



117
505
THS

FUNCTION MINIMIZING ALGORITHMS

Thesis for the Degree of Ph. D.
MICHIGAN STATE UNIVERSITY
DAVID GEORGE McDOWELL
1970

THESIS



This is to certify that the
thesis entitled


FUNCTION MINIMIZING ALGORITHMS

presented by

David George McDowell

has been accepted towards fulfillment
of the requirements for

Ph.D. degree in Mathematics


Major professor

Date 7-31-70

ABSTRACT

FUNCTION MINIMIZING ALGORITHMS

By

David George McDowell

We consider the problem of minimizing a real valued function f in n -dimensional Euclidean space by the method of conjugate directions.

First, the case when f is quadratic with a positive semi definite coefficient matrix is examined. Two examples of the conjugate direction method are considered, the Fletcher-Powell formulation of the Davidon algorithm and the conjugate gradient algorithm of M.R. Hestenes. An extension of the Fletcher-Powell method is given and shown to be theoretically equivalent to an extension of the conjugate gradient method given by M.R. Hestenes. As a result of this equivalence we show that the conjugate gradient and Fletcher-Powell methods are equivalent.

These two methods are then applied to the minimization of non-quadratic functions. Convergence is shown and conditions on the rates of convergence are derived. The rate of convergence for the conjugate gradient method is shown to be geometric and for the Fletcher-Powell method better than geometric.

Finally a general algorithm for the minimization of non-quadratic functions is given. The above two methods are essentially special cases of this algorithm. The rate of convergence for this general method is shown to be at least geometric.

FUNCTION MINIMIZING ALGORITHMS

By

David George McDowell

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

Department of Mathematics

1970

TABLE OF CONTENTS

Chapter	Page
I. INTRODUCTION	1
1. The Problem.	1
2. Preliminaries.	3
3. Iterative Procedures	9
II. GENERAL ALGORITHMS	18
4. Methods for Generating Conjugate Direction Vectors.	18
5. The Equivalence of Algorithms (4.1) and (4.3).	31
6. Converse to Theorem 4.1 and Theorem 4.2. .	35
7. The cg-Method.	39
8. The FP-Method.	41
9. Second Method for Constructing Vectors that are N-Conjugate	44
III. APPLICATIONS TO NON-QUADRATIC FUNCTIONS.	61
10. Algorithms for Non-quadratic Functions . .	61
11. The FP-algorithm for Non-quadratic Functions.	66
12. The cg-algorithm for Non-quadratic Functions.	70
13. Minimizing Non-quadratic Functions	77
BIBLIOGRAPHY	81

INTRODUCTION

1. The Problem.

This thesis deals with the study of iterative procedures for minimizing real valued functions on Euclidean n -space. It is divided into three parts, the first two parts deal with quadratic functions and the third part with non-quadratic functions.

In particular, the first chapter considers quadratic functions whose leading coefficients are positive semi-definite. The iteration method of conjugate directions is given along with a geometric interpretation. This material is an extension of results found in [1]* and [2] by E. Stiefel and M. R. Hestenes.

The second chapter starts with a description of the two main algorithms. The first is by M. R. Hestenes [2]. The second is an extension of Davidon's method [7] as formulated by Fletcher and Powell [3]. Properties of the algorithm are also given. The equivalence of the two algorithms is then proved. A special case of this equivalence is shown by G. Myers [5].

*Numbers in square brackets refer to bibliography.

Two special cases of the algorithms are given for minimizing a quadratic function with positive semi definite leading coefficient. These are the conjugate gradient method and the Fletcher-Powell formulation of the Davidon method. The case when the leading coefficient is only positive definite can be found in [1], [2], and [3]. It is shown that these two cases of the main algorithms are also equivalent. In [2], M. R. Hestenes proved that every conjugate direction method is a conjugate gradient method. From this and above results we are able to conclude that every conjugate direction method is also a Fletcher-Powell method.

The last section of Chapter 11 shows that minimizing the quadratic form by the method of conjugate directions is equivalent to performing Gauss elimination on the coefficient matrix. The case when the matrix is only positive definite is done in [1].

The final chapter applies the minimizing algorithms to non-quadratic functions. Convergence for the algorithms is shown and conditions on the rates of convergence are derived.

The Fletcher-Powell method is shown to converge faster than geometrically. Two forms of the conjugate gradient method are given, and both are shown to converge geometrically. The second scheme was suggested in [9]. In numerical tests [4], the second method did show better convergence than the first method.

The algorithms in the last chapter, rather than given in

terms of the function to be minimized, are presented in a form that is convenient for comparing and deriving convergence results. The equivalence between these algorithms and the minimization of a non-quadratic function is given in the last section.

2. Preliminaries.

We consider the quadratic form

$$(2.1) \quad E(x) = (r, Rr)$$

with

$$(2.2) \quad r = k - Ax$$

where R , and A are $n \times n$ dimensional matrices and x , k are n dimensional vectors. It will be assumed the elements of R , A , x , and k are real numbers, and the matrix R is positive definite. We are concerned with finding a minimum to (2.1) which is also a solution to the linear equation

$$(2.3) \quad Ax = k$$

(if one exists) by searching in conjugate directions.

Given any two vectors x and y where

$$x = (x^1, \cdot \cdot \cdot, x^n), \quad y = (y^1, \cdot \cdot \cdot, y^n)$$

we have

$$x+y = (x^1+y^1, \cdot \cdot \cdot, x^n+y^n)$$

$$ax = (ax^1, \cdot \cdot \cdot ax^n),$$

where a is a scalar. The inner product of two vectors is given by

$$(x, y) = x^1 y^1 + \dots + x^n y^n$$

and the length of the vector x is given by

$$|x| = (x, x)^{1/2}.$$

The conjugate transpose of a matrix A will be denoted by A^* and A^* satisfies the relation

$$(Ax, y) = (x, A^*y).$$

The matrix A is said to be positive definite if $(x, Ax) \geq 0$ for all x and equal to zero if and only if $x = 0$. It is positive semi definite if $(x, Ax) \geq 0$ for all x . If A is positive semi definite then $A = A^*$. The null space of a matrix A , indicated $n(A)$, is the set of all vectors that A maps into the zero vector. If A is positive semi definite and $(x, Ay) = 0$, with x and y not in the null space of A , then x and y are said to be A -orthogonal or A -conjugate.

At each vector or point x the vector defined by (2.2) will be called the residue at x . The function $E(x)$ defined by (2.1) will be called the error function. This comes from the fact that $E(x)$ is a measure of accuracy with which x approximates a solution to (2.3).

The error function $E(x) = 0$ if and only if x is a solution to (2.3), that is $r(x) = 0$. The error function is a positive quadratic form and in the following work we will be

looking for a minimum to $E(x)$. In the case where (2.3) has a solution then a point minimizes $E(x)$ if and only if it is also a solution to (2.3). Let M be the set of all points that minimize $E(x)$, that is

$$M = [x | E(x) = \text{minimum}].$$

If A is nonsingular then M contains a single point. If A is singular then M generally contains more than one point.

It should be noted that any positive quadratic form can be put in the form of (2.1) plus a constant. Therefore our methods for minimizing (2.1) hold for all positive quadratic forms.

Given two vectors x and y we have

$$(2.4) \quad E(x+y) = E(x) - 2(g,y) + (y,Ny)$$

where

$$g = A^*Rr$$

$$N = A^*RA$$

where r is given by (2.2) and A is an arbitrary $n \times n$ matrix. The matrix N is positive semi definite. The vector g will be called the gradient of $E(x)$ at x . It differs from the usual gradient by $-1/2$.

Setting $y = tp$ in equation (2.4), where t is a scalar, we obtain

$$(2.5) \quad E(x) - E(x+tp) = 2t(g,p) - t^2(p,Np).$$

Using the following lemma we will obtain some properties of the error function.

Lemma 2.1 Given $N = A*RA$ where R is positive definite then the following hold

- a) p is in $n(N)$ if and only if p is in $n(A)$.
- b) p is in $n(N)$ if and only if $(p, Np) = 0$.
- c) p is in $n(A)$ implies $(g, p) = 0$.

proof. p is in $n(A)$ implies p is in $n(N)$ by the definition of N . If p is in $n(N)$ then

$$0 = (p, Np) = (p, A*RAp) = (Ap, RAp).$$

Thus, $Ap = 0$ and p is in $n(A)$. This completes the proof of a). b) follows from a) and the proof of a). To show c) we have

$$(g, p) = (p, A*Rr) = (Ap, Rr) = 0$$

since $Ap = 0$.

We can now derive some elementary properties of $E(x)$.

Lemma 2.2 Given vectors x and p , not in $n(N)$, the function $E(x+tp)$ considered as a function of t is minimized when $t = a$, where

$$a = \frac{(g, p)}{(p, Np)}.$$

We also have

$$E(x) - E(x+ap) = a^2(p, Np),$$

and p is in n(N) if and only if

$$E(x) - E(x+tp) = 0$$

for all t.

Proof. If p is not in n(N) then by lemma 2.1 we have $(p, Np) \neq 0$. Hence minimizing $E(x) - E(x+tp)$ as a function of t yields $t = a$ and

$$E(x) - E(x+ap) = 2a(g, p) - a^2(p, Np) = a^2(p, Np).$$

The last part follows directly from (2.5) and lemma 2.1.

Lemma 2.3 Let B be a linear subspace of E^n , then

$$E(x+y) \geq E(x)$$

for all y in B if and only if g is orthogonal to B.

Proof. Equation (2.4) implies inequality holds whenever $(g, y) = 0$. Clearly $n(N)$ is contained in B by lemma 2.2. If $(g, p) \neq 0$ for some vector p not in $n(N)$ and p in B, then from lemma 2.2

$$E(x) - E(x+ap) = a^2(p, Np) > 0$$

and the inequality does not hold when $y = ap$.

Recall the set M was defined as the set of points that minimize the function $E(x)$. We now obtain an expression for the set M.

Lemma 2.4 $M = n(N) + h$ where h satisfies the inequality

$$E(h) \leq E(x)$$

for all x in E^n .

Proof. If v is in the $n(N)$ then by (2.4)

$$E(h+v) = E(h) - 2(g,v) + (v,Nv) = E(h)$$

by lemma 2.1. Thus $h + n(N)$ is contained in M . Let \bar{h} be in M and set $v = \bar{h} - h$. Then $\bar{h} = h + v$ and $E(\bar{h}) - E(h) = 0$ by the definition of M . Suppose v is not in $n(N)$ then lemma 2.1 implies $(v,Nv) \neq 0$. We have

$$E(h) - E(h+av) = a^2(v,Nv) \leq 0$$

since h minimizes $E(x)$. Thus $a = 0$ and $(g,v) = 0$. Therefore

$$0 = E(h) - E(\bar{h}) = E(h) - E(h+v) = -(v,Nv) .$$

This is a contradiction to the assumption v is not in $n(N)$. Therefore v is in $n(N)$ and M is contained in the set $h + n(N)$.

Lemma 2.5 $g(x) = 0$ if and only if $x \in M$.

Proof. $g(x) = 0$ implies

$$E(x) - E(x+tp) = -t^2(p,Np) \leq 0$$

for all t and all p , and hence x is in M . Suppose $g(x) \neq 0$ then if x is in M we have $E(x) - E(y) \leq 0$ for all y in E^n . In particular let $y = x+tg$, then

$$E(x) - E(x+tg) = 2t|g|^2 - t^2(g,Ng) > 0$$

for t sufficiently small. This is a contradiction to x being in M . Therefore $g(x) = 0$.

These lemmas give us some of the basic properties concerning the error function and the linear manifold M . Before considering finite iteration methods we note that

$$\begin{aligned} g(x) &= A^*Rr(x) = A^*R(k - Ax) \\ &= \bar{k} - Nx \end{aligned}$$

where $\bar{k} = A^*Rk$.

The previous lemma shows that x is in M if and only if $\bar{k} - Nx = 0$. We know from preliminary remarks that x in M solves $Ax = k$ if and only if $Ax = k$ has a solution. Therefore the two linear systems $Ax = k$ and $Nx = \bar{k}$ are equivalent if and only if $Ax = k$ has a solution.

3. Iterative procedures.

We will be considering iterations

$$(3.1) \quad x_{i+1} = x_i + t_i p_i \quad i = 0, 1, \dots$$

such that x_m is in M , $m \leq n$. Here p_i is a direction vector and t_i is a scalar. The iterative procedures will be used to minimize the error function $E(x)$. We set

$$(3.2) \quad \begin{aligned} r_i &= k - Ax_i & a_i &= c_i/d_i \\ g_i &= A^*Rr_i & c_i &= (g_i, p_i) \\ N &= A^*RA & d_i &= (p_i, Np_i) \end{aligned}$$

and we have from lemma 2.2 $E(x_{i+1}) - E(x_i)$ attaining its minimum when $t_i = a_i$ and p_i is not in $n(N)$, and $E(x_{i+1}) - E(x_i) = 0$ when p_i is in $n(N)$. Therefore for an arbitrary initial point x_0 our iteration will take the form

$$(3.3) \quad \begin{aligned} x_{i+1} &= x_i + a_i p_i \\ r_{i+1} &= r_i - a_i A p_i \end{aligned} \quad r_0 = k - A x_0$$

with

$$a_i = \begin{cases} \frac{(g_i, p_i)}{(p_i, N p_i)} & p_i \text{ not in } n(N) \\ 0 & p_i \text{ in } n(N). \end{cases}$$

Since $E(x_{i+1}) - E(x_i) = 0$ for all t_i when p_i is in $n(N)$ and $x_{i+1} = x_i + t_i p_i$, we pick $t_i = a_i = 0$ for convenience. Thus the iteration will be completely determined by our choice of p_0, p_1, \dots where x_0 and R have already been chosen.

