^n$ and any symmetric and positive definite $H_0$, then the solution $x^*$ will be obtained in at most $n$ steps.

**Proof:** The Hestenes-Stiefel CG with metric $H$ is

$$d_j^{CG}(H) = -Hg_j + \begin{bmatrix} y_{j-1}^T H g_{j-1} \\ y_{j-1}^T s_{j-1} \end{bmatrix} s_{j-1}$$  \hspace{1cm} (5.18)

Let $x_{i,j}$ be a point generated by the algorithm. We define

$$d_{i,j} = \text{displacement generated by Algorithm 2 at } x_{i,j}$$

$$d_{i,1}^B = -H_{i} g_{i} \hspace{1cm} i = 2, 3, \ldots, R$$

$$d_{i,j}^{CG} = (5.18) \text{ evaluated at } x_{i,j}.$$
We are using here the same notation as in Theorem (5.7) to describe when the updates are done. We will show that $d_{i,j} = d_{i,j}^{CG}(H_o)$ for all $(i,j)$ in the sequence. In other words, the Interleaved method is equivalent to using the CG with metric $H_o$ throughout.

Assume that at the $(j-1)$ cycle:

1. $d_{s,k} = d_{s,k}^{CG}(H_o)$ \quad $s = 0, \ldots, j-1$ ; \quad $k = 1, \ldots, F_s$
2. $H_s g_{j-1,t} = H_o g_{i-1,t}$ \quad $s = 0, \ldots, j-2$ ; \quad $t = 1, \ldots, F_s$ \quad (5.19)
3. $H_{j-1} g_{j-1,t} = H_o g_{j-1,t}$ \quad $t = 2, 3, \ldots, F_{j-1}$

The assumption is clearly true for $j = 1$. Now we show that (5.19) holds for the $j$-th cycle. Recall that $g_{i-1,F_j-1} = g_{j,1}$

\begin{align*}
  d_{j,1}^B &= - H_j g_{j,1} = - H_j g_{j,1} - a_j - 1 g_{j-1,F_j-1,F_j-1}^{T} + b_j - 1 y_{j-1,F_j-1,F_j-1}^{T} \\
  &= - H_o g_{j,1} - b_j y_{j-1,F_j-1,F_j-1}^{T} + b_j - 1 y_{j-1,F_j-1,F_j-1}^{T} \\
  &= - H_o g_{j,1} + b_j - 1 y_{j-1,F_j-1,F_j-1}^{T} g_{j-1,F_j-1,F_j-1} \\
  &= d_{j,1}^{CG}(H_o) \quad \text{(see (5.15)-(5.16))}
\end{align*}

Using the conjugacy and orthogonality properties of the CG we have

\begin{align*}
  H_j g_{j,2} &= H_o g_{j,2} + a_o s_{j,F_j}^{T} g_{j,2} + b_o y_{o,F_j}^{T} h_{o,F_j} g_{j,2} \\
  &= H_o g_{j,2}
\end{align*}
Assume that $H_k g_{j,2} = H_0 g_{j,2}$ for $1 < k < j$, then

$$H_{k+1} g_{j,2} = H_k g_{j,2} + a_k s_k F g_{j,2} + b_k y_k F H_k g_{j,2}$$

$$= H_0 g_{j,2}.$$ 

Therefore $H_k g_{j,2} = H_0 g_{j,2}$ for $k = 1, 2, \ldots, j$. Now

$$d_{j,2}^{CG}(H_j) = - H_j g_{j,2} + \frac{y_{j,1}^T H_j g_{j,2}}{y_{j,1}^T d_{j,1}} d_{j,1}$$

$$= - H_0 g_{j,2} + \frac{y_{j,1}^T H_0 g_{j,2}}{v_{j,1}^T d_{j,1}} d_{j-1}$$

$$= d_{j,2}^{CG}(H_0).$$

Reasoning as before we can show that for $k = 1, 2, \ldots, j$

$$H_k g_{j,i} = H_0 g_{j,i} \quad \left\{ \begin{array}{l}
\text{i = 2, 3, \ldots, F_j} \\
\end{array} \right.$$

So that (5.19) holds for the (j-th) cycle. The proof now follows from

the quadratic termination of the CG method // .

If instead of BFGS one uses some other member of Broyden's $\beta$-class

(1.3) one does not obtain (n-step) quadratic termination:

- 35 -
Lemma 5.22  If in Algorithm 2 one uses a member of Broyden's Class (1.3) different than BFGS, then the quadratic termination property is lost.

Proof: We write (1.3) in a slightly different form

$$
\tilde{H} = H - \frac{H^T y}{y^T y} \frac{s^T}{y^T y} + \beta \left( \frac{H^T y}{y^T y} \right) \left( \frac{s^T}{y^T y} \right)^T
$$

(5.23)

where $\beta_{\text{BFGS}} = 1/y^T y$.

Suppose that $n-1$ CG steps with metric $H$ have been performed and that the solution has not been reached. From Theorem 5.17 it follows that if the solution is to be obtained in the next step, the new direction should be parallel to the BFGS. Let $s$ be the last displacement and $g_+$ the current gradient. Recall that we are doing exact line searches:

$$
s^T g_+ = 0
$$

$$
d_\beta = \tilde{H} g_+ = H g_+ - \frac{y^T H g_+}{y^T y} y^T y + \beta \left( \frac{y^T y}{y^T y} \right) y^T H g_+
$$

$$
= H g_+ + y^T y \left( \frac{y^T H g_+}{y^T y} + \beta y^T H g_+ \right) - s \beta \left( \frac{y^T y}{s^T y} \right) y^T H g_+
$$

$$
= H g_+ + y^T y \left( \frac{y^T H g_+}{y^T y} + \frac{y^T H g_+}{y^T y} \right) - s \left( \frac{y^T H g_+}{s^T y} \right)
$$

$$
= H g_+ - \frac{y^T H g_+}{s^T y} s
$$

(5.24)

$$
d_{\text{BFGS}} = H g_+ + y^T y \left( \frac{y^T H g_+}{y^T y} + \frac{y^T H g_+}{y^T y} \right) - s \left( \frac{y^T H g_+}{s^T y} \right)
$$

(5.25)

Suppose that $k \cdot d_{\text{BFGS}} = d_\beta$ for some number $k$. Then

$$
(k - 1)H g_+ + y^T y \left( \frac{y^T H g_+}{y^T y} - \beta y^T H g_+ \right) - s \left( \frac{y^T H g_+}{s^T y} \right) \left( k \cdot d_{\text{BFGS}} \right) = 0
$$

(5.26)
This equation has the form

\[ a_1 g_+ + a_2 H_g + a_3 s = 0. \]

As \( s = \sum a_i g_i \), where the summation is over all \( i \) such that \( g_i \)

preceeds \( g_+ \), and as the gradients are conjugate with respect to \( H \) we

have \( a_1 = a_3 = 0 \). But \( a_3 = 0 \) implies

\[ k = \beta y^T H y; \quad (5.27) \]

\[ a_1 = (k - 1) + \frac{y^T H g_+}{y^T H y} - \beta y^T H g_+ = 0; \]

\[ \beta y^T H y - 1 = \beta y^T H g_+ - \frac{y^T H g_+}{y^T H y}; \]

\[ \beta(y^T H y - y^T H g_+) = 1 - \frac{y^T H g_+}{y^T H y} = \frac{y^T H y - y^T H g_+}{y^T H y} \]

therefore \( \beta = \frac{1}{y^T H y} \) which is BFGS. //

The above results show a special relation between the CG and the

BFGS methods. The proof of Theorem 5.17 implies the following: If BFGS

and CG are applied to a quadratic function using the same initial matrix,

and if exact line searches are used then the displacements generated by

the two algorithms are the same in direction and magnitude. Huang [18]

showed that in this case all members of his class (which includes Broyden's)

generate parallel directions (except for sign). And if the initial matrix

is the identity these directions are parallel to the CG directions. BFGS

gives the same direction and magnitude as CG.
In a similar way to Theorem 5.17 one can show that DFP in conjunction with it is derived CG method (4.36) has quadratic termination. Moreover, for each member of Broyden's class one can find its dual as described in Chapter 4 Section D and obtain an Interleaved method with quadratic termination. However, except for the method derived from the BFGS, not much is known about the behavior of these conjugate gradient algorithms. Therefore we will not consider them in what follows.

We can avoid line searches in the conjugate gradient steps and retain quadratic termination by using the TTR method.

Algorithm 3 Consider the Interleaved method that uses BFGS and the TTR method in the following way: (a) at the end of a sequence of TTR steps, the correction (4.7) is done, and (b) every BFGS step is performed with an exact line search.

Theorem 5.30 If Algorithm 3 is applied to the quadratic function (1.6) starting from any \( x_o \in \mathbb{R}^n \) and any symmetric and positive definite matrix \( H_o \), then the solution will be obtained in at most \( n \)-steps. (As the correction step does not involve function evaluations it is not counted as a step).

Proof: First we will show that the directions generated by the TTR are parallel to the conjugate gradient directions if \( d_o = -g_o \).