We have the following definitions concerning p_0, p_1, \dots, p_r , $r \leq n-1$. We say the vectors p_0, p_1, \dots, p_r are linear independent if

$$\sum_{i=0}^r a_i p_i = 0 \text{ implies } a_i = 0, i = 0, \dots, r.$$

If B is an $n \times n$ matrix then the vectors p_0, p_1, \dots, p_r are B -linear independent if

$$\sum_{i=0}^r a_i p_i \text{ in } n(B) \text{ implies } a_i = 0, i = 0, 1, \dots, r.$$

Clearly B -linear independence implies linear independence and if B is nonsingular then the two definitions are equivalent.

Let $m = \text{rank of } A$, then $n-m = \text{dimension of } n(A)$. We also have $m = \text{rank of } N$ and $n-m = \text{dimension of } n(N)$. The next two lemmas give us the properties concerning the

vectors p_0, p_1, \dots .

Lemma 3.1 If m vectors, $m \leq n$, are chosen so that

$$(3.4) \quad d_i = (p_i, Np_i) > 0, \quad (p_i, Np_j) = 0 \quad i \neq j$$

then p_0, \dots, p_{m-1} are N -linear independent and hence also A -linear independent. Independent of the choice of x_0 , the vector x_m defined by (3.3) is in M . We also have the following conditions satisfied,

$$(3.5) \quad (g_i, p_j) = 0 \quad j = 0, \dots, i-1$$

$$(3.6) \quad (g_i, p_i) = (g_j, p_i) \quad j = 0, \dots, i-1.$$

Proof. We first note that $d_i > 0$ implies that p_i is not in $n(N)$. Condition (3.6) shows that if

$$a_i = \frac{(g_i, p_i)}{(p_i, Np_i)}$$

then

$$a_i = \frac{(g_0, p_i)}{(p_i, Np_i)}.$$

We note

$$\sum_{i=0}^{m-1} c_i p_i \text{ in } n(N) \text{ means } N\left(\sum_{i=0}^{m-1} c_i p_i\right) = 0,$$

therefore

$$\sum_{i=0}^{m-1} c_i Np_i = 0 \text{ and } c_j (p_j, Np_j) = 0$$

for $j = 0, \dots, m-1$. Hence $c_j = 0$, $j = 0, \dots, m-1$ and p_1, \dots, p_{m-1} are N and also A -linear independent. We next prove conditions (3.5) and (3.6). We have

$$g_{i+1} = A^* R r_{i+1} = g_i - a_i N p_i$$

and hence

$$(g_{i+1}, p_j) = (g_i, p_j) - a_i (N p_i, p_j).$$

For $i \neq j$ this becomes by (3.4)

$$(g_{i+1}, p_j) = (g_i, p_j),$$

and the formula for a_i implies $(g_{i+1}, p_i) = 0$. Therefore we get $(g_i, p_j) = 0$ for $i > j$. We also obtain

$$(g_{i+1}, p_j) = (g_i, p_j) - a_i (N p_i, p_j) = (g_i, p_j), \quad j > i+1$$

which implies

$$(g_{i-1}, p_i) = (g_{i-2}, p_i) = \dots = (g_0, p_i).$$

This completes the proof of (3.5) and (3.6). To show x_m is in M we note that

$$(g_m, p_j) = 0 \quad j = 0, \dots, m-1$$

and g_m is orthogonal to $n(N)$ by lemma 2.1. The dimension of $n(N)$ is $n-m$, thus g_m is orthogonal to E^n . Hence $g_m = 0$ and lemma 2.5 implies x_m is in M .

We have as a converse to lemma 3.1 the following lemma.

Lemma 3.2 Let vectors p_0, p_1, \dots, p_{m-1} be A-linear independent.

Then there exists a positive definite matrix R such that (3.4) holds with $N = A^*RA$. If in iteration (3.1) x_m is in M then $t_i = a_i$.

Proof. Let P be a matrix whose columns are p_0, \dots, p_{m-1} . Let $Q = AP$ with columns v_0, \dots, v_{m-1} , that is $v_i = Ap_i, i = 0, \dots, m-1$. The vectors v_0, \dots, v_{m-1} are linear independent since the vectors p_0, \dots, p_{m-1} are A-linear independent. Extend v_0, \dots, v_{m-1} to a basis for E^n . Thus we have n linear independent vectors v_0, \dots, v_{n-1} . Let \bar{Q} be the nonsingular matrix whose columns are v_0, \dots, v_{n-1} . If we set $R = (\bar{Q}\bar{Q}^*)^{-1}$, then R is a positive definite matrix. We have

$$\bar{Q}^*R\bar{Q} = \bar{Q}^*(\bar{Q}\bar{Q}^*)^{-1}\bar{Q} = \bar{Q}^*\bar{Q}^{-1} = I,$$

that is

$$(v_i, Rv_j) = 0 \quad i \neq j \quad i, j = 0, \dots, n-1$$

and

$$(v_i, Rv_i) = 1.$$

Therefore if we set $N = A^*RA$ we have

$$P^*NP = (AP)^*R(AP) = Q^*RQ = I,$$

and (3.4) holds.

To show $t_i = a_i$ we note that

$$x_m = x_1 + t_1 p_1 + \dots + t_{m-1} p_{m-1}$$

and $r_m = k - Ax_m$. Since x_m is in M we have $g_m = 0$, therefore

$$g_m = A^* R r_m = A^* R (k - A x_m) = \bar{k} - N x_m = 0.$$

Now

$$\begin{aligned} g_i &= A^* R r_i = \bar{k} - N x_i = N(x_m - x_i) \\ &= t_i N p_i + \dots + t_{m-1} N p_{m-1}. \end{aligned}$$

Therefore

$$(g_i, p_i) = t_i (N p_i, p_i)$$

with p_i not in $n(N)$ and hence

$$t_i = \frac{(g_i, p_i)}{(p_i, N p_i)} = a_i.$$

An iteration (3.3) in which the vectors p_0, \dots, p_{m-1} are chosen as in (3.4) is called the method of conjugate directions (cd-method) by M. R. Hestenes and E. Stiefel [1].

We have the following theorem concerning the cd-method.

Theorem 3.1 In the cd-method the point x_i minimizes $E(x)$ on the line $x = x_{i-1} + t p_{i-1}$ and also on the i -dimensional plane of points

$$x = x_0 + t_0 p_0 + \dots + t_{i-1} p_{i-1}.$$

The point x_i is the center of the $(i-1)$ -dimensional ellipsoid which is the intersection of this plane with the level curve $E(x) = E(x_0)$.

This theorem follows from lemmas 2.2, 2.3 and 3.1. This

theorem differs from theorem 3.1 in [2] because there the level curve $E(x) = E(x_0)$ is an $(n-1)$ -dimensional ellipsoid whereas here this level curve is an $(m-1)$ -dimensional ellipsoidal cylinder with center M . Recall the set $M = h + n(N)$, where h is a point that minimizes $E(x)$.

When the matrix A is nonsingular the level curves $E(x) = \text{constant}$ form a one-parameter family of $(n-1)$ -dimensional ellipsoids with center M (a single point), the solution to $Ax = k$.

In the case where A is singular and has rank m then the level curves $E(x) = \text{constant}$ form a one-parameter family of $(m-1)$ -dimensional ellipsoidal cylinders with center M .

In order to see this we consider the linear mapping

$$A: E^n \rightarrow D,$$

where D is an m -dimensional linear subspace of E^n , and where dimension $n(A) = n-m$. We have

$$E^n = n(A) + B_m$$

where B_m is an m -dimensional linear subspace of E^n and

$$A: B_m \rightarrow D$$

one-to-one. Let \bar{A} be a nonsingular matrix such that $\bar{A}x = Ax$ for all x in B_m . We have $\bar{E}(x) = E(x)$ for all x in B_m where

$$\bar{E}(x) = (k - \bar{A}x, R(k - \bar{A}x)).$$

The level curves $\bar{E}(x) = \text{constant}$ form a one-parameter family

of $(n-1)$ -dimensional ellipsoids. Thus the intersection of the level curves $\bar{E}(x) = \text{constant}$ and B_m form a one-parameter family of $(m-1)$ -dimensional ellipsoids in B_m . But since $E(x)$ and $\bar{E}(x)$ agree on B_m then the intersection of the level curves $E(x) = \text{constant}$ and B_m are $(m-1)$ -dimensional ellipsoids in B_m where the center minimizes $E(x)$ in E^n . We also have

$$E^n = n(A) + B_m$$

and

$$E(x+q) = E(x)$$

for all q in $n(A)$, hence the level curves $E(x) = \text{constant}$ are $(m-1)$ -dimensional ellipsoidal cylinders with center axis M .

If in theorem 3.1 we let

$$B_i = \left\{ \sum_{j=0}^{i-1} t_j p_j \mid t_j \text{ arbitrary} \right\},$$

$i = 1, \dots, m$, then B_i is an i -dimensional linear subspace generated by i A -linear independent vectors and B_m is as above. We have $B_i \subset B_{i+1}$ for each i , hence the level curve $E(x) = \text{constant}$ intersects B_i in an $(i-1)$ -dimensional ellipsoid.

In the cd-method we have an arbitrary initial point x_0 . This results in a displacement of each B_i by an amount x_0 . The geometrical interpretation is the same as above except we are dealing with hyperplanes that are not subspaces. The points x_0, \dots, x_i are in $x_0 + B_i$ and x_i is the center of

the $(l-1)$ -dimensional ellipsoid which is the intersection of the level curve $E(x) = \text{constant}$ and $x_0 + B_1$. The point x_m minimizes $E(x)$ and is the intersection of $x_0 + B_m$ and M .

The previous two sections are an extension of the results found in [1] and [2], to the case where A is a singular matrix.

We now consider various methods for generating vectors that satisfy condition (3.4).

GENERAL ALGORITHMS

4. Methods for generating conjugate direction vectors.

Given a positive semi definite matrix N we can generate vectors that are N -conjugate by taking n basis vectors u_1, \dots, u_n and then N -orthogonalizing them by a Gram-Schmidt process. This method will be looked at in section 9.

In this section we will present two algorithms. The first is by M. R. Hestenes and can be found in [2]. The second method is an extension of the Davidon algorithm, as formulated by Fletcher and Powell in [3]. We will show in section 5 that the two algorithms are equivalent.

The following theorem is due to M. R. Hestenes and can be found in [2].

Theorem 4.1 Let K and N be positive semi definite matrices and let g_0 be an arbitrary vector not in $n(K)$.

The algorithm

$$\begin{aligned}
 (4.1) \quad & p_0 = Kg_0 \quad g_{i+1} = g_i - a_i Np_i \\
 & p_{i+1} = Kg_{i+1} + b_i p_i \\
 & a_i = c_i / d_i, \quad b_i = e_i / d_i \\
 & d_i = (p_i, Np_i), \quad c_i = (g_i, p_i) \\
 & e_i = - (Np_i, Kg_{i+1})
 \end{aligned}$$

generates non-zero vectors g_0, g_1, \dots and p_0, p_1, \dots satisfying the relations

$$(4.2) \quad (g_i, Kg_j) = 0, (p_i, Np_j) = 0 \quad i \neq j.$$

The algorithm will terminate in r steps where $r \leq \min(m, s)$ where m is the rank of N and s is the rank of K . At the r^{th} step one of the following will hold. i) $g_r = 0$, ii) $g_r \neq 0$, $Kg_r = 0$, iii) $Kg_r \neq 0$, $Np_r = 0$ with $p_r \neq 0$.

Corollary 1. If K is positive definite and $(z, g_0) = 0$ for all z in $n(N)$ then algorithm (4.1) will terminate in r steps, $r \leq \text{rank of } N$, if and only if $g_r = 0$.

Proof. Clearly case ii) of theorem 4.1 cannot occur with K positive definite. If $(z, g_0) = 0$ for all z in $n(N)$ then

$$(z, g_1) = (z, g_0) - a_0 (z, Np_0) - \dots - a_{l-1} (z, Np_l) = 0$$

for all z in $n(N)$ and $l \leq r$. In case iii) $Np_r = 0$ means

$$p_r = Kg_r + b_{r-1} p_{r-1}$$

is in $n(N)$. Therefore

$$\begin{aligned} 0 &= (g_r, p_r) = (g_r, Kg_r) - b_{r-1} (g_r, p_{r-1}) \\ &= (g_r, Kg_r). \end{aligned}$$

Thus $g_r = 0$ and algorithm terminates in r steps if and only if $g_r = 0$.