We will denote by \( x_1^*, x_2^* \ldots \) the sequence generated by the CG and \( x_1, x_2, \ldots \) that produced by the TTR method.
\[ x_1^* = x_0 + \alpha_o d_o \]
\[ x_1 = x_0 + d_o \]

therefore
\[ g_1^* = g_1 + (\alpha_o - 1)y_o = \alpha_o y_o + g_o \quad ; \quad y_o^* = \alpha_o y_o \]
\[ y_o^{*T} g_1 = \alpha_o y_o^{T} [g_1 + \alpha_o y_o - y_o] \]
\[ y_o^{*T} d_o = \alpha_o y_o^{T} d_o \]

Using the Hestenes–Stiefel CG we have

\[ d_1^* = - \alpha_o y_o - g_o + \alpha_o \frac{y_o^{T} g_1 + y_o^{T} y_o \alpha_o - y_o^{T} y_o}{\alpha_o y_o^{T} d_o} d_o \]
\[ = - \alpha_o y_o + \left[ 1 + \frac{y_o^{T} g_1 + \alpha_o y_o^{T} y_o - y_o^{T} y_o}{y_o^{T} d_o} \right] d_o \]
\[ = - \alpha_o y_o + \left[ \frac{y_o y_o + \alpha_o y_o^{T} y_o - y_o^{T} y_o}{y_o^{T} d_o} \right] d_o \]
\[ = - \alpha_o y_o + \frac{y_o^{T} y_o}{y_o^{T} d_o} d_o = \alpha_o d_1 \]

where \( d_1, d_2, \ldots \) are given by (4.12).

Induction: Assume that \( d_j^* = \alpha_{j-1} d_j \), \( j = 1, 2, \ldots, k \).
From (4.8)

\[ g_{k+1}^* = g_{k+1} - \sum_{j=1}^{k} \frac{g_{j+1}}{d_j} y_j \]  

So

\[ y_k^* = y_k - \frac{g_{k+1}}{d_k} y_k \]

\[ y_k = y_k - \frac{g_k}{d_k} y_k \]

\[ y_k^* = \alpha_k y_k \quad \text{with} \quad \alpha_k = -\frac{g_k}{d_k} y_k \]

Now

\[ d_{k+1} = -\frac{1}{\alpha_k} y_k^* + \frac{1}{\alpha_k} y_k^* y_k^* d_k^* + \frac{1}{\alpha_k} y_k^* y_k^* d_k^* \]

\[ = \frac{1}{\alpha_k} \left[ -g_{k+1}^* + g_k^* + \frac{y_k^* y_k^*}{y_k^* d_k^*} d_k^* + \frac{y_k^* y_k^*}{y_k^* d_k^*} d_k^* \right] \]  

\[ d_k^* = -g_k^* + \frac{y_{k-1}^* g_k^*}{y_{k-1}^* d_{k-1}^*} d_{k-1}^* \]  

Also

\[ -d_k^* + \frac{y_k^* y_k^*}{y_k^* d_k^*} d_k^* = \left[ -\frac{y_k^* d_k^* + y_k^* y_k^*}{y_k^* d_k^*} \right] d_k^* \]
Using (5.32) and the orthogonality of gradients

\[
\begin{align*}
&= \frac{y_k^{\ast T} g_k^{\ast} + y_k^{\ast T} g_{k+1}^{\ast} - y_k^{\ast} g_{k+1}^{\ast}}{y_k^{\ast T} d_k^{\ast}} \frac{y_k^{\ast T} g_{k-1}^{\ast}}{y_k^{\ast T} d_k} d_k^{\ast} = \frac{y_k^{\ast T} g_{k-1}^{\ast}}{y_k^{\ast T} d_k} d_k^{\ast} \\
&= \frac{y_k^{\ast T} g_{k-1}^{\ast}}{y_k^{\ast T} d_k} d_k^{\ast} \\
&= \frac{y_k^{\ast T} g_{k-1}^{\ast}}{y_k^{\ast T} d_k} d_k^{\ast}
\end{align*}
\]

(5.33)

Substituting (5.32) in (5.31) and using (5.33) and orthogonality

\[
d_{k+1} = \frac{1}{\alpha_k} \left[ -g_{k+1}^{\ast} - d_k^{\ast} + \frac{y_{k-1}^{\ast T} g_k^{\ast}}{y_{k-1}^{\ast T} d_k} d_{k-1}^{\ast} + \frac{y_k^{\ast T} g_k^{\ast}}{y_k^{\ast T} d_k} d_k^{\ast} + \frac{y_{k-1}^{\ast T} y_k^{\ast}}{y_{k-1}^{\ast T} d_k} d_{k-1}^{\ast} \right]
\]

\[
= \frac{1}{\alpha_k} \left[ -g_{k+1}^{\ast} + \frac{y_k^{\ast T} g_{k+1}^{\ast}}{y_k^{\ast T} d_k} d_k^{\ast} \right] + \frac{1}{\alpha_k} \left[ \frac{y_{k-1}^{\ast T} g_k^{\ast}}{y_{k-1}^{\ast T} d_k} d_{k-1}^{\ast} + \frac{y_{k-1}^{\ast T} y_k^{\ast}}{y_{k-1}^{\ast T} d_k} d_{k-1}^{\ast} \right]
\]

\[
= \frac{1}{\alpha_k} \left[ -g_{k+1}^{\ast} + \frac{y_k^{\ast T} g_{k+1}^{\ast}}{y_k^{\ast T} d_k} d_k^{\ast} \right] = \frac{1}{\alpha_k} d_{k+1}^{\ast}
\]

Induction holds. Therefore CG and the TTR generate parallel directions.

After applying the correction step the TTR method produces the same point as the CG method. Now suppose that a cycle of TTR steps plus correction step is completed and that the matrix will be updated. Let s be the last step of the TTR method. Then \( s = \lambda s^{\ast} \) where \( s^{\ast} \) is the corresponding CG step and \( \lambda \) is a constant. Then \( y = Cs \), \( y^{\ast} = \lambda Cs \), so \( y = \frac{1}{\lambda} y^{\ast} \). Looking at the BFGS update formula one readily sees that it is equivalent to use \( (s,y) \) or \( (s^{\ast},y^{\ast}) \). Therefore using the TTR method is equivalent to using the CG method. The result now follows from Theorem 5.7. ///
The CG method generates descent directions, hence from the first part of the proof it follows:

**Corollary 5.34**  The TTR method generates descent directions when applied to quadratic functions.

It is clear that the same argument used in proving Theorem 5.30 could have been applied to the GP method or to the Multistep method.

**Corollary 5.35**  If in Algorithm 3 one uses the GP or the Multistep Method instead of the TTR method one obtains quadratic termination.

**B. Variable Storage Conjugate Gradient Method**

The second combined method that we will study consists of doing only conjugate gradient iterations. The metric will vary or stay the same from one iteration to the other, depending on the storage available. The matrix that defines the metric will be constructed using quasi-Newton updates. The algorithm, which we shall call the variable storage conjugate gradient method or VSCG suggested by Nazareth [19] is as follows:

Let \{x_1, x_2, \ldots \} be the points generated and \{H_1^1, H_1^2, H_1^3, \ldots \} the matrices. \(H_{j_2}^3\) indicates that it is the third matrix and that it was generated at \(x_{j_2}\).
\( d_{k}^{CG}(H) \) will denote a conjugate gradient direction at \( x_{k} \) using metric \( H \). Let \( T = \{ j_{1}, j_{2}, \ldots \} \) be the set of indices where updates are performed. The VSCG iteration is

\[
H_{k}^{r+1} = \begin{cases} 
U(H_{\lambda}^{r}, s_{r-1}, y_{r-1}) & \text{if } k \in T \\
\text{undefined for } k \notin T
\end{cases}.
\]

Here \( U \) denotes the update function of Broyden's \( \beta \)-class and \( \lambda \in T \) is the integer preceding \( k \) in \( T \).

\[
d_{k} = d_{k}^{CG}(H_{\lambda}^{r}), \\
x_{k+1} = x_{k} + \alpha_{k} d_{k}.
\]

Note that matrix used is not the most recent one, but the previous one.

**Theorem 5.36** Let \( H \) be any symmetric and positive definite matrix. Then the VSCG method with exact line searches, starting with \( H_{1}^{1} = H \) has the quadratic termination property.

**Proof:** \( d_{1} = d_{1}^{CG}(H), \ d_{2} = d_{2}^{CG}(H) \).

By the orthogonality properties of the CG:

\[
g_{3}^{T} H_{g_{2}} = g_{3}^{T} H_{g_{1}} = g_{2}^{T} H_{g_{1}} = 0
\]

\[
g_{3}^{T} d_{2} = g_{3}^{T} d_{1} = g_{2}^{T} d_{1} = 0.
\]
Assume that $d_k^{CG}(H_j^m) = d_k^{CG}(H)$ for $2 \leq k$ where $H_j^m \in \{H_1^1, H_2^2, \ldots, H_t^t\}$ and $t \leq k-1$. (That is, we are assuming that it is equivalent to use any of the previous matrices to do the step.)

Also assume that

$$\begin{cases} g_{k+1}^T H g_j = 0 \\ g_{k+1}^T d_j = 0 \end{cases} \quad j = 1, 2, \ldots, k$$

We write Broyden's formula as

$$H_{j+1} = H_j + a_j s_j^T + b_j y_j^T H_j.$$ 

Then

$$H_j^m g_{k+1} = H_{j-1}^{m-1} g_{k+1} + a_j^{m-1} s_j^{m-1} g_{k+1} + b_j^{m-1} y_j^{m-1} H_{j-1}^{m-1} g_{k+1}$$

$$H_{j-2}^{m-1} g_{k+1} = H_{j-2}^{m-2} g_{k+1} + a_j^{m-2} s_j^{m-2} g_{k+1} + b_j^{m-2} y_j^{m-2} H_{j-2}^{m-2} g_{k+1}$$

$$\vdots$$

$$H_1^2 g_{k+1} = H_1 g_{k+1} + a_1 s_1 g_{k+1} + y_1^T H_1 g_{k+1}.$$ 

By the induction hypothesis

$$H_1^2 g_{k+1} = H g_{k+1}, \ldots, H_{j-2}^{m-1} g_{k+1} = H g_{k+1}, \ldots, H_j^m g_{k+1} = H g_{k+1}.$$
Then
\[
d_{k+1}^{\text{CG}} (H_{j+m}^m) = - g_{k+1} + \frac{g_{k+1}^T H_{j+m}^m y_k}{d_k^T y_k} d_k
\]

\[
= - g_{k+1} + \frac{g_{k+1}^T H y_k}{d_k^T y_k} d_k = d_{k+1}^{\text{CG}} (H)
\]

So the induction holds and shows that each step of the VSCG method is the same as a CG step with metric \( H \). The result follows from the quadratic termination property of the CG method. ///

Observe that the above result is independent of what member of Broyden's class one chooses. This is in sharp contrast with the Interleaved Method where BFGS has to be used.

Some specific updating strategies allowed by Theorem 5.36 are the following: (a) updating \( H_i \) at every iteration, (b) resetting \( H_i \) to \( H \) after a certain number of steps, and (c) resetting \( H_i \) any previous matrix \( H_j \).
CHAPTER VI
LOCAL CONVERGENCE

In this section we study the convergence of the combined QN-CG methods. Powell [31,32] has established global convergence of the DFP for a uniformly positive definite matrix. After him several convergence analysis for QN methods have been given by a number of authors. We shall base the analysis of this section on the work of Stoer [26]. Even though we won't prove a global result as Powell's but only a local one, we will readily establish Q-linear convergence. The general setting of Stoer's presentation enables us to study several algorithms in the same framework. In the practical implementation of the combined methods we will employ less than $O(n^2)$ storage, therefore the bounded deterioration theorem (see Dennis and More’ [33]) cannot be used, since the matrices $H_k$ cannot be expected to be near to $H(x^*)$, the Hessian at the solution.

In Ortega and Rheinholdt [28] it is shown that if $f$ is continuously differentiable, the level set is compact and $p_k$ is such that $p_k^T g_k < 0$ then the iteration

$$x_{k+1} = x_k + \alpha_k p_k$$

satisfies $\lim_{k \to \infty} \frac{g_k^T p_k}{\|p_k\|} = 0$, for various choices of the steplength. These include the Curry-Altman, Curry and Goldstein principles. For a QN method $p_k = -H_k g_k$. Assume that the condition number of the $H_k$ is bounded by $\gamma$ then

$$0 = \lim_{k \to \infty} - \frac{g_k^T p_k}{\|p_k\|} = \lim_{k \to \infty} \frac{g_k^T H_k g_k}{\|H_k g_k\|}$$

$$\geq \lim_{k \to \infty} \frac{\|g_k\|}{\|H_k\| \cdot \|H_k^{-1}\|} \geq \lim_{k \to \infty} \frac{\|g_k\|}{\gamma}$$
and therefore \( g_k^T p_k \to 0 \). Essentially, convergence can be established for a QN method if the condition number of the matrices can be bounded. For the CG method with metric \( H_k \), the situation is somewhat more complicated. Here \( p_k = - H_k g_k + \beta_k p_{k-1} \), so that

\[
- g_k^T p_k / \| p_k \| = \left( g_k^T H_k g_k - \beta_k g_k^T p_{k-1} \right) / \| - H_k g_k + \beta_k p_{k-1} \| .
\]

However, the approach will be the same. We will find a step length for which the condition number of the \( H_k \) remains bounded. With this we will be able to establish not only local convergence but also a Q-linear rate of convergence.

Let \( f: \mathbb{R}^n \to \mathbb{R} \) be the function to be minimized, \( g \) its gradient and \( h \) its Hessian. Let \( x^* \) be a local minimizer of \( f \).

**Assumption 6.1** Let \( U(x^*) \) be a neighborhood of \( x^* \). We assume that

1. \( f \in C^2 \) on \( U(x^*) \)
2. \( h(x^*) \) is positive definite
3. \( h \) is Lipschitz continuous at \( x^* \).

For the convergence analysis we may assume, without loss of generality, that \( x^* = 0 \), \( f(x^*) = 0 \) and \( h(x^*) = I \).

We will denote Broyden's update by \( U_B \); \( H = U_B(s,y,H,\theta) \) where \( U_B \) is given by (1.3). Assumptions (6.1) imply that there is a neighborhood of \( x^* = 0 \) and a Lipschitz constant \( \Lambda \) such that
\[ f(x) = \frac{1}{2} x^T x + R(x) , \quad |R(x)| \leq (\Lambda/6) \| x \|^3 \]
\[ g(x) = x + r(x) \quad \| r(x) \| \leq (\Lambda/2) \| x \|^2 \] (6.2)
\[ h(x) = I + s(x) \quad \| s(x) \| \leq \Lambda \| x \| \]

holds for all \( x \) in that neighborhood. Given \( 0 \leq \mu_1 \leq \mu < 1 \) the Curry-Altmann steplength is given by
\[ \alpha_1 = \left\{ \min t: t \geq 0, \ g(x_1 - tp_1) = \mu_1 g(x_1) \right\} . \] (6.3)

Let \( \rho > 0 \), then for all \( \| x \| < \rho \) we define \( \overline{L}(\rho, x) = \{ z: \| z \| < \rho , \ f(z) \leq f(x) \} \). Let \( L(\rho, x) \) be the path-connected component of \( \overline{L} \) that contains \( x \) . Given \( x, p, 0 \leq \theta , 0 \leq \mu < 1 \) and \( H(\text{pos. def.}) \) we define
\[ x^* = x + \alpha_p , \quad H^* = U_B(x^* - x, g^* - g, H, \theta) \] (6.4)

where \( \alpha \) is the Curry-Altmann steplength. The following Proposition is proved in Stoer [26], Propositions 3.1, 3.2, 3.3).

**Proposition 6.5** Under the Assumptions 6.1 there exists \( \rho > 0 \) such that for all \( \| x \| < \rho , \ x \neq 0 \), the level set \( L(\rho, x) \) is compact and convex. If in addition \( p \) is a descent direction i.e., \( p^T g(x) < 0 \), then

1. \( \alpha > 0 \quad \alpha_p > 0 \), where \( q = g(x^*) - g(x) \)
2. \( f(x^*) < f(x) \) and \( x^* \in L(\rho, x) \)
3. \( H^* \) is positive definite
4. \( f(x) - f(x^*) = \frac{1}{2} (1 - \mu)^2 \left( g^T p \right)^2 \frac{p^T p}{p^T p} \left( 1 + 0 (\| x \|) \right) \)

- 48 -
We shall study the convergence of the following algorithm.

**Algorithm A:** \( \mu, x_0, 0 \leq \theta \) and \( H_0 \) (pos. def.) are given.

\[
x_{i+1} = x_i + \alpha_i p_i
\]

(6.5)

where

\[
\alpha_i = \left\{ \min \left\{ t : t \geq 0, \ g(x_i + tp_i)^T p_i = \mu g(x_i)^T p_i \right\} \right. 
\]

(6.6)

\( \mu_i \) is any number satisfying

\[
\begin{cases} 
0 \leq \mu_i \leq \mu < 1 & \text{if } i \equiv 0 \mod(n) \\
0 \leq \mu_i \leq \min \left\{ \mu, C\|g_i\| \right\} & \text{if } i \not\equiv 0 \mod(n)
\end{cases}
\]

(6.7)

Here \( C \) is a positive constant.

\[
H_i = U_B(x_i - x_{i-1}, g_i - g_{i-1}, H_{i-1}, \theta)
\]

(6.8)

\[
p_i = -H_i g_i + \beta_i p_{i-1}
\]

(6.9)

with

\[
\beta_i = \left\{ \begin{array}{cl}
0 & \text{if } i \equiv 0 \mod(n) \\
\frac{g_i^T H_i g_i}{g_{i-1}^T H_{i-1} g_{i-1}} & \text{if } i \not\equiv 0 \mod(n)
\end{array} \right.
\]

(6.10)

Algorithm A is a CG method restarted every \( n \) steps that uses a metric updated by Broyden's class. The Fletcher-Reeves parameter was chosen but
the results in this section can be obtained similarly for the Hestenes-Stiefel or Polak-Ribiere parameters. As will be observed later, Algorithm A can be changed slightly to give Quasi-Newton, Interleaved or VSCG methods. The difference in the precision of the search expressed in (6.7) is due to the fact that a pure CG step requires a more accurate line search to ensure descent than a pure gradient step.