Corollary 2 If both K and N are positive definite then

(4.1) will terminate in r steps, $r \leq n$, if and only if $g_r = 0$.

It will be convenient to describe the material dealing with the Davidon algorithm as formulated by Fletcher and Powell in terms of the Dirac bracket notation. Fletcher and Powell used this notation in [3]. The reason for this is that outer products of vectors occur in this algorithm and they have a convenient formulation in this notation. A column vector (x^1, \dots, x^n) is written as $|x\rangle$. The row vector with these same elements is denoted by $\langle x|$. The scalar product of $\langle x|$ and $|y\rangle$ is written $\langle x|y\rangle$. The notation $|x\rangle\langle y|$ is the outer or vector product of the two vectors $|x\rangle$ and $\langle y|$. It is the matrix whose elements are $(x_i y_j)$. If H is an $n \times n$ matrix then $H|x\rangle$ is a column vector and $\langle x|H$ is a row vector and $\langle x|H|y\rangle$ is a scalar.

The following theorem gives an algorithm that generates vectors which are N -conjugate. It is an extension of the Davidon algorithm as formulated by Fletcher and Powell [3].

Theorem 4.2 Given positive semi definite matrices H_0 and N and a vector g_0 not in $n(H_0)$ then the algorithm

$$\begin{aligned} |s_0\rangle &= H_0 |g_0\rangle & |g_{i+1}\rangle &= |g_i\rangle - a_i N |s_i\rangle \\ |s_{i+1}\rangle &= H_{i+1} |g_{i+1}\rangle \\ H_{i+1} &= H_i + A_i + B_i, \end{aligned}$$

(4.3) where

$$A_i = \frac{|s_i\rangle\langle s_i|}{\langle s_i|y_i\rangle}, \quad B_i = - \frac{H_i |y_i\rangle\langle y_i| H_i}{\langle y_i|H_i|y_i\rangle}$$

and

$$|\sigma_i\rangle = a_i |s_i\rangle, \quad |y_i\rangle = |g_i\rangle - |g_{i+1}\rangle$$

$$a_i = c_i/d_i, \quad c_i = \langle g_i | s_i \rangle, \quad d_i = \langle s_i | N | s_i \rangle$$

generates non-zero vectors s_0, s_1, \dots satisfying the relation

$$\langle s_i | N | s_j \rangle = 0 \quad i \neq j.$$

The algorithm terminates in r steps when one of the following holds. i) $|g_r\rangle = 0$, ii) $|g_r\rangle \neq 0, H_{r-1}|g_r\rangle = 0$,

iii) $H_{r-1}|g_r\rangle \neq 0, N|s_r\rangle = 0$ with $|s_r\rangle \neq 0$.

Theorem 4.2 will be proved by means of two lemmas.

Lemma 4.1 The scalars a_i, c_i , and d_i are positive for
 $i < r$. Moreover, the following hold

- (4.4)
- a) $\langle \sigma_i | g_{i+1} \rangle = 0$
 - b) $H_{i+1} N | \sigma_i \rangle = | \sigma_i \rangle$
 - c) $\langle \sigma_i | N | \sigma_{i+1} \rangle = 0$
 - d) $\langle x | H_i | x \rangle \geq 0$ for all x
 - e) $\langle g_i | H_i | g_{i+1} \rangle = 0$
 - f) $\langle \sigma_{i-1} | g_{i+1} \rangle = 0$
 - g) $\langle y_i | H_i | y_i \rangle > 0, i < r$.

The algorithm terminates in r steps if and only if one of the following holds. i) $|g_r\rangle = 0$, ii) $|g_r\rangle \neq 0, H_{r-1}|g_r\rangle = 0$,

iii) $H_{r-1}|g_r\rangle \neq 0, N|s_r\rangle = 0$, in this case $|s_r\rangle \neq 0$.

Proof. a_i is chosen so that $\langle s_i | g_{i+1} \rangle = 0$, therefore

$$\langle \sigma_i | g_{i+1} \rangle = a_i \langle s_i | g_{i+1} \rangle = 0.$$

To show b) we have

$$\begin{aligned} H_{i+1} N | \sigma_i \rangle &= H_{i+1} | y_i \rangle = H_i | y_i \rangle + A_i | y_i \rangle + B_i | y_i \rangle \\ &= H_i | y_i \rangle + | \sigma_i \rangle - H_i | y_i \rangle = | \sigma_i \rangle. \end{aligned}$$

For c)

$$0 = \langle g_{i+1} | \sigma_i \rangle = \langle g_{i+1} | H_{i+1} N | \sigma_i \rangle = \langle s_{i+1} | N | \sigma_i \rangle$$

implying

$$\langle \sigma_{i+1} | N | \sigma_i \rangle = 0.$$

We prove d) by induction. We are given H_0 positive semi definite. Assume H_1, \dots, H_k are positive semi definite and let $|p\rangle = (H_k)^{1/2} |x\rangle$, $|q\rangle = (H_k)^{1/2} |y_k\rangle$ with $|q\rangle \neq 0$. For $k < r$ we will show in 4.4 g) that $|q\rangle \neq 0$. Then

$$\begin{aligned} \langle x | H_{k+1} | x \rangle &= \frac{\langle p | p \rangle \langle q | q \rangle - \langle p | q \rangle^2}{\langle q | q \rangle} + \frac{\langle x | \sigma_k \rangle^2}{\langle \sigma_k | y_k \rangle} \\ &\geq \frac{\langle x | \sigma_k \rangle^2}{\langle \sigma_k | y_k \rangle}, \end{aligned}$$

since

$$\langle p | p \rangle \langle q | q \rangle - \langle p | q \rangle^2 \geq 0$$

by Schwartz's Inequality. Also

$$\langle \sigma_k | y_k \rangle = \langle \sigma_k | N | \sigma_k \rangle > 0, \quad k < r.$$

Therefore $\langle x | H_{k+1} | x \rangle \geq 0$ for all x .

To show e) we observe that

$$\langle g_i | H_i | g_{i+1} \rangle = \langle s_i | g_{i+1} \rangle = 0.$$

Now

$$\langle \sigma_{i-1} | g_{i+1} \rangle = \langle \sigma_{i-1} | g_i \rangle - \langle \sigma_{i-1} | N | \sigma_i \rangle = 0$$

establishing f).

For g) we note that

$$\begin{aligned} \langle y_i | H_i | y_i \rangle &= \langle g_i - g_{i+1} | H_i | g_i - g_{i+1} \rangle \\ &= \langle g_i | H_i | g_i \rangle + \langle g_{i+1} | H_i | g_{i+1} \rangle. \end{aligned}$$

Since $i < r$ and H_i is positive semi definite then

$$\langle g_i | H_i | g_i \rangle > 0 \text{ and } \langle g_{i+1} | H_i | g_{i+1} \rangle \geq 0$$

and therefore

$$\langle y_i | H_i | y_i \rangle > 0, \quad i < r.$$

The scalars a_i , c_i , and d_i are positive for $i < r$ since

$$c_i = \langle g_i | s_i \rangle = \langle g_i | H_i | g_i \rangle > 0$$

and

$$d_i = \langle s_i | N | s_i \rangle > 0$$

and

$$a_i = c_i / d_i.$$

We now prove the last statement of the lemma. If

$|g_r\rangle = 0$ then $|s_r\rangle = H_r|g_r\rangle = 0$ and the algorithm terminates.

If $|g_r\rangle \neq 0$ and $H_{r-1}|g_r\rangle = 0$ then

$$|s_r\rangle = H_r|g_r\rangle = H_{r-1}|g_r\rangle + A_{r-1}|g_r\rangle + B_{r-1}|g_r\rangle$$

with $A_{r-1}|g_r\rangle = 0$ and $B_{r-1}|g_r\rangle = 0$, thus $|s_r\rangle = 0$ and

algorithm terminates. If $H_{r-1}|g_r\rangle \neq 0$ and $N|s_r\rangle = 0$

then $|y_r\rangle = N|s_r\rangle = 0$ and hence H_{r+1} is undefined since

both A_r and B_r are undefined. Therefore the algorithm

terminates in r steps. In this case we have $|s_r\rangle \neq 0$ since

if $|s_r\rangle = 0$ then $H_r|g_r\rangle = 0$, and

$$H_{r-1}|g_r\rangle + B_{r-1}|g_r\rangle = 0.$$

Writing this equation out gives us

$$(1+t) H_{r-1}|g_r\rangle - t|s_{r-1}\rangle = 0$$

where

$$t = \frac{\langle y_{r-1} | H_{r-1} | g_r \rangle}{\langle y_{r-1} | H_{r-1} | y_{r-1} \rangle}.$$

Thus if $t \neq 0$ then

$$0 = \langle g_r | s_{r-1} \rangle = \frac{1+t}{t} \langle g_r | H_{r-1} | g_r \rangle$$

implying $H_{r-1}|g_r\rangle = 0$, which is a contradiction. If

$t = 0$ then

$$0 = H_r |g_r\rangle = H_{r-1} |g_r\rangle .$$

This is again a contradiction since in case iii) it is assumed $H_{r-1} |g_r\rangle \neq 0$.

If none of the three conditions hold then $|y_r\rangle = N|\sigma_r\rangle \neq 0$. Also $H_r |y_r\rangle \neq 0$ for if $H_r |y_r\rangle = 0$ then $H_r |g_r\rangle = H_r |g_{r+1}\rangle$ and

$$\langle g_r | H_r |g_r\rangle = \langle g_r | H_r |g_{r+1}\rangle = \langle s_r | g_{r+1}\rangle = 0 .$$

Hence $|s_r\rangle = H_r |g_r\rangle = 0$ and $N|\sigma_r\rangle = 0$, which is a contradiction to the assumption $N|\sigma_r\rangle \neq 0$; thus $H_r |y_r\rangle \neq 0$. We therefore have

$$\langle y_r | \sigma_r \rangle = \langle \sigma_r | N | \sigma_r \rangle > 0 \text{ and } \langle y_r | H_r | y_r \rangle > 0 ,$$

hence both A_r and B_r are defined and H_{r+1} is a positive semi definite matrix. Thus the algorithm does not terminate at the r^{th} step and

$$|s_{r+1}\rangle = H_{r+1} |g_{r+1}\rangle , \text{ with } |g_{r+1}\rangle = |g_r\rangle - N|\sigma_r\rangle .$$

This completes the proof of the lemma.

Lemma 4.2 The vectors s_0, s_1, \dots generated by
algorithm (4.3) satisfy the following relations:

$$(4.5) \quad \begin{array}{ll} \text{a)} & \langle s_i | N | s_j \rangle = 0 \quad i \neq j \\ \text{b)} & H_k N | \sigma_j \rangle = | \sigma_j \rangle \quad j < k \\ \text{c)} & \langle s_i | g_j \rangle = 0 \quad i = 0, \dots, j-1 \\ \text{d)} & \langle s_i | g_j \rangle = c_i \quad j = 0, \dots, i \end{array}$$

Proof. In view of lemma 4.1 relations (4.5) a), b) and c) hold for $i, j \leq 1$. Assume they hold for $i, j \leq k$ and we shall prove they hold for $i, j \leq k+1$. We have

$$|g_k\rangle = |g_{k-1}\rangle - N|\sigma_{k-1}\rangle = |g_{1+1}\rangle - N|\sigma_{1+1}\rangle - \dots - N|\sigma_{k-1}\rangle$$

and

$$\begin{aligned}\langle\sigma_i|g_k\rangle &= \langle\sigma_i|g_{1+1}\rangle - \langle\sigma_i|N|\sigma_{1+1}\rangle - \dots - \langle\sigma_i|N|\sigma_{k-1}\rangle \\ &= 0 \text{ for } i < k.\end{aligned}$$

Also, we have

$$\langle\sigma_i|g_{k+1}\rangle = \langle\sigma_i|g_k\rangle - \langle\sigma_i|N|\sigma_k\rangle = 0$$

for $i < k+1$. From 4.5 b) we obtain

$$\begin{aligned}\langle\sigma_i|N|\sigma_k\rangle &= a_k \langle\sigma_i|N|s_k\rangle = a_k \langle\sigma_i|NH_k|g_k\rangle \\ &= a_k \langle\sigma_i|g_k\rangle = 0 \text{ for } i < k.\end{aligned}$$

Since a_k is positive then $\langle\sigma_i|g_k\rangle = 0$. From this we observe that

$$\langle y_k|H_k N|\sigma_i\rangle = \langle y_k|\sigma_i\rangle = \langle\sigma_k|N|\sigma_i\rangle = 0$$

for $i < k$. Therefore

$$\begin{aligned}H_{k+1}N|\sigma_i\rangle &= H_k N|\sigma_i\rangle + \frac{|\sigma_k\rangle\langle\sigma_k|N|\sigma_i\rangle}{\langle\sigma_k|y_k\rangle} \\ &- \frac{H_k|y_k\rangle\langle y_k|H_k N|\sigma_i\rangle}{\langle y_k|H_k|y_k\rangle} = H_k N|\sigma_i\rangle = |\sigma_i\rangle\end{aligned}$$

for $i < k$ and

$$H_{k+1}N|\sigma_k\rangle = |\sigma_k\rangle.$$

From this we get

$$\begin{aligned} 0 &= \langle g_{k+1} | \sigma_i \rangle = \langle g_{k+1} | H_{k+1} N | \sigma_i \rangle \\ &= \langle s_{k+1} | N | \sigma_i \rangle \quad \text{for } i < k + 1. \end{aligned}$$

Hence $\langle s_i | N | s_j \rangle = 0 \quad i, j \leq k + 1$

and (4.5) a), b) and c) are proved for $i, j \leq k + 1$.

To prove d) we have

$$\langle s_1 | g_0 \rangle = \langle s_1 | g_1 \rangle + a_0 \langle s_1 | N | s_0 \rangle = \langle s_1 | g_1 \rangle = c_1.$$

Assume d) holds for $j \leq k$, that is

$$\langle s_k | g_j \rangle = c_k \quad j = 0, \dots, k.$$

Then we note that

$$\begin{aligned} \langle s_{k+1} | g_j \rangle &= \langle s_{k+1} | g_{k+1} \rangle + a_j \langle s_{k+1} | N | s_j \rangle \\ &+ \dots + a_k \langle s_{k+1} | N | s_k \rangle = \langle s_{k+1} | g_{k+1} \rangle = c_{k+1} \end{aligned}$$

and lemma 4.2 is proved.

Theorem 4.2 follows from lemmas 4.1 and 4.2.

The vectors s_0, s_1, \dots, s_{r-1} and $\sigma_0, \sigma_1, \dots, \sigma_{r-1}$ are N -linear independent and hence also linear independent.

The next two lemmas give some properties of the matrices H_i , $i = 0, \dots, r$.

Lemma 4.3 $n(H_0) = n(H_1)$, $i > 0$.

Proof. Assume z is in $n(H_1)$, $1 \leq k$ and show that z is in $n(H_{k+1})$. We have

$$\begin{aligned} H_{k+1}|z\rangle &= H_k|z\rangle + A_k|z\rangle + B_k|z\rangle \\ &= A_k|z\rangle = \frac{|\sigma_k\rangle\langle\sigma_k|z\rangle}{\langle\sigma_k|y_k\rangle}. \end{aligned}$$

We note that $|\sigma_k\rangle = 1/a_k H_k|g_k\rangle$, hence

$$\langle\sigma_k|z\rangle = 1/a_k \langle g_k|H_k|z\rangle = 0$$

and therefore $A_k|z\rangle = 0$. Thus $H_{k+1}|z\rangle = 0$ and z is in $n(H_{k+1})$. We therefore have $n(H_1)$ contained in $n(H_{1+i})$ for each i , and hence $n(H_0)$ contained in $n(H_1)$ for each i .

To complete the proof we need only show that if z is in $n(H_k)$ then z is in $n(H_{k-1})$. Assume z is in $n(H_k)$ then

$$\begin{aligned} \langle z|H_k|z\rangle &= \langle z|H_{k-1}|z\rangle + \frac{\langle z|\sigma_{k-1}\rangle^2}{\langle\sigma_{k-1}|y_{k-1}\rangle} \\ &\quad - \frac{\langle y_{k-1}|H_{k-1}|z\rangle^2}{\langle y_{k-1}|H_{k-1}|y_{k-1}\rangle}. \end{aligned}$$

Let $|p\rangle = (H_{k-1})^{1/2}|z\rangle$ and $|q\rangle = (H_{k-1})^{1/2}|y_{k-1}\rangle \neq 0$ then

$$\begin{aligned} (4.6) \quad 0 &= \langle z|H_k|z\rangle \\ &= \frac{\langle p|p\rangle\langle q|q\rangle - \langle p|q\rangle^2}{\langle q|q\rangle} + \frac{\langle z|\sigma_{k-1}\rangle^2}{\langle y_{k-1}|\sigma_{k-1}\rangle}. \end{aligned}$$

By Schwartz's Inequality we have

$$\langle p|p\rangle\langle q|q\rangle - \langle p|q\rangle^2 \geq 0.$$

Thus, for the right hand side of equation (4.6) to be zero we must have

$$\langle p|p\rangle\langle q|q\rangle - \langle p|q\rangle^2 = 0$$

and

$$\langle z|\sigma_{k-1}\rangle = 0.$$

Therefore,

$$\begin{aligned} 0 = H_k|z\rangle &= H_{k-1}|z\rangle - \frac{H_{k-1}|y_{k-1}\rangle\langle y_{k-1}|H_{k-1}|z\rangle}{\langle y_{k-1}|H_{k-1}|y_{k-1}\rangle} \\ &= H_{k-1}|z - t_{k-1}y_{k-1}\rangle \end{aligned}$$

where

$$t_{k-1} = \frac{\langle y_{k-1}|H_{k-1}|z\rangle}{\langle y_{k-1}|H_{k-1}|y_{k-1}\rangle}.$$

We therefore have $z - t_{k-1}y_{k-1}$ in $n(H_{k-1})$. However, we have just shown that if a vector is in $n(H_{k-1})$ then it is in $n(H_k)$. Hence $|z - t_{k-1}y_{k-1}\rangle$ is in $n(H_k)$ and

$$\begin{aligned} 0 = H_k|z - t_{k-1}y_{k-1}\rangle &= -t_{k-1}H_k|y_{k-1}\rangle \\ &= -t_{k-1}|\sigma_{k-1}\rangle, \text{ where } |\sigma_{k-1}\rangle \neq 0. \end{aligned}$$

This implies $t_{k-1} = 0$ and z is in $n(H_{k-1})$. Therefore $n(H_i)$ is contained in $n(H_0)$ for each i and the proof is complete.

Lemma 4.4 if H_0 is positive definite then H_i is positive

definite for each i.

Proof. The proof of this follows directly from (4.4d) of lemma 4.1, and lemma 4.3.

We complete this section with two corollaries to theorem 4.2

Corollary 1 if H_0 is positive definite and $\langle z | g_0 \rangle = 0$ for all z in $n(N)$ then algorithm (4.3) will terminate in r steps, $r \leq \text{rank of } N$, if and only if $|g_r\rangle = 0$.

Proof. If $\langle z | g_0 \rangle = 0$ for all z in $n(N)$ then $\langle z | g_i \rangle = 0$ for all $i \leq r$.

It is only necessary to show that case ii) and case iii) of theorem 4.2 cannot occur. Clearly case ii) cannot occur since H_{r-1} is positive definite by lemma 4.4.

In case iii) we have $H_{r-1}g_r \neq 0$ and $N|s_r\rangle = 0$ with

$$|s_r\rangle = H_r|g_r\rangle = H_{r-1}|g_r\rangle + B_{r-1}|g_r\rangle.$$

Since $|s_r\rangle$ is in $n(N)$ and $\langle z | g_i \rangle = 0$ for all z in $n(N)$, then

$$0 = \langle g_r | s_r \rangle = \langle g_r | H_{r-1} | g_r \rangle - \frac{\langle y_{r-1} | H_{r-1} | g_r \rangle^2}{\langle y_{r-1} | H_{r-1} | y_{r-1} \rangle}.$$

In order for the right hand side of this equation to be zero it is necessary that

$$|g_r\rangle = t|y_{r-1}\rangle$$

for some scalar t . If $t \neq 0$ then

$$|s_r\rangle = H_r |g_r\rangle = t H_r |y_{r-1}\rangle = t |\sigma_{r-1}\rangle$$

and $N|\sigma_{r-1}\rangle = 0$. This contradicts our assumption that the algorithm terminates at the r^{th} step. If $t = 0$ then $|g_r\rangle = 0$ and $H_{r-1}|g_r\rangle = 0$, which contradicts $H_{r-1}|g_r\rangle \neq 0$. Therefore case iii) cannot occur and the algorithm terminates if and only if $|g_r\rangle = 0$.

Corollary 2 if both H_0 and N are positive definite then algorithm (4.3) will terminate in r -steps, $r \leq \text{rank of } N$, if and only if $|g_r\rangle = 0$.

5. The equivalence of algorithms (4.1) and (4.3).

In this section we prove the equivalence of algorithms (4.1) and (4.3). We do this by showing that if $H_0 = K$ then the two algorithms generate vectors that are scalar multiples of each other. G. E. Myers proved a special case of this in [5]. There H_0 and K were both equal to I and N was positive definite.

In order to prove this equivalence we need the following two lemmas.

Lemma 5.1 $\langle g_j | H_{i-1} | g_i \rangle = 0$ for $i < j \leq r$

Proof. Since

$$|s_i\rangle = H_i |g_i\rangle = H_{i-1} |g_i\rangle - \frac{H_{i-1} |y_{i-1}\rangle \langle y_{i-1} | H_{i-1} | g_i \rangle}{\langle y_{i-1} | H_{i-1} | y_{i-1} \rangle},$$

and

$$\langle s_i | g_j \rangle = 0 \quad i < j \leq r$$

by 4.5 c) , we have

$$(5.1) \quad 0 = \langle g_j | s_i \rangle = \langle g_j | H_{i-1} | g_i \rangle \\ - \frac{\langle g_j | H_{i-1} | y_{i-1} \rangle \langle y_{i-1} | H_{i-1} | g_i \rangle}{\langle y_{i-1} | H_{i-1} | y_{i-1} \rangle} .$$

We note that

$$\langle g_j | H_{i-1} | y_{i-1} \rangle = - \langle g_j | H_{i-1} | g_i \rangle$$

and

$$\langle y_{i-1} | H_{i-1} | y_{i-1} \rangle = \langle g_{i-1} | H_{i-1} | g_{i-1} \rangle + \langle g_i | H_{i-1} | g_i \rangle .$$

Therefore (5.1) can be written as

$$(5.2) \quad 0 = \langle g_j | H_{i-1} | g_i \rangle \left[1 - \frac{\langle g_i | H_{i-1} | g_i \rangle}{\langle g_{i-1} | H_{i-1} | g_{i-1} \rangle + \langle g_i | H_{i-1} | g_i \rangle} \right] .$$

Also

$$\langle g_{i-1} | H_{i-1} | g_{i-1} \rangle > 0$$

since $i - 1 < r$. Therefore equation (5.2) is zero if and only if

$$\langle g_j | H_{i-1} | g_i \rangle = 0 \text{ for } i < j \leq r .$$

Lemma 5.2 $H_i | g_j \rangle = H_0 | g_j \rangle$ for $i < j \leq r$.

Proof. We first note that (4.5c) implies

$$A_i | g_j \rangle = 0 \text{ for } i = 0, \dots, j-1, \text{ hence}$$

$$\begin{aligned}
H_i |g_j\rangle &= H_{i-1} |g_j\rangle - \frac{H_{i-1} |y_{i-1}\rangle \langle y_{i-1} | H_{i-1} |g_j\rangle}{\langle y_{i-1} | H_{i-1} | y_{i-1}\rangle} \\
&= H_{i-1} |g_j\rangle - H_{i-1} |y_{i-1}\rangle \left[\frac{\langle g_{i-1} | H_{i-1} |g_j\rangle - \langle g_i | H_{i-1} |g_j\rangle}{\langle y_{i-1} | H_{i-1} | y_{i-1}\rangle} \right].
\end{aligned}$$

However,

$$\langle g_i | H_{i-1} |g_j\rangle = 0$$

by the previous lemma, and

$$\langle g_{i-1} | H_{i-1} |g_j\rangle = \langle s_{i-1} |g_j\rangle = 0.$$

Therefore

$$H_i |g_j\rangle = H_{i-1} |g_j\rangle \text{ for } i < j \leq r$$

and

$$H_i |g_j\rangle = H_{i-1} |g_j\rangle = \cdots = H_0 |g_j\rangle$$

which proves the lemma.

Theorem 5.1 If $K = H_0$ then algorithms (4.1) and (4.3) generate vectors in the same directions.

Proof. The proof of this theorem is by induction. We have

$$|s_0\rangle = H_0 |g_0\rangle = K |g_0\rangle = |p_0\rangle.$$

Assume $|s_i\rangle = t_i |p_i\rangle$ with $t_i > 0$ and $i \leq k < r$ where

$|p_i\rangle$, $i = 0, \dots, k$, are generated by algorithm (4.1)

and $|s_i\rangle$, $i = 0, \dots, k$, are generated by algorithm (4.3).