For any \( i \)

\[
g_i^T p_i = -g_i^T H_i g_i + \beta_i g_i^T p_{i-1}
\]

\[
= -g_i^T H_i g_i + \beta_i \mu_i g_i^T p_{i-1}.
\]

(6.11)

From this equation we can obtain necessary and sufficient conditions on \( \mu_i \) (as a function of \( \beta_i \)) that insure descent. In particular if \( H_i = I \) and \( \beta_i \) is the Hestenes-Stiefel parameter we have that

\[
\mu_i < g_i+1^T g_i+1 + g_i g_i^T \]

which is the condition used by Lenard [30] to show convergence. Although condition (6.7)-b is too restrictive we will use it because it applies to all our algorithms.

The following Theorem is the key result in the convergence analysis.

First we need some Definitions:

\[
k(H) = \max \left\{ 1, \| H^{-1} \| \right\} \quad ; \quad k(H^{-1}) = \max \left\{ 1, \| H^{-1} \| \right\}
\]

\[
a(H) = K(H) \cdot K(H^{-1})
\]

\[
\text{cond}(H) = \| H^{-1} \| \cdot \| H^{-1} \|
\]

(6.12)
where \( \| \cdot \| \) denotes \( \| \cdot \|_2 \).

**Theorem 6.13** (Stoer [26]) Let \( B \) be pos. def. and \( 0 \leq \theta \leq 1 \).

There exists a neighborhood \( U(0) \) such that the following holds. If 
\( x', x \in U(0), \ x', x \neq 0, x \neq x, f(x') < f(x) \) and 
\( p^T q > 0 \) (where \( p = x' - x, q = g' - g \)) then the matrix \( H' = U_B(x' - x, g' - g, H, \theta) \) satisfies

\[
\begin{align*}
(1) & \quad k(H') \leq k(H) \left[ 1 + O(\| x \|) \right] \\
(2) & \quad k(H'^{-1}) \leq k(H'^{-1}) \left[ 1 + O(\| x \|) \right] \\
(3) & \quad \text{cond } (H') \leq a(H') \leq a(H) \left[ 1 + O(\| x \|) \right]
\end{align*}
\]

As usual \( O(\| x \|) \) denotes a "quantity of order \( \| x \| \)." Note that this result is stated only for \( 0 \leq \theta \leq 1 \) which defines "Broyden's restricted class."

**Proposition 6.14** For \( m > 2 \), under the assumptions 6.1, there exists a neighborhood \( U(0) \) such that Algorithm A is a descent method and

\[ b_k \leftarrow g_k^T p_{k-1} \leq \frac{2}{m} g_k^T H_k g_k \]

holds for all \( k \).

**Proof:** \( g_0^T p_o = -g_0^T H_0 g_o < 0 \) then Proposition 6.5 guarantees that \( H_1 \) is pos. def. and that \( f(x_1) < f(x_0) \), if \( x_0 \) is sufficiently close to the solution. Assumptions 6.1 imply that in a neighborhood of the solution \( f(x) < f(x) \Rightarrow \| g(x) \| \leq \| g(x) \| \) and \( \| x \| \leq \| x \| \).

**Induction Hypothesis:** Assume that for \( i = 1,2,\ldots,(k-1) < n-1 \)

\[
\begin{align*}
(1) & \quad g_1^T p_1 < 0 \\
(2) & \quad g_1^T p_1 > 0 \quad (\Rightarrow H_{i+1} \text{ is pos. def.}) \\
(3) & \quad \left| g_1^T p_1 \right| \leq \gamma_i g_1^T H_i g_1 \\
(4) & \quad \| g_{i+1} \| \leq \| g_i \|
\end{align*}
\]
Equations (6.15) hold for \( i = 0 \) with \( \gamma_0 = 1 \). Then

\[
\begin{align*}
\mathbf{s}_k^T \mathbf{p}_k & = -\mathbf{g}_k^T \mathbf{H}_k \mathbf{s}_k + \beta_k \mathbf{g}_k^T \mathbf{p}_{k-1} \\
& = -\mathbf{g}_k^T \mathbf{H}_k \mathbf{s}_k + \frac{\mathbf{g}_k^T \mathbf{H}_k \mathbf{g}_k}{\varepsilon_{k-1}^T \mathbf{H}_k \varepsilon_{k-1}} \varepsilon_{k-1}^T \mathbf{p}_{k-1} \\
& \leq -\varepsilon_k^T \mathbf{H}_k \varepsilon_k + \frac{\mathbf{g}_k^T \mathbf{H}_k \mathbf{g}_k}{\varepsilon_{k-1}^T \mathbf{H}_k \varepsilon_{k-1}} C \| \varepsilon_{k-1} \| \gamma_{k-1} \varepsilon_{k-1}^T \mathbf{H}_k \varepsilon_{k-1} \\
& \leq -\varepsilon_k^T \mathbf{H}_k \varepsilon_k \left(1 - C \gamma_{k-1} \| \mathbf{H}_k^{-1} \| \varepsilon_{k-1}^T \mathbf{H}_k^{-1} \| \varepsilon_{k-1} \| \right) \quad (6.16)
\end{align*}
\]

Using Theorem 6.13

\[
\begin{align*}
k(\mathbf{H}_k) & \leq k(\mathbf{H}_{k-1}) \left[1 + 0 (\| x_{k-1} \|)\right] \leq k(\mathbf{H}_{k-2}) \left[1 + 0 (\| x_{k-2} \|)\right] \leq \cdots \\
& \leq k(\mathbf{H}_0) \left[1 + 0 (\| x_0 \|)\right]
\end{align*}
\]

\[
k(\mathbf{H}_k^{-1}) \leq k(\mathbf{H}_0^{-1}) \left[1 + 0 (\| x_0 \|)\right]
\]

Therefore (6.16) gives

\[
\begin{align*}
\mathbf{s}_k^T \mathbf{p}_k & \leq -\mathbf{g}_k^T \mathbf{H}_k \mathbf{s}_k \left(1 - C \gamma_{k-1} k(\mathbf{H}_0^{-1}) k(\mathbf{H}_0) \left[1 + 0 (\| x \|)\right] \| \varepsilon_{k-1} \| \right)
\end{align*}
\]

Choose \( x_0 \) small enough so that \( C k(\mathbf{H}_0^{-1}) k(\mathbf{H}_0) \| g_0 \| \left[1 + 0 (\| x_0 \|)\right] \leq \frac{1}{m} \).
Then

\[ g_k^T p_k \leq -g_k^T H_k g_k \left( 1 - \gamma_{k-1}/m \right). \quad (6.17) \]

Define \( \gamma_k = (1 - \gamma_{k-1}/m) \) with \( \gamma_0 = 1 \). Then \( \gamma_2 = \left( 1 - \frac{1}{m} \right), \ldots, \gamma_k = \left( 1 - \frac{1}{m} + \frac{1}{2^2} + \ldots - \frac{1}{m^{k-2}} + \frac{1}{m^{k-1}} \right) < 2 \). Therefore

\[ g_k^T p_k \leq -g_k^T H_k g_k \left( 1 - (2/m) \right) < 0 \quad (6.18) \]

\[ \beta_k \left| g_k^T p_{k-1} \right| \leq (2/m) g_k^T H_k g_k \]

Proposition 6.5 and (6.18) imply that the Eqs. (6.15) hold for

\[ i = 0, 1, \ldots, n-1. \quad \text{Consider the next cycle} \quad i = n, n+1, \ldots 2n-1. \]

\[ g_n^T p_n = -g_n^T H_n g_n < 0. \]

Assume that (6.15) holds for \( i = n, n+1, \ldots r-1 < 2n-1 \), then using (6.16)

\[ g_r^T p_r \leq -g_r^T H_r g_k \left( 1 - C \gamma_{r-1} \|H_r^{-1}\| \|H_r\| \|g_{r-1}\| \right) \]

Again we have

\[ K(H_r) \leq K(H_n) \left[ 1 + 0 (\|x_n\|) \right] \leq K(H_0) \left[ 1 + 0 (\|x_0\|) \right] \]

\[ K(H_r^{-1}) \leq K(H_0^{-1}) \left[ 1 + 0 (\|x_0\|) \right] \]

\[ g_r^T p_r \leq -g_r^T H_r g_r \left( 1 - C K(H_r^{-1}) K(H_r) \gamma_{r-1} \|g_{r-1}\| \right) \]

\[ \leq -g_r^T H_r g_r \left( 1 - \gamma_{r-1}/m \right), \quad (6.19) \]
because of descent: \( \| s_{r-1} \| \leq \| s_0 \| , \| x_n \| \leq \| x_0 \| \) . Therefore the \( \gamma_i \) are independent of the cycle number and are bounded by 2 . We conclude that

\[
g_1^T p_1 = -g_1^T H_1 g_1 < 0 \quad \text{for all } i \equiv 0 \mod(n) \]

\[
g_1^T p_1 \leq -g_1^T H_1 g_1 (1 - 2/m) < 0 \quad \text{for all } i \equiv 0 \mod(n) .
\]

This proves that Algorithm A is a descent method, that \( H_1 \) is positive definite and that \( \beta_1 |g_1^T p_1| \leq (2/m) g_1^T H_1 g_1 \) holds for all \( i \).