We have

$$(5.3) \quad |s_{k+1}\rangle = H_{k+1}|g_{k+1}\rangle = H_k|g_{k+1}\rangle + B_k|g_{k+1}\rangle \\ = H_0|g_{k+1}\rangle - \frac{H_k|y_k\rangle\langle y_k|H_k|g_{k+1}\rangle}{\langle g_k|H_k|g_k\rangle + \langle g_{k+1}|H_k|g_{k+1}\rangle}.$$

Now

$$(5.4) \quad H_k|y_k\rangle = H_k|g_k\rangle - H_k|g_{k+1}\rangle = |s_k\rangle - H_0|g_{k+1}\rangle,$$

and

$$(5.5) \quad \langle y_k|H_k|g_{k+1}\rangle = \langle y_k|H_0|g_{k+1}\rangle = -\langle g_{k+1}|H_0|g_{k+1}\rangle.$$

Substituting (5.4) and (5.5) into (5.3) and collecting terms we obtain

$$(5.6) \quad |s_{k+1}\rangle = t_{k+1} H_0|g_{k+1}\rangle + \bar{b}_k |s_k\rangle$$

where

$$t_{k+1} = \frac{\langle g_k|H_k|g_k\rangle}{\langle g_k|H_k|g_k\rangle + \langle g_{k+1}|H_0|g_{k+1}\rangle} > 0$$

and

$$\bar{b}_k = \frac{\langle g_{k+1}|H_0|g_{k+1}\rangle}{\langle g_k|H_k|g_k\rangle} = \frac{1}{t_k} \frac{\langle g_{k+1}|K|g_{k+1}\rangle}{\langle g_k|K|g_k\rangle}.$$

Using (4.1) and (4.2) we can show

$$b_k = \frac{\langle g_{k+1}|K|g_{k+1}\rangle}{\langle g_k|K|g_k\rangle},$$

hence $\bar{b}_k = 1/t_k b_k$.

Therefore (5.6) becomes

$$|s_{k+1}\rangle = t_{k+1} K |g_{k+1}\rangle + b_k |p_k\rangle = t_{k+1} |p_{k+1}\rangle.$$

Corollary 1 If $H_0 = K$ is positive definite and $\langle z | g_0 \rangle = 0$ for all z in $n(N)$ then algorithms (4.1) and (4.3) are equivalent in the sense described above.

Lemma 5.3 $\langle g_i | H_k | g_j \rangle = 0$ for $i < j$ and $k \leq i$.

Proof. For $k = i$ we have

$$\langle g_i | H_i | g_j \rangle = \langle s_i | g_j \rangle = 0 \text{ for } i < j.$$

For $k < i$ we have

$$\langle g_i | H_k | g_j \rangle = \langle g_i | H_{i-1} | g_j \rangle = 0,$$

with lemma 5.2 implying the first equality and lemma 5.1 the second.

We note that in particular

$$\langle g_i | H_0 | g_j \rangle = 0 \text{ for } i \neq j.$$

6. Converse to theorem 4.1 and theorem 4.2.

Algorithms (4.1) and (4.3) can be used to generate any set of N -conjugate vectors where N is a positive semi definite matrix. In [2] this is shown for algorithm (4.1) when N and K are positive definite matrices.

Theorem 6.1 Let N be a positive semi definite matrix with rank m and let p_0, \dots, p_{m-1} be m vectors such that

$$(6.1) \quad d_i = (p_i, N p_i) > 0, \quad (p_i, N p_j) = 0, \quad i \neq j.$$

Let c_0, \dots, c_{m-1} be m positive real numbers and set
 $a_i = c_i/d_i \quad i = 0, \dots, m-1$ and

$$(6.2) \quad g_0 = a_0 N p_0 + \dots + a_{m-1} N p_{m-1}.$$

Let g_1, \dots, g_{m-1} be generated by

$$(6.3) \quad g_{i+1} = g_i - a_i N p_i.$$

Then there exists a positive semi definite matrix K such that

$$(6.4) \quad p_0 = K g_0, \quad p_{i+1} = K g_{i+1} + b_i p_i$$

with $b_i = c_{i+1}/c_i$. Also

$$(6.5) \quad c_i = (g_i, K g_i), \quad (g_i, K g_j) = 0, \quad i \neq j.$$

Proof. Let G and P be the $n \times n$ matrices whose columns are $g_0, \dots, g_{m-1}, 0, \dots, 0$ and $p_0, \dots, p_{m-1}, 0, \dots, 0$ respectively. Let A, C, D be diagonal matrices whose diagonals are $a_0, \dots, a_{m-1}, 0, \dots, 0$; $c_0, \dots, c_{m-1}, 0, \dots, 0$; and $d_0, \dots, d_{m-1}, 0, \dots, 0$. Let $\bar{A}, \bar{C}, \bar{D}$ be diagonal matrices whose diagonals are $1/a_0, \dots, 1/a_{m-1}, 0, \dots, 0$; $1/c_0, \dots, 1/c_{m-1}, 0, \dots, 0$; and $1/d_0, \dots, 1/d_{m-1}, 0, \dots, 0$.

Set $b_i = c_{i+1}/c_i$, and let B be the matrix with $b_0, \dots, b_{m-2}, 0, \dots, 0$ just above the diagonal and zeros elsewhere. Let $V = (v_{ij})$ with $v_{ij} = 1$ when $i = j+1$ and $v_{ij} = 0$ elsewhere, and set $T = I - V$.

We have

$$A = C\bar{D} = \bar{D}C, B = \bar{C}V^*C, P^*NP = D.$$

Relation (6.3) has the form

$$GT = NPA.$$

It follows that

$$P^*GT = P^*NPA = DA = C.$$

Since $C^* = C$, we have

$$C = C^* = (P^*GT)^* = T^*G^*P$$

and

$$G^*P = T^{*-1}C.$$

If we define $Q = P\bar{C}T^*$ then

$$G^*Q = G^*P\bar{C}T^* = T^{*-1}C\bar{C}T^* = T^{*-1}\bar{I}T^* = \bar{I}$$

where

$$\bar{I} = \begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix}$$

with I an $m \times m$ identity matrix. Define $K = QCQ^*$ then

$$KG = QCQ^*G = QC\bar{I}^* = QC\bar{I} = QC.$$

Since

$$P\bar{C}T^*C = QC = KG$$

and

$$P\bar{C}T^*C = P(I-B) = P-PB,$$

we have

$$KG = P-PB.$$

Hence

$$P = KG+PB$$

establishing (6.4). Also

$$G*KG = G*QCQ*G = C$$

and K is positive semi definite. This completes the proof of the theorem.

We can also choose K positive definite. Let \bar{K} be a positive semi definite matrix whose null space is generated by g_0, \dots, g_{m-1} , then K defined by $K = K + \bar{K}$ is a positive definite matrix. The matrix K also satisfies conditions (6.4) and (6.5) of theorem 6.1.

Theorem 6.1 is an extension of theorem 5.1 in reference [2], from N and K positive definite to N and K positive semi definite.

Theorem 6.2 Let N be a positive semi definite matrix of rank m and p_0, \dots, p_{m-1} m vectors such that

$$(p_i, Np_i) > 0, (p_i, Np_j) = 0 \quad i \neq j.$$

If (6.2) and (6.3) hold then there exists a positive semi definite matrix H_0 such that

$$(6.6) \quad p_0 = H_0 g_0, \quad p_{i+1} = H_{i+1} g_{i+1}$$

where H_{i+1} is defined as in algorithm (4.3). Also

$$(6.7) \quad (g_i, H_k g_i) > 0, \quad (g_i, H_k g_j) = 0 \quad k \leq i < j.$$

Proof. Statement (6.6) follows from the equivalence of algorithms (4.1) and (4.3) and theorem 6.1. We get (6.7) from lemma 5.3 and the fact that H_k is positive semi definite.

We now describe the conjugate gradient method (cg-method) and the Davidon method as formulated by Fletcher and Powell (FP-method).

7. The cg-method.

The following is essentially the method described in [1] and [2]. However here we are allowing A to be a singular matrix. The method of conjugate gradients used to minimize the function $E(x)$, where $E(x) = (r, Rr)$ with R a positive definite matrix and $r = k - Ax$, is as follows. Select a positive definite matrix K and an initial estimate x_0 and compute $r_0 = k - Ax_0$, $g_0 = A^* R r_0$, and $p_0 = K g_0$. The vector g_0 is a negative scalar multiple of the usual gradient of $E(x)$ at $x = x_0$, and is therefore in a direction of decrease of $E(x)$. Having obtained x_i , r_i , g_i and p_i we compute x_{i+1} , r_{i+1} , g_{i+1} , and p_{i+1} by the following. We have

$$a_i = \frac{(g_i, p_i)}{(p_i, N p_i)} = \frac{(g_i, K g_i)}{(p_i, N p_i)}$$

with $N = A^* R A$

$$\begin{aligned}
 (7.1) \quad & x_{i+1} = x_i + a_i p_i \\
 & r_{i+1} = k - Ax_{i+1} = r_i - a_i A p_i \\
 & g_{i+1} = A^* R r_{i+1} \\
 & b_i = - \frac{(N p_i, K g_{i+1})}{(p_i, N p_i)} = - \frac{(g_{i+1}, K g_{i+1})}{(g_i, K g_i)} \\
 & p_{i+1} = K g_{i+1} + b_i p_i.
 \end{aligned}$$

We note that

$$g_{i+1} = A^* R r_{i+1} = g_i - a_i N p_i.$$

The vectors g_0, g_1, \dots are orthogonal to $n(N)$ and hence algorithm (7.1) satisfies the hypothesis of corollary 1 to theorem 4.1. Therefore algorithm (7.1) terminates in r steps, $r \leq \text{rank of } N$, when $g_r = 0$. By lemma 2.5 x_r minimizes $E(x)$ and is therefore a solution to $Ax = k$ if one exists.

Theorem 7.1 The cg-method is a cd-method and the solution x_r is in M . The points x_0, \dots, x_i ($i \leq r$) generated by the cg-method lie in an i -dimensional subspace π_i . The point x_i minimizes the function $E(x)$ in π_i .

This theorem is proved in [2], however it is shown above that x_r is in M .

The following theorem completes this section on the cg-method. It essentially shows that every cd-method is a cg-method.

Theorem 7.2 Let N be a positive semi definite matrix such that m is the rank of N and let p_0, \dots, p_{m-1} be m vectors such that

$$d_i = (p_i, Np_i) > 0, \quad (p_i, Np_j) = 0 \quad i \neq j.$$

Then there exists a cg-method such that the algorithm described in section 7 generates p_0, \dots, p_{m-1} .

Proof. To prove this we need only show that g_0 lies in the space spanned by Np_0, \dots, Np_{m-1} and then apply theorem 6.1 and the paragraph following it. If $m = n$ then Np_0, \dots, Np_{m-1} spans E^n and we are done. If $m < n$, then E^n is spanned by $Np_0, \dots, Np_{m-1}, q_m, \dots, q_{n-1}$ where q_m, \dots, q_{n-1} spans the null space of N . Thus,

$$g_0 = \alpha_0 Np_0 + \dots + \alpha_{m-1} Np_{m-1} + v$$

where v is in $n(N)$. Hence

$$0 = (v, g_0) = (v, v)$$

and therefore $v = 0$. This means

$$g_0 = \alpha_0 Np_0 + \dots + \alpha_{m-1} Np_{m-1}$$

where

$$\alpha_i = \frac{(g_0, p_i)}{(p_i, Np_i)} = a_i.$$

The material in this section for the case when A is nonsingular can be found in [2].

8. The FP-method.

We consider here the application of the FP-method to a quadratic function, in particular to the error function $E(x)$.

The method is as follows.

Select an initial point x_0 and compute

$$|r_0\rangle = |k\rangle - A|x_0\rangle, \quad |g_0\rangle = A^*R|r_0\rangle,$$

and $|s_0\rangle = H_0|g_0\rangle$, where H_0 is a given positive definite matrix. Having obtained H_i , x_i , r_i , g_i , and s_i we compute H_{i+1} , a_i , x_{i+1} , r_{i+1} , g_{i+1} and s_{i+1} by the following formulas. We have

$$a_i = \frac{\langle g_i | s_i \rangle}{\langle s_i | N | s_i \rangle}$$

$$x_{i+1} = x_i + a_i |s_i\rangle$$

$$|r_{i+1}\rangle = |r_i\rangle - a_i A |s_i\rangle$$

$$(8.1) \quad |g_{i+1}\rangle = A^*R|r_{i+1}\rangle = |g_i\rangle - a_i N |s_i\rangle$$

$$H_{i+1} = H_i + A_i + B_i$$

$$A_i = \frac{|\sigma_i\rangle \langle \sigma_i|}{\langle \sigma_i | y_i \rangle}$$

$$B_i = \frac{-H_i |y_i\rangle \langle y_i| H_i}{\langle y_i | H_i | y_i \rangle}$$

$$|\sigma_i\rangle = a_i |s_i\rangle, \quad |y_i\rangle = |g_i\rangle - |g_{i+1}\rangle = N |\sigma_i\rangle$$

$$|s_{i+1}\rangle = H_{i+1} |g_{i+1}\rangle.$$

This algorithm is equivalent to algorithm (4.3) where g_0, \dots, g_r are the negative gradient vectors of $E(x)$ at x_0, \dots, x_r respectively. As in the last section the gradient vectors are orthogonal to the null space of N , and hence this algorithm satisfies the conditions of corollary 1

to theorem 4.2. Therefore the algorithm terminates if and only if $|g_r\rangle = 0$. This implies that x_r minimizes $E(x)$.

In view of corollary 1 to theorem 5.1 the cg-method and the FP-method are equivalent. This means that if H_0 and K are equal and the two methods start at the same initial point x_0 then they generate direction vectors that are scalar multiples of each other and hence generate the same points x_1, \dots, x_r .

Theorem 8.1 Theorem 7.1 holds with cg-method replaced by FP-method.

Theorem 8.2 Every cd-method is an FP-method.

Proof. This follows from the equivalence of the cg and FP-methods and theorem 7.2.

From theorem 8.1 we have that $\sigma_0, \dots, \sigma_{r-1}$ spans an i -dimensional subspace π_i and the point x_i minimizes $E(x)$ in π_i . If we let W be the space spanned by $\sigma_1, \dots, \sigma_{r-1}$ then the matrix H_i projects the gradient vector g_i into W in the direction of σ_i . This becomes the next direction of search for a minimum to $E(x)$. We note that W is N orthogonal to π_i . If N is positive definite then $E^n = \pi_i + W$ where $W = \pi_i^N$.

Lemma 8.1 if N is positive semi definite then H_r is a left inverse to N on the space generated by $\sigma_0, \dots, \sigma_{r-1}$. If N is positive definite and $r = n$ then

$$\begin{aligned} \text{i)} \quad H_n &= N^{-1} \\ \text{ii)} \quad N^{-1} &= \sum_{i=0}^{n-1} A_i \end{aligned}$$

Proof. We have

$$(8.2) \quad H_r N |\sigma_j\rangle = |\sigma_j\rangle, \quad j = 0, \dots, r-1.$$

Hence H_r is a left inverse to N on the space generated by $\sigma_0, \dots, \sigma_{r-1}$. If N is positive definite and $r = n$ then (8.2) implies $H_n = N^{-1}$.

We observe that $S'NS = \Lambda$ where S is the matrix whose columns are $\sigma_0, \dots, \sigma_{n-1}$ and S' is the transpose of S and Λ is diagonal with $\langle \sigma_i | N | \sigma_i \rangle$, $i = 0, \dots, n-1$, along the diagonal. The matrix N satisfies the equation

$$N = S'^{-1} \Lambda S^{-1} = (S \Lambda^{-1} S')^{-1},$$

hence

$$\begin{aligned} N^{-1} &= S \Lambda^{-1} S' = \sum_i (\Lambda^{-1})_{ii} |\sigma_i\rangle \langle \sigma_i| \\ &= \sum_i \langle \sigma_i | N | \sigma_i \rangle^{-1} |\sigma_i\rangle \langle \sigma_i| = \sum_i A_i. \end{aligned}$$

This completes the proof of the lemma and also completes section 8.

The last two parts of lemma 8.1 can be found in [3].

9. Second method for constructing vectors that are N-conjugate.

In section 4 we mentioned another method for generating N -conjugate vectors. This is the usual Gram-Schmidt method. The connection between the cd-method and Gauss elimination will be examined in this section. This is considered in [1] but with the restriction that the matrix we are dealing with be positive definite. Here we will relax this restriction to

include positive semi definite.

Let N be a positive semi definite matrix and u_1, \dots, u_n be n linear independent vectors. We generate p_1, \dots, p_n by the following algorithm

$$(9.1) \quad \begin{aligned} p_1 &= u_1 \\ p_{i+1} &= u_{i+1} - \sum_{j=1}^i t_{ij} p_j \end{aligned}$$

where

$$t_{ij} = \begin{cases} \frac{(p_j, Nu_{i+1})}{(p_i, Np_j)} & p_j \text{ not in } n(N) \\ 0 & p_j \text{ in } n(N). \end{cases}$$

If p_j is in $n(N)$ then t_{ij} can be arbitrary for each i , however we pick $t_{ij} = 0$ for convenience. Some elementary properties are stated in the following lemma.

Lemma 9.1 If p_0, \dots, p_n are generated by (9.1) then the following hold.

- a) The vectors p_1, \dots, p_n are linear independent.
- b) If u_i is in $n(N)$ then $p_i = u_i$.
- c) If Γ is the set of indices for which p_r is in $n(N)$ then i, j not in Γ implies $(p_i, Np_j) = 0$ $i \neq j$ and $(p_i, Np_i) > 0$.
- d) The set of vectors $\{p_r | r \text{ in } \Gamma\}$ form a basis for $n(N)$.
- e) The set of vectors $\{p_i | i \text{ not in } \Gamma\}$ are N -linear independent.
- f) $(p_i, Nu_j) = 0$ for $j < i$ and $(p_i, Nu_i) = (p_i, Np_i)$.

Proof. To prove a) we note that (9.1) implies u_1, \dots, u_n can be written as a linear combination of p_1, \dots, p_n . Therefore p_1, \dots, p_n span E^n and are linear independent. If u_i is in $n(N)$ then by (9.1) $t_{i-1j} = 0$ for $j = 1, \dots, i-1$ and $p_i = u_i$, establishing b). We note that c) follows directly from (9.1) and the definition of t_{ij} . For d) we observe that if v is in $n(N)$ then

$$v = \sum_{i=1}^n a_i p_i$$

and

$$0 = Nv = \sum_{i \notin \Gamma} a_i Np_i,$$

hence

$$0 = a_j (p_j, Np_j), \quad j \notin \Gamma$$

implying $a_j = 0$ for j not in Γ . Therefore

$$v = \sum_{i \in \Gamma} a_i p_i.$$

We get e) directly from c).

The vectors u_j can be written as a linear combination of p_1, \dots, p_j where $(p_i, Np_j) = 0$ for $i \neq j$. This implies that $(p_i, Nu_j) = 0$ for $j < i$ and

$$(p_i, Nu_i) = (p_i, N(p_i + v))$$

where v is a linear combination of p_1, \dots, p_{i-1} . Therefore

$$(p_i, Nu_i) = (p_i, Np_i),$$

completing the proof of the lemma.

This method of orthogonalizing vectors is just the standard Gram-Schmidt method.

We have a second method of computing the vectors p_1, \dots, p_n from u_1, \dots, u_n . This is found in [1] and it is through this method we show the relationship between Gauss elimination method for solving a linear system and the cd-method.

Let p_1, \dots, p_n be generated by the following algorithm.

$$(9.2) \quad \begin{aligned} u_i^1 &= u_i & i &= 1, \dots, n \\ p_j &= u_j^j \\ u_i^{j+1} &= u_i^j - t_{ij} p_j & i &= j+1, \dots, n \\ t_{ij} &= \begin{cases} \frac{(u_i^j, Nu_j)}{(p_j, Nu_j)} & , \quad i > j, p_j \text{ not in } n(N) \\ 0 & p_j \text{ in } n(N) \end{cases} \end{aligned}$$

The following lemma indicates some of the properties concerning this algorithm

Lemma 9.2 If p_1, \dots, p_n are generated by (9.2) then the conditions in lemma 9.1 hold along with the following additions

$$\begin{aligned} g) \quad (u_i^k, Nu_j) &= 0 & j < k \\ h) \quad (u_i^k, Np_j) &= 0 & j < k \\ i) \quad u_i^k &= u_i - t_{i1}p_1 - \dots - t_{i,k-1}p_{k-1} \end{aligned}$$

We now look at the connection between Gauss elimination

method and algorithm (9.2). Consider the linear system $Nx = k$ where N is a positive semi definite matrix. The following shows that solving this linear system by Gauss elimination is equivalent to solving it by the cd-method where the direction vectors are found by the N -orthogonalization of the vectors e_1, \dots, e_n where

$$e_i = (0, 0, \dots, 0, 1, 0, \dots, 0)$$

with 1 in the i^{th} position. This is done in [1] for the case where N is positive definite.

Iteration (3.3), with initial point taken as zero, becomes

$$(9.3) \quad \begin{aligned} x_1 &= 0 \\ x_{i+1} &= x_i + a_i p_i \\ a_i &= \begin{cases} \frac{(g_i, p_i)}{(p_i, Np_i)} & p_i \text{ not in } n(N) \\ 0 & p_i \text{ in } n(N) \end{cases} \\ g_i &= k - Nx_i \end{aligned}$$

and with

$$g_1 = k - Nx_1 = k.$$

Lemma 3.1 states that for p_i not in $n(N)$

$$a_i = \frac{(g_i, p_i)}{(p_i, Np_i)} = \frac{(g_i, p_i)}{(p_i, Np_i)} = \frac{(k, p_i)}{(p_i, Np_i)}.$$

The vectors p_1, \dots, p_n are generated by the recursion formula

$$\begin{aligned}
 (9.4) \quad & u_1^1 = e_1 \\
 & p_1 = u_1^1 \\
 & u_i^{j+1} = u_i^j - t_{ij} p_j \quad i = j+1, \dots, n \\
 & t_{ij} = \begin{cases} \frac{(Nu_i^j, e_j)}{(Np_j, e_j)} & p_j \text{ not in } n(N) \\ 0 & p_j \text{ in } n(N). \end{cases}
 \end{aligned}$$

These formulas generate mutually conjugate vectors p_1, \dots, p_n and by (9.3) corresponding points x_1, \dots, x_n . The vectors p_1, \dots, p_n satisfy lemma 9.2. The inner products in t_{ij} and a_i can be computed in an easy way. The method is the one used in elimination. We also get a basis for the null space of the matrix N in the process.

First we write down the matrices N , I and the vector k .

$$\begin{aligned}
 (9.5) \quad & \begin{matrix} r_{11} & r_{12} & \dots & r_{1n} & 1 & 0 & \dots & 0 & k_1 \\ r_{21} & r_{22} & \dots & r_{2n} & 0 & 1 & 0 & \dots & 0 & k_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ r_{n1} & r_{n2} & \dots & r_{nn} & 0 & \dots & 0 & 1 & k_n \end{matrix}
 \end{aligned}$$

where N is an $n \times n$ matrix, I is the identity matrix and k is an n vector. We will assume the equation $Nx = k$ has a solution. By (9.3) and (9.4) we have

$$\begin{aligned}
 p_1 &= u_1^1 = e_1 \\
 x_2 &= x_1 + a_1 p_1 = a_1 p_1
 \end{aligned}$$

$$a_i = \begin{cases} \frac{(k, p_1)}{(Np_1, e_1)} & p_1 \text{ not in } n(N) \\ 0 & p_1 \text{ in } n(N). \end{cases}$$

We also have

$$u_i^2 = u_i^1 - t_{i1}p_1 = e_i - t_{i1}e_1$$

where

$$t_{i1} = \begin{cases} \frac{(Nu_i^1, e_1)}{(Np_1, e_1)} = \frac{r_{i1}}{r_{11}} & p_1 \text{ not in } n(N) \\ 0 & p_1 \text{ in } n(N). \end{cases}$$

Multiplying the first row by t_{i1} and subtracting from the i^{th} row ($i=2, \dots, n$), we obtain the new matrix

$$(9.6) \quad \begin{array}{cccccccc} r_{11} & r_{12} & \cdot & \cdot & \cdot & r_{1n} & p_{11} & \cdot & \cdot & \cdot & p_{1n} & k \\ 0 & r_{22}^2 & \cdot & \cdot & \cdot & r_{2n}^2 & p_{21} & \cdot & \cdot & \cdot & p_{2n} & k_2^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & u_{31}^2 & \cdot & \cdot & \cdot & u_{3n}^2 & k_3^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & r_{n2}^2 & \cdot & \cdot & \cdot & r_{nn}^2 & u_{n1}^2 & \cdot & \cdot & \cdot & u_{nn}^2 & k_n^2 \end{array}$$

where

$$r_{ij}^2 = r_{ij} - t_{i1}r_{1j} \quad \begin{array}{l} i = 2, \dots, n \\ j = 1, \dots, n \end{array}$$

with

$$p_2 = u_2^2 \text{ and } k_1^2 = k_1 - t_{11}k_1.$$

We also have

$$p_1 = (p_{11}, \dots, p_{1n})$$

$$p_2 = (p_{21}, \dots, p_{2n})$$

$$u_i^2 = (u_{i1}^2, \dots, u_{in}^2) \quad i = 3, \dots, n.$$

The following two lemmas summarize the results.