**Proposition 6.20** Under the assumptions of Proposition 6.14 there exist constants \( B,D > 0 \) such that

\[
|g_k^T p_k| \geq D \| g_k \|^2 / \| H_k^{-1} \| \quad \text{and}
\]

\[
\text{and} \quad \| p_k \|^2 \leq B \| g_k \|^2 \| H_k \|^2 \quad \text{hold for all } k .
\]

**Proof:** For \( i \equiv 0 \mod(n) \)

\[
|g_1^T p_1| = g_1^T H_1 g_1 \geq \| g_1 \|^2 \| H_1^{-1} \| \quad (6.21)
\]

For \( k \not\equiv 0 \mod(n) \), we have from Proposition 6.14

\[
|g_k^T p_k| \geq g_k^T H_k g_k - \beta_k |g_k^T p_k| \]

\[
\geq g_k^T H_k g_k - (2/m) g_k^T H_k g_k
\]

\[
\geq g_k^T H_k g_k \left( 1 - (2/m) \| g_k \|^2 / \| H_k^{-1} \| \right) . \quad (6.22)
\]

So the first part of the Proposition holds with \( D = 1 - (2/m) \). Now

\[
\| p_1 \|^2 = g_1^T H_1^2 g_1 \quad \text{for } i \equiv 0 \mod(n) . \quad (6.23)
\]

For \( k \not\equiv 0 \mod(n) \)
\[ \| p_k \|^2 \leq g_k^T H_k^2 g_k + 2 \beta_k |g_k^T H_k p_{k-1}| + \beta_k^2 \| p_{k-1} \|^2 \] (6.24)

\[ \beta_k |g_k^T H_k p_{k-1}| = \frac{g_k^T H_k g_k}{g_k^T H_k g_{k-1}} |g_k^T H_k p_{k-1}| \]

\[ \leq (g_k^T H_k g_k) \| g_{k-1} \| \cdot \| H_k^{-1} \| \cdot \| H_k \| \cdot \| g_k^T p_{k-1} \| \]

\[ = (g_k^T H_k g_k) \| g_{k-1} \| \cdot \| H_k^{-1} \| \cdot \| H_k \| \cdot \| g_{k-1}^T p_{k-1} \| \]

\[ \leq (g_k^T H_k g_k) \| g_{k-1} \|^2 \cdot \| H_k^{-1} \| \cdot \| H_k \| \cdot C \cdot g_{k-1}^T \cdot \gamma_{k-1} g_{k-1} \cdot H_k \]

(from 6.15 - 3)

\[ \leq (g_k^T H_k g_k) \| H_k^{-1} \| \cdot \| H_k \| \cdot \| H_k^{-1} \| \cdot 2C \cdot g_{k-1} \| / m \]

\[ \leq \| g_k \|^2 \cdot \| H_k \|^2 \left( K(H_k^{-1}) \cdot K(H_k^{-1}) \cdot 2C \cdot g_{k-1} \| / m \right) \] (6.25)

**Induction:** assume that \( \| p_i \|^2 \leq \delta_i \| g_i \|^2 \| H_i \|^2 \) holds for \( i = 1, 2, \ldots, k-1 < n-1 \).

From (6.23) we may choose \( \delta_i = 1 \) for all \( i \equiv 0 \) mod(n). Then

\[ \| p_{k-1} \|^2 \beta_k^2 \leq \frac{(g_k^T H_k g_k)^2}{(g_k^T H_k g_{k-1})^2} \cdot \delta_{k-1} \| g_{k-1} \|^2 \| H_{k-1} \|^2 \]

\[ \leq \| g_k \|^4 \| g_k \|^2 \cdot \| H_k \|^2 \| H_k^{-1} \|^2 \left( \delta_{k-1} \| g_{k-1} \|^2 \| H_{k-1} \|^2 \right) \]

\[ \leq \| g_k \|^2 \| H_k \|^2 \delta_{k-1} \| H_{k-1} \|^2 \cdot \| H_k^{-1} \| \] (6.26)
As (6.25) holds for every iteration \( k \neq 0 \mod(n) \) we have from (6.24) – (6.26)

\[
\| p_k \| \leq \| g_k \| H_k \| \left(1 + \frac{2}{m} \right) C K(H^{-1}_k) K(H_{k-1}) \| g_{k-1} \| + \delta_{k-1} \cdot H_k^{-1} \| H_k \| \frac{1}{2}.
\]

By Theorem (6.13) and descent

\[
\| p_k \| \leq \| g_k \| \| H_k \| \left(1 + \frac{2}{m} \right) C K(H^{-1}_k) K(H) \left[1 + 0 \left(\| x_o \|\right)\right] \| g_{k-1} \|
\]

\[
+ \delta_{k-1} \| H_k^{-1} \| \| H_k \| \frac{1}{2}
\]

\[
\leq \| g_k \| \| H_k \| \left(1 + \frac{2}{m} \right) \left[1 + 0 \left(\| x_o \|\right)\right] \delta_{k-1} \| H_k^{-1} \| \| H_k \| \frac{1}{2}
\]

\[
\leq \| g_k \| \| H_k \| \left(1 + \frac{2}{m} \right) \left[1 + \frac{2}{m} \right] \| H_k^{-1} \| \| H_k \| \frac{1}{2} + \delta_{k-1} \| H_k^{-1} \| \| H_k \| \| x_o \| \| t \|
\]

where

\[
t \triangleq 1 + 0 \left(\| x_o \|\right)
\]

\[
\leq \| g_k \| \| H_k \| \left(2 + \delta_{k-1} \text{cond}(H) \right) \frac{1}{2}
\]

(6.27)

Let \( \rho \geq \text{cond}(H) \cdot t \). We define \( \delta_k \triangleq \left(2 + \delta_{k-1} \rho \right) \). Then we have that

\[
\| p_k \| \leq \delta_k \| g_k \| \| H_k \| \frac{1}{2}.
\]

From the induction hypothesis \( \delta_0 = 0 \), hence

\[
\delta_1 = \left(2 + \rho \right), \quad \delta_2 = 2 + 2\rho + \rho^2, \ldots,
\]

\[
\delta_k = 2 + 2\rho + \rho^2 + 2\rho^{k-1} + \rho^k.
\]

This constant \( \delta \rightarrow \infty \). We could not proceed with the proof if the Algorithm is not restarted. However,
\[ \delta_k \leq B + 2 + 2\rho + \ldots 2\rho^{n-1} + \rho^n. \]

Therefore the induction holds
\[ \|p_i\| \leq B\|g_i\|^2 \|H_i\|^2 \quad i = 2, 3, \ldots, n-1. \quad (6.28) \]

Now we may proceed as in the proof of Proposition 6.14. We consider the next cycle \( i = n, n+1, \ldots, 2(n-1) \). Note that (6.24) and (6.25) hold and as \( \delta_n = 1 \) we may proceed by induction to establish (6.28). The constants \( \delta_k \) will be the same as for the previous cycle. This proves the second part. ///

We can now establish local coverage for Algorithm A.

**Theorem 6.29** Under the assumptions 6.1, there exists a \( \rho > 0 \) such that Algorithm A starting from \( \|x_0\| < \rho \), \( H_0 \) pos. def. and \( 0 \leq \theta \leq 1 \) is Q-linearly convergent.

**Proof:** If \( x_k = 0 \) for some \( k \) we are done; so we assume that \( \{x_i\} \) is an infinite sequence of nonzero vectors. From proposition 6.5 and 6.20
\[
f(x_i) - f(x_{i+1}) = \frac{1}{2} (1 - \mu)^2 \frac{(g_i^T p_i)}{p_i^T p_i} (1 + O(\|x_i\|))
\]

\[
f(x_i) - f(x_{i+1}) \geq \frac{\rho(1 - \mu^2)B^2\|g_i\|^4 (1 + O(\|x_i\|))}{B\|g_i\|^2 \|H_i\|^2 \|H_i^{-1}\|^2}. \quad (6.30)
\]

We can write \( T.f(x_i) \leq \|g_i\|^2 \) for some constant \( T > 0 \). Therefore there is a constant \( 0 < s < 1 \) such that

- 57 -
\[ f(x_i) - f(x_{i+1}) \leq sf(x_i)/\text{cond}(H_i)^2 \]

hence

\[ f(x_{i+1}) \leq f(x_i) \left(1 - \frac{s}{\text{cond}(H_i)^2}\right) \]

\[ \leq f(x_i) \left(1 - \frac{s}{a(H_i)^2}\right) \quad (6.31) \]

and from Theorem 6.13 we have, since \( \|x_i\|^2 = 0(f(x_i)) \),

\[ a(H_{i+1})^2 \leq a(H_i)^2\left[1 + d\sqrt{f(x_i)}\right] \]

where \( d \) is a constant, \( d > 0 \). Define \( a_i \Delta \equiv a(H_i)^2 \), then

\[ a_{i+1} \leq a_i \left[1 + d\sqrt{f_i}\right] \quad (6.32) \]

Now we finish the proof as in Stoer [26]. Let \( \epsilon \) be the basis of \( \mathbb{R} \), define \( \gamma^2 \Delta \equiv 1 - s/(ea_0) \). Clearly \( 0 \leq \gamma \leq 1 \). Now

\[ a_i \leq a_0\left(1 + d\sqrt{f_0}\right) \leq ea_0 \quad \text{if} \quad x_0 \quad \text{is sufficiently small.} \]

Proceeding by induction assume that \( a_i \leq ea_0 \)

\[ a_{i+1} \leq a_i \left(1 + d\sqrt{f_i}\right) \leq a_i \exp \left(d\sqrt{f_i}\right) \leq \ldots a_0 \exp \left(d\sum_{j=0}^{1}\sqrt{f_i}\right) \]

\[ \leq a_0 \exp \left(d\sqrt{f_0}(1 + \gamma + \gamma^2 + \ldots + \gamma^2)\right) \leq a_0 \exp \left(d\sqrt{n_0}/1 - \gamma\right) \quad (6.33) \]

Now \( (1 - \gamma^2) = (1 - \gamma)(1 + \gamma) = \frac{s}{ea_0} \implies 1/1 - \gamma \leq 2ea_0/s \quad . \)
Hence

\[ a_{i+1} \leq a_0 \exp \left( d \sqrt{F_0} \cdot 2e\alpha_0 / s \right). \]

If \( x_0 \) is sufficiently small \( d \sqrt{F_0} 2e\alpha_0 / s \leq 1 \), therefore \( a_{i+1} \leq e\alpha_0 \) and the induction holds. From (6.31)

\[ f(x_{i+1}) \leq f(x_i)(1 - s/e\alpha_0) \quad \text{for all} \quad i. \]

Hence \( \{f_i\} \) converges \( Q \) linearly and clearly, so does \( \{x_i\} \). ///

Algorithm A restarted every iteration is Broyden's restricted class of quasi-Newton methods.

**Corollary 6.34** Under the assumption 6.1, for any pos. def. matrix \( H_0 \), \( 0 \leq \mu_1 \leq \mu < 1 \) and \( 0 \leq \theta \leq 1 \), Broyden's class is locally and \( Q \)-linearly convergent.

The Interleaved and VSCG methods will set \( H = I \) regularly, whenever it is considered that the information stored in \( H \) is no longer useful (see Chapter 9). Clearly the convergence result of thm. 6.29 is still valid in this case. The Interleaved method can be described using Algorithm A but varying the frequency of restarts. Instead of \( i \equiv 0 \mod(n) \) in (6.7) and (6.10) we shall write \( i \in T \), where \( T \) specifies the set of indices where \( QN \) steps are performed. The VSCG is also a variation of Algorithm A that differs only in the restarting frequency and the resetting \( H = I \). The details on the implementation of these two methods are given in Chapter 9. It should be clear that Theorem 6.29 holds for both methods.
Corollary 6.35 The Interleaved and VSCG methods using Broyden's update with $0 \leq \theta \leq 1$ are locally and Q-linearly convergent, if the stepsize is specified as follows:

For Interleaved method $\mu_1$ satisfies $0 \leq \mu_1 \leq \mu < 1$ if the step is a QN step, and $0 \leq \mu_1 \leq \min \left\{ \mu, C \|g_1\| \right\}$ if the step is a CG step. (Here $\mu$ and $C$ are arbitrary positive constants).

For the VSCG $\mu_1$ satisfies $0 \leq \mu_1 \leq \mu < 1$ for all restarting iterations and $0 \leq \mu \leq \min \left\{ \mu, C \|g_1\| \right\}$ for all other iterations.
CHAPTER VII

RATE OF CONVERGENCE

The conjugate gradient method and Broyden's restricted class of quasi-Newton methods are \( n \)-step quadratically convergent. Since the Interleaved and the VSCG methods combine CG and QN steps in such a way that quadratic termination is preserved, we would expect them to be also \( n \)-step quadratically convergent. In this Chapter we show that this is indeed the case.

There are many implementations of the Interleaved and VSCG methods that have quadratic termination. They are obtained by varying the frequency of updating and restarting. Instead of studying a general algorithm that would cover all the cases we will concentrate on one particular implementation. The analysis for the others would be essentially the same.

Consider the Interleaved method; a cycle of \( n \) iterations will consist of \( r \) QN updates followed by \( l = n - r \) CG steps. We shall use the theory developed by Stoer [27]. We specify an algorithm by the 5-type

\[
M = \{A; \{\mu_i\}, \{\phi_i\}, \{\psi_i\}, \{B_i\}\}
\]

where \( A \) is a set of starting parameters, \( \mu_i \) determines the accuracy of the line searches and \( \phi_i, \psi_i, B_i \) are functions to be defined later. \( M \) produces a sequence of iterates in the following way:

Given \( x_0 \in \mathbb{R}^n \) and \( a_0 \in A \) we let for \( i = 0,1, \ldots \)

\[
(1) \quad s_i = \phi_i(g_i, a_i, b_i)
\]

\[
(2) \quad x_{i+1} = x_i + \lambda_i s_i, \quad p_i = x_{i+1} - x_i
\]
where \( \lambda_1 \) is chosen so that \( g_{i+1}^T s_i = \mu_1 g_1^T s_i \) and \( f(x_{i+1}) \leq f(x_i) \)

\[
(3) \quad a_{i+1} = \psi_1(p_1, g_i, s_{i+1}, a_i)
\]

\[
(4) \quad b_{i+1} = B_1(g_i, a_i, b_i)
\]

We now specify the Interleaved Method completely by letting \( A \) to be the set of \( nxn \) positive definite and symmetric matrices, \( a_1 \equiv H_1, \mu_1 \) being any number with \( |\mu_1| \leq \mu < 1 \), and defining

\[
\tilde{\phi}_1(g_i, H_1 b_i) = \left\{ \begin{array}{ll}
\tilde{\phi}(H_1, g_i) & i=0,1,\ldots,r-1 \\
\frac{b_i}{\| b_i \|} & \text{if } b_i \neq 0 \\
\frac{g_i}{\| g_i \|} & \text{if } b_i = 0 
\end{array} \right.
\]  

(7.1)

where \( \tilde{\phi}(x) = x/\| x \| \).

\[
\psi_1(p_1, g_{i+1}, H_1) = \left\{ \begin{array}{ll}
\tilde{\psi}(p_1, g_{i+1}, -g_i, H_1) & i=0,1,\ldots,r-1 \\
H_1 & i=r,\ldots,n-1 
\end{array} \right.
\]

(7.2)

where

\[
\tilde{\psi}(p_1, q_i, H_1) = H_1 + \frac{1}{p_1^T q_i} \left[ 1 + \frac{q_i^T H_1 q_i}{p_1^T q_i} \right] p_1 p_1^T
\]

\[
- \frac{1}{p_1^T q_i} \left( p_1^T H_1 q_i + q_i^T p_1^T H_1 p_1 \right)
\]

(7.3)

is the BFGS update formula and as in the previous section \( q_i = q_{i+1} - g_i \).
\[ B_i (H_i, g_i, b_i) = \begin{cases} H_i g_i & i = 0, 1, \ldots, r-1 \\ H_i g_i + \frac{g_{i+1}^2}{\| g_i \|^2 b_i} & i = r, \ldots, n-1 \end{cases} \] 

(7.4)

**Assumption 7.5** There is a neighborhood \( U(x^*) \) where \( f \) satisfies the following:

1. \( f \in C^2 \) on \( U(x^*) \)
2. There exists \( m, M > 0 \) such that \( m\| x \|^2 \leq x^T h(x^*) x \leq M\| x \|^2 \) for all \( x \in \mathbb{R}^n \)
3. \( h \) is Lipschitz continuous at \( x^* \).

Again, without loss of generality we assume that \( x^* = 0 \) and \( f(x^*) = 0 \).