Lemma 9.3 If $p_1 = e_1$ is not in $n(N)$ then the following hold.

$$a) \quad r_{ij}^2 = (Nu_i^2, e_j) \quad i = 2, \dots, n$$

$$j = 1, \dots, n$$

$$b) \quad k_i^2 = (k, u_i^2) \quad i = 2, \dots, n$$

$$c) \quad r_{ij}^2 = r_{ji}^2 \quad i, j = 2, \dots, n$$

$$d) \quad g_2 = (0, k_2^2, \dots, k_n^2)$$

$$e) \quad Np_2 = (0, r_{22}^2, \dots, r_{2n}^2)$$

Proof. a) $r_{ij}^2 = r_{ij} - t_{i1}r_{1j}$

$$= (Ne_j, e_i) - t_{i1}(Ne_j, e_1) = (Ne_i, e_j) - t_{i1}(Ne_i, e_j)$$

$$= (N(e_i - t_{i1}e_1), e_j) = (Nu_i^2, e_j).$$

In particular we have

$$r_{22}^2 = (Np_2, e_2), \quad r_{i2}^2 = (Nu_i^2, e_2).$$

$$b) \quad k_i^2 = k_i - t_{i1}k_1 = (k, e_i) - t_{i1}(k, e_1)$$

$$= (k, e_i - t_{i1}e_1) = (k, u_i^2)$$

We observe here that $k_2^2 = (k, p_2)$.

$$c) \quad \text{We have } r_{ij} = r_{ji} \text{ since } N = N^* \text{ and also}$$

$$t_{11}r_{1j} = \frac{r_{11}}{r_{11}} r_{1j} = \frac{r_{j1}}{r_{11}} r_{11} = t_{j1}r_{11} ,$$

hence

$$r_{1j}^2 = r_{1j} - t_{11}r_{1j} = r_{j1} - t_{j1}r_{11} = r_{j1}^2$$

for $i, j = 2, \dots, n$.

$$\begin{aligned} \text{d) } g_2^1 &= (g_2, e_1) = (g_1, e_1) - a_1(Np_1, e_1) \\ &= k_1 - \frac{k_1}{r_{11}} r_{11} = k_1 - t_{11}k_1 = k_1^2 , \quad i = 2, \dots, n \end{aligned}$$

with

$$g_2^1 = k_1 - \frac{k_1}{r_{11}} r_{11} = 0 .$$

$$\text{e) } \text{If } r_{1j}^2 = (Nu_1^2, e_j) \text{ then}$$

$$r_{2j}^2 = (Np_2, e_j) \quad j = 1, \dots, n.$$

Therefore

$$Np_2 = (0, r_{22}^2, \dots, r_{2n}^2)$$

and

$$r_{21}^2 = (Np_2, p_1) = 0 .$$

Lemma 9.4 If $p_1 = e_1$ is in $n(N)$ then $r_{11} = r_{11} = 0$ for
 $i = 1, \dots, r$ and the five conditions in lemma 9.3 reduce to

- a) $r_{ij}^2 = r_{ji}^2$, $i = 2, \dots, n$
 $j = 1, \dots, n$
- b) $k_i^2 = k_i$, $i = 2, \dots, n$ and $k_1 = 0$
- c) $r_{ij}^2 = r_{ji}^2$, $i, j = 2, \dots, n$.

By a) this reduces to $r_{ij} = r_{ji}$.

- d) $g_2 = (0, k_2, \dots, k_n)$
- e) $Np_2 = (0, r_{22}, \dots, r_{2n})$

Proof. We have $r_{ii} = r_{ii} = 0$ for $i = 1, \dots, n$ since $Np_1 = Ne_1 = 0$. Conditions a) through e) are a direct consequence of $t_{ii} = a_i = 0$. In b) $k_1 = 0$ since we have assumed $Nx = k$ has a solution. This ends the proof of lemma 9.4.

In the case p_1 is in $n(N)$ we have

$$x_2 = x_1 + a_1 p_1 = 0.$$

We will consider another step before looking at the general case. From algorithms (9.3) and (9.4) we obtain

$$\begin{aligned}
 x_3 &= x_2 + a_2 p_2 \\
 a_2 &= \begin{cases} \frac{(k, p_2)}{(Np_2, e_2)} & p_2 \text{ not in } n(N) \\ 0 & p_2 \text{ in } n(N) \end{cases} \\
 u_i^3 &= u_i^2 - t_{i2} p_2 \quad i = 3, \dots, n
 \end{aligned}$$

where

$$t_{12} = \begin{cases} \frac{(Nu_1^2, e_2)}{(Np_2, e_2)} & p_2 \text{ not in } n(N) \\ 0 & p_2 \text{ in } n(N). \end{cases}$$

If p_2 is not in $n(N)$ then by lemma 9.3

$$t_{12} = r_{12}^2 / r_{22}^2, \quad a_2 = k_2^2 / r_{22}^2.$$

Multiplying the second row of (9.6) by t_{12} and subtracting from the i^{th} row ($i=3, \dots, n$), we obtain the new matrix

$$(9.7) \quad \begin{array}{cccccccc} r_{11} & r_{12} & r_{13} & \cdot & \cdot & \cdot & r_{1n} & p_{11} & \cdot & \cdot & \cdot & p_{1n} & k_1 \\ 0 & r_{22}^2 & r_{23}^2 & \cdot & \cdot & \cdot & r_{2n}^2 & p_{21} & \cdot & \cdot & \cdot & p_{2n} & k_2^2 \\ 0 & 0 & r_{33}^3 & \cdot & \cdot & \cdot & r_{3n}^3 & p_{31} & \cdot & \cdot & \cdot & p_{3n} & k_3^3 \\ 0 & 0 & r_{43}^3 & \cdot & \cdot & \cdot & r_{4n}^3 & u_{41}^3 & \cdot & \cdot & \cdot & u_{4n}^3 & k_4^3 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & r_{n3}^3 & \cdot & \cdot & \cdot & r_{nn}^3 & u_{n1}^3 & \cdot & \cdot & \cdot & u_{nn}^3 & k_n^3 \end{array}.$$

We have

$$\begin{aligned} r_{ij}^3 &= r_{ij}^2 - t_{12} r_{2j}^2 & i &= 3, \dots, n \\ & & j &= 2, \dots, n \\ u_i^3 &= u_i^2 - t_{12} p_i & i &= 3, \dots, n \end{aligned}$$

with

$$p_3 = u_3^3 \text{ and } k_1^3 = k_1^2 - t_{12} k_2^2$$

where $p_3 = (p_{31}, \dots, p_{3n})$ and $u_1^3 = (u_{11}^3, \dots, u_{1n}^3)$.

The following two lemmas summarize the results.

Lemma 9.5 If p_2 is not in $n(N)$ then the following hold

- a) $r_{ij}^3 = (Nu_i^3, e_j)$ $i = 3, \dots, n$
 $j = 2, \dots, n$
b) $k_i^3 = (k, u_i^3)$ $i = 3, \dots, n$
c) $r_{ij}^3 = r_{ji}^3$ $i, j = 3, \dots, n$
d) $g_3 = (0, 0, k_3^3, \dots, k_n^3)$
e) $Np_3 = (0, 0, r_{33}^3, \dots, r_{3n}^3)$.

Lemma 9.6 If p_2 is in $n(N)$ then $r_{12}^2 = r_{21}^2 = 0$ for
 $i = 2, \dots, n$ and the five conditions in lemma 9.5 reduce to

- a) $r_{ij}^3 = r_{ij}^2$ $i = 3, \dots, n$
 $j = 2, \dots, n$
b) $k_i^3 = k_i^2$ $i = 3, \dots, n$ and $k_2^2 = 0$
c) $r_{ij}^3 = r_{ji}^3$ $i, j = 3, \dots, n$.

By a) this reduces to $r_{ij}^2 = r_{ji}^2$.

- d) $g_3 = (0, 0, k_3^2, \dots, k_n^2)$
e) $Np_3 = (0, 0, r_{33}^2, \dots, r_{3n}^2)$

and also $x_3 = x_2$.

If p_2 is in $n(N)$, then by lemma 9.6, the matrix (9.7) would look like

$$(9.8) \quad \begin{array}{cccccccc} r_{11} & r_{12} & r_{13} & \dots & r_{1n} & p_{11} & \dots & p_{1n} & k_1 \\ 0 & 0 & \dots & \dots & 0 & p_{21} & \dots & p_{2n} & 0 \\ 0 & 0 & r_{33}^2 & \dots & r_{3n}^2 & u_{31}^2 & \dots & u_{3n}^2 & k_3^2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & r_{n3}^2 & \dots & r_{nn}^2 & u_{n1}^2 & \dots & u_{nn}^2 & k_n^2. \end{array}$$

In this case we have

$$p_3 = u_3^3 = u_3^2 = (u_{31}^2, \dots, u_{3n}^2).$$

It should be noted that if we had not assumed the existence of a solution to $Nx = k$ then k_2^2 may not have been zero in the case of p_2 in $n(N)$.

We now consider the general case. Suppose we have p_1, \dots, p_m and x_1, \dots, x_m then by algorithms (9.3) and (9.4) we have

$$\begin{aligned} x_{m+1} &= x_m + a_m p_m \\ a_m &= \begin{cases} \frac{(k, p_m)}{(Np_m, e_m)} & p_m \text{ not in } n(N) \\ 0 & p_m \text{ in } n(N) \end{cases} \\ u_i^{m+1} &= u_i^m - t_{im} p_m \quad i = m+1, \dots, n \end{aligned}$$

where

$$t_{im} = \begin{cases} \frac{(Nu_i^m, e_m)}{(Np_m, e_m)} & p_m \text{ not in } n(N) \\ 0 & p_m \text{ in } n(N). \end{cases}$$

If p_m is not in $n(N)$ then we have

$$t_{im} = r_{im}^m / r_{mm}^m \quad \text{and} \quad a_m = km^m / r_{mm}^m.$$

The matrix is in the form

$$\begin{array}{cccccccccccc}
 r_{11} & r_{12} & \cdot & \cdot & \cdot & r_{1m} & \cdot & \cdot & \cdot & r_{1n} & p_{11} & \cdot & \cdot & \cdot & p_{1n} & k_1 \\
 0 & r_{22}^2 & \cdot & \cdot & \cdot & r_{2m}^2 & \cdot & \cdot & \cdot & r_{2n}^2 & p_{21} & \cdot & \cdot & \cdot & p_{2n} & k_2^2 \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 0 & 0 & \cdot & \cdot & 0 & r_{mm}^m & \cdot & \cdot & \cdot & r_{mn}^m & p_{m1} & \cdot & \cdot & \cdot & p_{mn} & k_m^m \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 0 & 0 & \cdot & \cdot & 0 & r_{nm}^m & \cdot & \cdot & \cdot & r_{nn}^m & u_{n1}^m & \cdot & \cdot & \cdot & u_{nn}^m & k_n^m.
 \end{array}
 \tag{9.9}$$

Multiplying the m^{th} row of (9.9) by t_{im} and subtracting from the i^{th} row ($i = m+1, \dots, n$), we obtain the new matrix

(9.10)

$$\begin{array}{cccccccccccc}
 r_{11} & r_{12} & \cdot & \cdot & \cdot & r_{1m} & \cdot & \cdot & \cdot & r_{1n} & p_{11} & \cdot & \cdot & \cdot & p_{1n} & k_1 \\
 0 & r_{22}^2 & \cdot & \cdot & \cdot & r_{2m}^2 & \cdot & \cdot & \cdot & r_{2n}^2 & p_{21} & \cdot & \cdot & \cdot & p_{2n} & k_2^2 \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 0 & 0 & \cdot & \cdot & 0 & r_{mm}^m & \cdot & \cdot & \cdot & r_{mn}^m & p_{m1} & \cdot & \cdot & \cdot & p_{mn} & k_m^m \\
 0 & 0 & \cdot & \cdot & 0 & 0 & r_{m+1}^{m+1} & \cdot & \cdot & r_{m+1}^{m+1} & p_{m+11} & \cdot & \cdot & \cdot & p_{m+1n} & k_{m+1}^{m+1} \\
 \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
 0 & 0 & \cdot & \cdot & \cdot & \cdot & r_{n+1}^{m+1} & \cdot & \cdot & r_{nn}^{m+1} & u_{n1}^{m+1} & \cdot & \cdot & \cdot & u_{nn}^{m+1} & k_n^{m+1}.
 \end{array}$$

We have

$$\begin{aligned}
 r_{ij}^{m+1} &= r_{ij}^m - t_{im} r_{mj}^m & i &= m+1, \dots, n \\
 & & j &= m, \dots, n \\
 u_i^{m+1} &= u_i^m - t_{im} p_m & i &= m+1, \dots, n \\
 k_i^{m+1} &= k_i^m - t_{im} k_m^m & i &= m+1, \dots, n
 \end{aligned}$$

and

$$p_{m+1} = (p_{m+11}, \dots, p_{m+1n})$$

$$u_i^{m+1} = (u_{i1}^{m+1}, \dots, u_{in}^{m+1}) \quad i = m+1, \dots, n.$$

The following two lemmas summarize the results.

Lemma 9.7 If p_m is not in $n(N)$ then the following hold

- a) $r_{ij}^{m+1} = (Nu_i^{m+1}, e_j) \quad i = m+1, \dots, n$
 $j = m, \dots, n$
- b) $k_i^{m+1} = (k, u_i^{m+1}) \quad i = m+1, \dots, n$
- c) $r_{ij}^{m+1} = r_{ji}^{m+1} \quad i, j = m+1, \dots, n$
- d) $g_{m+1} = (0, \dots, 0, k_{m+1}^{m+1}, \dots, k_n^{m+1})$
- e) $Np_{m+1} = (0, \dots, 0, r_{m+1m+1}^{m+1}, \dots, r_{m+1n}^{m+1})$.

Lemma 9.8 If p_m is in $n(N)$ then $r_{im}^m = r_{mi}^m$
 $= 0$ ($i = m, \dots, n$) and the five conditions in lemma 9.7
reduce to

- a) $r_{ij}^{m+1} = r_{ij}^m \quad i, j = m+1, \dots, n$
- b) $k_i^{m+1} = k_i^m \quad i = m+1, \dots, n$
- c) $r_{ij}^{m+1} = r_{ji}^{m+1} \quad i, j = m+1, \dots, n$.

By a) this reduces to $r_{ij}^m = r_{ji}^m$.