**Theorem 7.6** Let \( f \) satisfy assumptions 7.5. Consider the Interleaved Method defined by (7.1) - (7.4) that uses asymptotically perfect line searches, \( |u_i| \leq \alpha \| g_i \| \). Then for any \( H \in A \) there exist constants \( \varepsilon, C > 0 \) such that for all \( \| x_0 \| \leq \varepsilon \) the algorithm generates a sequence \( \| x_i \| \) such that for all \( i \equiv 0 \mod(n) \)

\[ \| x_{i+n} \| \leq C \| x_i \|^2. \]

**Proof:** We will only outline the proof since it is rather long and it follows along the same lines as Stoer's [27] and Cohen's [12].
Part 1: As shown in Chapter 6 the choice of steplength insures that
\( p_i^T q_i > 0 \) and therefore the matrices \( H_i \) are positive definite.
Assuming that \( g_i \neq 0 \) for all \( i \) the algorithm will be well defined.
In Chapter V it was shown that it has the quadratic termination property
if exact line searches are used. It is also homogeneous in the following
sense
\[
\phi_i(\alpha g_i, H_i, b_i) = \phi_i(g_i, H_i, b_i) \quad \text{for } i = 0, 1, \ldots, r-1
\]
\[
\phi_i(\alpha g_i, H_i, \alpha b_i) = \phi_i(g_i, H_i, b_i) \quad \text{for } i = r, \ldots, n-1
\]
\[
\psi_i(\alpha p_i, \alpha g_i, \alpha g_{i+1}, H_i) = \psi_i(p_i, g_i, g_{i+1}, H_i) \quad \text{for } i = 0, 1, \ldots, n-1
\]
\[
B_i(H_i, \alpha g_i, \alpha b_i) = \alpha B_i(H_i, g_i, b_i) \quad \text{for } i = r, \ldots, n-1
\]

Note that the functions \( \psi_i \) are essentially only used during the first
\( r \) iterations and the functions \( B_i \) only during the last \( n-r \).

Part 2: Bounds on the Fréchet derivatives.
Under the assumptions 7.5 there is a neighborhood \( \tilde{U}(x^*) \) such that there
exists \( m, M > 0 \) satisfying
\[
1/2 m \| y \|^2 \leq y^T h(x) y \leq 2M \| y \|^2 \quad (7.8)
\]
for all \( y \in \mathbb{R}^n, x \in \tilde{U}(x^*) \). Let \( C \equiv h(x^*) \) and define \( \tilde{f}(x) = \frac{1}{2} x^T C x \).
Let \( \{ \tilde{x}_i \}, \{ \tilde{H}_i \}, \{ \tilde{b}_i \} \) be the sequences generated by the algorithm
when applied to \( \tilde{f} \) using exact line searches, starting from \( x_0 \) and \( H_0 \).

For \( i = 0, 1, \ldots, r-1 \) one can show that there exist constants
\( \gamma(H_0), \Gamma(H_0) \) such that
$$\| D\phi_i (g_i^r, H_i^r, b_i^r) \| \leq \Gamma \quad \text{for all } g_i^r, H_i^r, b_i^r$$

$$\| D\psi_i (p_i^r, s_i^r, g_{i+1}^r, H_i^r) \| \leq \Gamma \quad \text{for all } g_i^r, H_i^r, p_i^r, g_{i+1}^r$$

$$i = 0, 1, \ldots, r-1$$

for all $g_i^r, H_i^r, p_i^r, g_{i+1}^r$ satisfying

$$\left\| \frac{g_i^r - \tilde{g}_i^r}{\| \tilde{g}_i^r \|} \right\| + \left\| H_i^r - \tilde{H}_i ^r \right\| + \left\| p_i^r - \tilde{p}_i ^r \right\| + \left\| g_{i+1}^r - \tilde{g}_{i+1}^r \right\| \leq \gamma .$$

(note that $\phi_i$ is independent of $b$ for $i = 0, \ldots, k-1$). The matrix $H_i$ is not updated during the last $n-r$ iterations, therefore

$$\| H_i - \tilde{H}_i \| \leq \gamma \quad \text{for } i = r, \ldots, n-1 .$$

Similarly, for $i = 0, 1, \ldots, r-1$ there exist $\gamma^r(H_o), \Gamma^r(H_o)$ such that

$$\| D\phi_i (g_i^r, H_i^r, b_i^r) \| \leq \Gamma^r$$

$$\| DB_i (H_i^r, g_i^r, b_i^r) \| \leq \Gamma^r \quad i = r, \ldots, n-1$$

for all $g_i^r, H_i^r, b_i^r$ satisfying

$$\left\| \frac{g_i^r - \tilde{g}_i^r}{\| \tilde{g}_i^r \|} \right\| + \left\| H_i^r - \tilde{H}_i ^r \right\| + \left\| b_i^r - \tilde{b}_i ^r \right\| + \left\| g_{i+1}^r - \tilde{g}_{i+1}^r \right\| \leq \gamma^r .$$

We can choose $\bar{\gamma} = \min \left\{ \gamma, \gamma^r \right\}$, $\bar{\Gamma} = \max \left\{ \Gamma, \Gamma^r \right\}$ and obtain bounds for the derivatives for the whole cycle.

**Part 3:** We compare the differences $x_j - \tilde{x}_j$ for $0 \leq j \leq n-1$. One shows that for $x_0$ sufficiently small there exists constants
\( \xi_j(m, M, \Lambda, \alpha, \bar{\Gamma}, \bar{\gamma}) \) and \( \lambda_j(m, M, \Lambda, \alpha, \bar{\Gamma}, \bar{\gamma}) \) (\( \Lambda \) is the Lipschitz constant) such that

\[
\begin{align*}
\| x_j - \tilde{x}_j \| & \leq \xi_j \| x_0 \|^2 \\
\| H_j - \tilde{H}_j \| & \leq \lambda_j \| x_0 \|^2 \| x_k \|^{-1} \quad \text{where} \quad k = \max\{|0, j-1|\}
\end{align*}
\]

For the last \( n-r \) iterations \( (H_j - \tilde{H}_j) \) remains constant and we need only compute the differences \( b_j - \tilde{b}_j \). Indeed

\[
\begin{align*}
\| x_j - \tilde{x}_j \| & \leq \xi_j \| x_0 \|^2 \\
\| b_j - \tilde{b}_j \| & \leq \lambda_j \| x_0 \|^2 \quad \text{for} \quad j = r, \ldots, n-1
\end{align*}
\]

holds.

**Part 4: Bounding the constants.**

Consider several cycles of the algorithm and let \( H^i \quad i = 0, 1, \ldots \) be the starting matrices for each cycle. The constants \( \bar{\Gamma} \) and \( \bar{\gamma} \) satisfy

\( \bar{\Gamma}(H^{i+1}) \leq \bar{\Gamma}(H^i) \) and \( \bar{\gamma}(H^{i+1}) \leq \bar{\gamma}(H^i) \). After a certain number of cycles the current \( H^i \) will be dropped and replaced by the starting matrix \( H_0 \) (in practice \( H_0 \) is a diagonal matrix). Therefore the constants \( \bar{\Gamma} \) and \( \bar{\gamma} \) are bounded. All of the conditions of Stoer's theory are satisfied and the theorem is proved. ///

The VSCG can be analyzed in a similar way, the main difference being that the functions \( \psi \) and \( B \) act simultaneously during some iterations.
CHAPTER VIII

AN APPLICATION TO MULTIPLIER METHODS

Using CG methods without line searches we will derive new multiplier update formulas. This section follows the presentation in Tapia [21].

Let $f: \mathbb{R}^n \to \mathbb{R}$ and $g: \mathbb{R}^n \to \mathbb{R}^m$ be twice continuously differentiable. Consider the problem

$$\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad g(x) = 0
\end{align*} \tag{8.1}$$

Define $F: \mathbb{R}^{n+m} \to \mathbb{R}$ and $L: \mathbb{R}^{n+m+1} \to \mathbb{R}$ by

$$F(x, \lambda) = f(x) + \lambda^T g(x) \tag{8.2}$$

$$L(x, \lambda, c) = f(x) + \lambda^T g(x) + (c/2) g(x)^T g(x) \text{ with } c \geq 0 . \tag{8.3}$$

We say that $x^*$ is a "nonsingular critical point" of problem (8.1) if there exists $\lambda^* \in \mathbb{R}^m$ such that $\nabla F(x^*, \lambda^*) = 0$ and $\nabla^2 F(x^*, \lambda^*)$ is invertible.

**Proposition 9.4 [22]** Let $x^*$ be a nonsingular solution of problem (8.1) with multiplier $\lambda^*$. Then there exists $\hat{c}$ such that for all $c \geq \hat{c}$ the matrix $\nabla^2_{xx} L(x^*, \lambda^*, c)$ is positive definite and $x^*$ is a locally unique solution to $\min L(x, \lambda^*, c) .

For $x^*, \lambda^*$ and $c$ as defined in Proposition 8.4 we can apply the implicit function theorem to $\nabla_x L(x^*, \lambda^*, c) = 0$ : there exists a neighborhood
$U(\lambda^*)$ and a unique function $X: U \subset \mathbb{R}^m \rightarrow \mathbb{R}^n$ such that

1. $X(\lambda^*) = x^*$,
2. $\nabla_x L(X(\lambda), \lambda, C) = 0$ for $\lambda \in U(\lambda^*)$,
3. $x \in C^1$ on $U(\lambda^*)$.

We can then define $h: U \rightarrow \mathbb{R}^m \rightarrow \mathbb{R}^n$ by

$$h(\lambda) = \min_x L(x, \lambda, C) \quad (8.6)$$

**Proposition 8.5** [9],[23] Consider the dual problem to (8.1):

$$\max_{\lambda \in U(\lambda^*)} h(\lambda) \quad (8.7)$$

If $x^*$ solves (8.1), its associated multiplier $\lambda^*$ solves the dual (8.7).

We have that $h(\lambda) = L(X(\lambda), \lambda, C)$, therefore

$$\nabla h(\lambda) = \nabla_x L(X(\lambda), \lambda, C) \Delta X(\lambda) + g(X(\lambda))$$

$$= g(X(\lambda)) \quad (8.8)$$

$$\nabla^2 h(\lambda) = \nabla^2 g(X(\lambda))^T \nabla^2 L(X(\lambda), \lambda, C)^{-1} \nabla^2 g(X(\lambda))$$

The above two Propositions suggest the following Primal-Dual Algorithm introduced by Hestenes [22] and known as the Method of Multipliers:
For given $\epsilon, \lambda_0, C_0$ and $k = 0,1,2\ldots$

(1) (Primal step) $x_k$ is the solution of $\min_x L(x, \lambda_k, C_k)$
   i.e., $x_k = X(\lambda_k)$

(2) If $\|\nabla L(x_k, \lambda_k, C_k)\| < \epsilon$ stop; else

(3) (Dual step) $C_{k+1} = w(x_k, \lambda_k, C_k)$

   $\lambda_{k+1} = v(x_k, \lambda_k, C_k)$;

where different choices for the functions $w$ and $v$ give rise to different algorithms. The Hestenes-Powell formula ([22],[24])

$$\lambda_{k+1} = \lambda_k + C_0 g(x_k)$$

(Using (8.8))

$$= \lambda_k + C_0 \nabla h(x_k)$$

is the steepest ascent method with stepsize $C_0$. Buy's formula [25] and Tapia's formula [27] which in this case are equivalent),

$$\lambda_{k+1} = \lambda_k + \left[\nabla^2 g(x_k)^T \nabla^2 L(x_k, \lambda_k, C_k)^{-1} \nabla g(x_k)\right]^{-1} g(x_k)$$

$$= \lambda_k + \left[\nabla^2 h(\lambda_k)\right]^{-1} \nabla h(\lambda_k)$$

give Newton's method. We could think on doing the CG method on the variable $\lambda$. Doing line searches is out of question, since to evaluate $h$ at some other point requires a complete minimization: $h(\lambda) = \min_x L(x, \lambda, C)$. We are therefore lead to consider the CG methods without line searches of Chapter 4. The GP, TTR and Multistep methods will each give us a multiplier update formula. For the TTR (4.12) we have
\[ \lambda_{k+2} = \lambda_{k+1} - y_k + \left( y_k^T y_k \right) d_k + \left( y_k^T y_{k-1} \right) d_{k-1} \]  

(8.9)

where \( y_j = g(x_{j+1}) - g(x_j) \), \( d_j = \lambda_{j+1} - \lambda_j \).

The correction step (4.7) will be

\[ \bar{\lambda}_{i+1} = \lambda_{i+1} - \sum_{j=1}^{i} \frac{g(x_{i+1}) d_j}{d_j y_j} \quad d_j \]  

(8.10)

In a practical implementation of this multiplier formula we would include a steplength in (8.9) using for example a quadratic model. The iteration should also be guarded against too long displacements.
CHAPTER IX

NUMERICAL EXAMPLES

The following problems were solved using the combined QN-CG methods. Our aim is to compare the performance and to see what effect in efficiency is obtained by increasing the number of updates stored. An IBM 370/168 and double precision arithmetic were used. We will call \textsc{store} the number of rank 1 corrections that can be stored. In general, the way of implementing the combined algorithms will depend on how large \textsc{store} is compared with \( n \). In practice we would use these algorithms only if \( n \) is large. Unless \textsc{store} is also large we think that the best approach is to build the metric as soon as possible, since doing many iterations but few updates will not scale the steps well enough.

The QN updates will be done by means of the symmetric rank 1 formula (see [33] for instance)

\[
\overline{H} = H + \frac{(H - s)(H - s)^T}{(H - s)^Ty}.
\]

(9.1)

This formula was used for its simplicity, even though the results of Chapters VI and VII were established only for Broyden's restricted class, which does not include it. Equation (9.1) is not defined if \((H - s)^Ty = 0\), therefore, when this quantity is small \((10^{-6})\) no updating was done.

The interleaved method was implemented as follows: we start by doing \textsc{store} iterations of QN steps; then we do either \( n + 1 \) or \( n + 1 - \textsc{store} \) iterations of CG steps. After this we restart.

The VSCG method was implemented similarly; during the first \textsc{store} iterations we update the matrix; it is held fixed for \( n \) or \( n - \textsc{store} \) iterations. Following this, we restart.

- 70 -
A general program of Larry Nazareth was used. The problems are documented in [35], they are the following standard test problems: Wood's function, Powell's 4 variable singular, Bigg's exponential of dimension 6 and the trigonometric functions. Tables 1 and 2 show the results for the problems of smaller dimension. GNORM stand for the final value of the norm of the gradient and NFCALL for the number of function evaluations. The results for the two methods are similar.

For higher dimensional problems our computational experiments seem to indicate that the VSCG method is more efficient than the Interleaved method. Table 3 shows the results for the VSCG on the trigonometric function of order 20.

**TABLE 1**

<table>
<thead>
<tr>
<th>Problem</th>
<th>STORE</th>
<th>NFCALL</th>
<th>GNORM</th>
<th>NFCALL</th>
<th>GNORM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wood</td>
<td>0</td>
<td>407</td>
<td>.23D-06</td>
<td>363</td>
<td>.13D-06</td>
</tr>
<tr>
<td>(n=4)</td>
<td>1</td>
<td>310</td>
<td>.26D-08</td>
<td>305</td>
<td>.85D-08</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>287</td>
<td>.26D-06</td>
<td>458</td>
<td>.92D-06</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>287</td>
<td>.27D-06</td>
<td>198</td>
<td>.15D-06</td>
</tr>
<tr>
<td>POW</td>
<td>0</td>
<td>251</td>
<td>.18D-06</td>
<td>127</td>
<td>.97D-06</td>
</tr>
<tr>
<td>(n=4)</td>
<td>1</td>
<td>204</td>
<td>.99D-06</td>
<td>121</td>
<td>.57D-06</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>98</td>
<td>.46D-06</td>
<td>101</td>
<td>.36D-06</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>98</td>
<td>.46D-06</td>
<td>177</td>
<td>.58D-06</td>
</tr>
<tr>
<td>BIGG</td>
<td>0</td>
<td>174</td>
<td>.86D-06</td>
<td>202</td>
<td>.18D-06</td>
</tr>
<tr>
<td>(n=6)</td>
<td>1</td>
<td>205</td>
<td>.63D-07</td>
<td>127</td>
<td>.11D-06</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>133</td>
<td>.45D-09</td>
<td>170</td>
<td>.43D-08</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>132</td>
<td>.18D-06</td>
<td>267</td>
<td>.40D-06</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>129</td>
<td>.35D-08</td>
<td>170</td>
<td>.34D-06</td>
</tr>
<tr>
<td>Problem (n)</td>
<td>STORE</td>
<td>NFCALL</td>
<td>GNORM</td>
<td>NFCALL</td>
<td>GNORM</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>Wood (n=4)</td>
<td>0</td>
<td>469</td>
<td>.72D-06</td>
<td>399</td>
<td>.22D-07</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>305</td>
<td>.85D-08</td>
<td>223</td>
<td>.64D-09</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>278</td>
<td>.25D-06</td>
<td>381</td>
<td>.62D-06</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>216</td>
<td>.86D-13</td>
<td>176</td>
<td>.44D-10</td>
</tr>
<tr>
<td>POW (n=4)</td>
<td>0</td>
<td>251</td>
<td>.18D-06</td>
<td>127</td>
<td>.97D-06</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>121</td>
<td>.57D-06</td>
<td>143</td>
<td>.34D-06</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>89</td>
<td>.53D-06</td>
<td>97</td>
<td>.16D-06</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>77</td>
<td>.94D-07</td>
<td>164</td>
<td>.33D-06</td>
</tr>
<tr>
<td>BIGG (n=6)</td>
<td>0</td>
<td>178</td>
<td>.74D-06</td>
<td>198</td>
<td>.67D-07</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>127</td>
<td>.11D-06</td>
<td>122</td>
<td>.76D-06</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>117</td>
<td>.35D-06</td>
<td>139</td>
<td>.44D-06</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>108</td>
<td>.67D-06</td>
<td>134</td>
<td>.83D-06</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>111</td>
<td>.29D-07</td>
<td>246</td>
<td>.15D-06</td>
</tr>
</tbody>
</table>

**TABLE 3**

<table>
<thead>
<tr>
<th>Problem (n=20)</th>
<th>STORE</th>
<th>NFCALL</th>
<th>GNORM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trigonometry</td>
<td>2</td>
<td>452</td>
<td>.19D-04</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>422</td>
<td>.13D-04</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>381</td>
<td>.15D-05</td>
</tr>
</tbody>
</table>
REFERENCES


[34] Gill and Murray, Report to Appear.