- d) $g_{m+1} = (0, \dots, 0, k_{m+1}^m, \dots, k_n^m)$
- e) $Np_{m+1} = (0, \dots, 0, r_{m+1m+1}^m, \dots, r_{m+1n}^m)$

and $x_{m+1} = x_m$.

In this case we also have

$$p_{m+1} = u_{m+1}^{m+1} = u_{m+1}^m,$$

and again $k_m^m = 0$ since we are assuming that $Nx = k$ has a solution.

Recall that applying the cd-method to minimize the error function $E(x)$ is equivalent to solving the linear system $Nx = k$ if $E(x)$ is suitably chosen. Let $E(x) = (r, Rr)$ where R is a positive definite matrix and $r = \bar{k} - Ax$. The gradient vector

$$g = A^*Rr = k - Nx$$

where

$$k = A^*R\bar{k}, \quad N = A^*RA.$$

Since $E(x)$ is minimized when $g(x) = 0$ then we see that minimizing $E(x)$ is equivalent to solving the linear system $Nx = k$.

From the work in section 9 we see that in choosing our direction vectors p_1, \dots, p_n by N-orthogonalizing e_1, \dots, e_n the inner products in algorithms 9.3 and 9.4 are found by performing Gauss elimination on the system $Nx = k$. Hence the application of Gauss elimination to $Nx = k$ and solving $Nx = k$ through algorithms (9.3) and (9.4) are equivalent.

Let N be positive definite in the previous material of section 9. Since every cd-method is a cg-method then there exists a positive definite matrix K such that

$$\begin{aligned} g_1 &= k \\ (9.11) \quad p_1 &= Kg_1 \quad g_{i+1} = g_i - a_i Np_i \\ a_i &= \frac{(g_i, p_i)}{(p_i, Np_i)} = \frac{(k, p_i)}{(p_i, Np_i)} \end{aligned}$$

$$p_{i+1} = Kg_{i+1} + b_i p_i$$

$$b_i = c_{i+1}/c_i, \quad c_i = (g_i, p_i) = (k, p_i).$$

The vectors g_2, \dots, g_n are generated by (9.11) since

$$g_{i+1} = k - Nx_{i+1} = k - Nx_i - a_i Np_i = g_i - a_i Np_i.$$

From the proof of theorem 6.1 we see that if we choose

$$K = G^*{}^{-1}CG^{-1},$$

where G is the matrix whose columns are g_1, \dots, g_n and C is the diagonal matrix with c_i down the diagonal, then (9.11) is satisfied. Hence the vectors p_1, \dots, p_n generated by N -orthogonalization of e_1, \dots, e_n are also generated by (9.11), and solving $Nx = k$ by Gauss elimination is equivalent to solving it by the cg-method whose direction vectors are given by (9.11).

We now apply the cg-method and FP-method to the minimization of non-quadratic functions.

APPLICATIONS TO NON-QUADRATIC FUNCTIONS

10. Algorithms for non-quadratic functions.

The algorithms considered in this section and also sections 11 and 12 are for minimizing a non-quadratic function $f(x)$, bounded from below for all x and having a positive definite Hessian matrix at the minimum point. The minimum point found is of course just a local minimum. These algorithms differ only in the method for generating the direction vectors p_0, p_1, \dots . These vectors are the directions in which the function $f(x)$ is minimized. The algorithms presented in sections 10, 11, and 12 will not be given in terms of $f(x)$ but in a form that is convenient for comparing and deriving convergence results. In section 13 the relationship between the minimization of the function $f(x)$ and the algorithms in sections 10, 11, and 12 will be shown.

Let $\{C_i\}$ and $\{T_i\}$ be sequences of positive definite matrices converging to C and T respectively, where C and T are positive definite.

Theorem 10.1 If g_0 is a non-zero vector then the algorithm

$$(10.1) \quad \begin{aligned} p_0 &= T_0 g_0 & g_{i+1} &= g_i - a_i C_i p_i \\ a_i &= \frac{(g_i, p_i)}{(p_i, C_i p_i)} \end{aligned}$$

$$p_{i+1} = T_{i+1} g_{i+1}$$

generates non-zero vectors g_0, g_1, \dots and p_0, p_1, \dots such that

$$(10.2) \quad (g_{i+1}, p_i) = 0.$$

The algorithm terminates in r steps if for some r, $g_r = 0$;
otherwise

$$(10.3) \quad \lim_{i \rightarrow \infty} g_i = 0.$$

Proof. The scalar a_i is chosen so that (10.2) holds. If $g_r = 0$ for some r then $p_r = 0$ and the algorithm clearly terminates. The proof of (10.3) is a result of the following lemma and theorem.

Lemma 10.1 There exists positive real numbers m and M
such that

$$(10.4) \quad 0 < m \leq 1/a_i \leq M$$

for all i.

Proof. From (10.1) we have

$$a_i = \frac{(g_i, T_i^{-1} g_i)}{(p_i, C_i p_i)}.$$

The eigenvalues of $C_i T_i$ are zeros of the determinates of the matrices $C_i^{-1} \lambda - T_i$ and $T_i^{-1} \lambda - C_i$. Therefore $1/a_i$ is between the maximum and minimum eigenvalues of $C_i T_i$. Since C_i and T_i are positive definite then the matrix $C_i T_i$ has real

positive eigenvalues. Thus

$$0 < \lambda_{\min}^i \leq 1/a_i \leq \lambda_{\max}^i$$

where λ_{\max}^i and λ_{\min}^i are respectively the maximum and minimum eigenvalues of $C_i T_i$. The minimum and maximum eigenvalues of $C_i T_i$ converge respectively to the minimum and maximum eigenvalues of CT . Since the eigenvalues of CT are real and positive there exists m and M such that

$$0 < m \leq \lambda_{\min}^i \leq 1/a_i \leq \lambda_{\max}^i \leq M$$

for all i .

Theorem 10.2 If algorithm (10.1) does not terminate in a finite number of steps there exists positive real numbers v and w such that $0 \leq v < 1$ and for $i > w$

$$(10.5) \quad (g_{i+1}, C^{-1} g_{i+1}) \leq v (g_i, C^{-1} g_i).$$

Proof. We have

$$(10.6) \quad (g_{i+1}, C^{-1} g_{i+1}) = (g_i, C^{-1} g_i) - 2a_i (g_i, C^{-1} \varepsilon_i T_i g_i) + a_i^2 (g_i, T_i C_i C^{-1} C_i T_i g_i).$$

This equation can be written in the form

$$(10.7) \quad (g_{i+1}, C^{-1} g_{i+1}) = (g_i, C^{-1} g_i) - 2a_i (g_i, T_i g_i) G_i(g_i)$$

where

$$G_i(x) = \frac{(x, C^{-1} C_i T_i x)}{(x, T_i x)} - 1/2 \frac{(x, T_i C_i C^{-1} C_i T_i x)}{(x, T_i C_i T_i x)}.$$

Since C_i converges to C ,

$$\lim_{i \rightarrow \infty} G_i(x) = 1/2$$

for all $x \neq 0$. Hence there exists $w > 0$ such that for all $i > w$

$$(10.8) \quad G_i(x) > 1/4.$$

By lemma 10.1 $a_i \geq 1/M$. Hence from equation (10.7) we obtain

$$(g_{i+1}, C^{-1}g_{i+1}) \leq \left[1 - \frac{a_i (g_i, T_i g_i)}{2 (g_i, C^{-1}g_i)} \right] (g_i, C^{-1}g_i)$$

where

$$1 - \frac{a_i (g_i, T_i g_i)}{2 (g_i, C^{-1}g_i)} \leq 1 - m/2M$$

with $0 < m \leq M$. By choosing

$$v = 1 - m/2M,$$

we have $0 \leq v < 1$ and

$$(g_{i+1}, C^{-1}g_{i+1}) \leq v (g_i, C^{-1}g_i)$$

for all $i > w$.

As a consequence of (10.5) we have

$$\lim_{i \rightarrow \infty} (g_i, C^{-1}g_i) = 0,$$

and hence

$$\lim_{i \rightarrow \infty} g_i = 0.$$

Therefore the proof of theorem 10.2 completes the proof of theorem 10.1.

Theorem 10.2 shows that for $i > w$ the sequence $\{(g_i, C^{-1}g_i)\}$ converges as fast as a geometric progression with ratio v .

Before considering a special case of theorem 10.1 we finish this section with the following lemma and theorem.

Lemma 10.2 If $D_i = I - a_i C_i T_i$ then λ is an eigenvalue of D_i if and only if $(1-\lambda)/a_i$ is an eigenvalue of $C_i T_i$.

Proof. Since

$$D_i - \lambda I = a_i [I (1-\lambda)/a_i - C_i T_i]$$

and a_i is bounded away from zero for all i , then λ is a zero of the determinate of $D_i - \lambda I$ if and only if $(1-\lambda)/a_i$ is a zero of the determinate of $I (1-\lambda)/a_i - C_i T_i$.

Theorem 10.3 If in algorithm (10.1) p_i is the eigenvector corresponding to the maximum eigenvalue of $C_i T_i$ then

$$|g_{i+1}| < |g_i|$$

for all i .

Proof. Let λ_{\max}^i and λ_{\min}^i be respectively the maximum and minimum eigenvalues of D_i . By lemma 10.2 $(1-\lambda_{\min}^i)/a_i$ and $(1-\lambda_{\max}^i)/a_i$ are respectively the maximum and minimum eigenvalues of $C_i T_i$. We note that $\lambda_{\max}^i < 1$ since $(1-\lambda_{\max}^i)/a_i > 0$. Since

$$a_1 = \frac{(g_1, p_1)}{(p_1, C_1 p_1)} = \frac{(p_1, T_1^{-1} p_1)}{(p_1, C_1 p_1)} = \frac{1}{\text{max. eigenvalue of } C_1 T_1}$$

then $1/a_1$ is the maximum eigenvalue of $C_1 T_1$. This means $\lambda_{\min}^1 = 0$ and all eigenvalues of D_1 lie in the interval $[0, 1)$. Therefore, by (10.1),

$$g_{1+1} = g_1 - a_1 C_1 T_1 g_1 = D_1 g_1$$

and

$$|g_{1+1}| \leq \|D_1\| |g_1| < |g_1|.$$

11. The FP-algorithm for non-quadratic functions.

The FP-algorithm is a special case of algorithm (10.1). However, before looking at it we first consider the following result.

If $\{T_1\}$ is a sequence of positive definite matrices such that

$$\lim_{i \rightarrow \infty} T_i = uC^{-1}$$

where u is a positive real number, then algorithm (10.1) converges.

With the above choice for the sequence $\{T_1\}$ we get a convergence which is stated in the following theorem.

Theorem 11.1 If T_1 is defined as above and algorithm (10.1) does not terminate in a finite number of steps then g_0, g_1, \dots satisfy

$$(11.1) \quad |g_{1+1}| \leq v_1 |g_1|$$

where $v_i \geq 0$ and

(11.2)

$$\lim_{i \rightarrow \infty} v_i = 0$$

Proof. Let

$$L_i(x) = \frac{(x, T_i^{-1}x)}{(x, C_i x)}$$

then

$$\lim_{i \rightarrow \infty} L_i(x) = 1/u$$

for all $x \neq 0$. Therefore

$$\lim_{i \rightarrow \infty} a_i = \lim_{i \rightarrow \infty} L_i(p_i) = 1/u$$

and

$$\lim_{i \rightarrow \infty} C_i T_i = uI.$$

If we choose $v_i = \|D_i\|$ then

$$\lim_{i \rightarrow \infty} v_i = \lim_{i \rightarrow \infty} \|I - a_i C_i T_i\| = 0.$$

Since $g_{i+1} = D_i g_i$, we have

$$|g_{i+1}| \leq \|D_i\| |g_i| = v_i |g_i|.$$

We now give the FP-algorithm for minimizing a non-quadratic function.

Theorem 11.2 The conclusion of theorem 10.1 follows if we replace $\{T_i\}$ by $\{H_i\}$ where H_0 is a positive definite matrix and

$$\begin{aligned}
 H_{i+1} &= H_i + A_i + B_i \\
 A_i &= \frac{|\sigma_i\rangle\langle\sigma_i|}{\langle\sigma_i|\gamma_i\rangle} \\
 B_i &= \frac{-H_i|\gamma_i\rangle\langle\gamma_i|H_i}{\langle\gamma_i|H_i|\gamma_i\rangle} \\
 |\sigma_i\rangle &= a_i|p_i\rangle, \quad |\gamma_i\rangle = c_i|\sigma_i\rangle.
 \end{aligned}
 \tag{11.3}$$

Proof. To prove this theorem it is only necessary to show that H_i is positive definite for each i and converges to a positive definite matrix.

The proof is by induction. The matrix H_0 is positive definite by hypothesis. We assume H_1, \dots, H_k are positive definite and show that H_{k+1} is positive definite. We have

$$\langle x|H_{k+1}|x\rangle = \langle x|H_k|x\rangle + \frac{\langle x|\sigma_k\rangle^2}{\langle\sigma_k|\gamma_k\rangle} - \frac{\langle x|H_k|\gamma_k\rangle^2}{\langle\sigma_k|\gamma_k\rangle}.$$

If $|p\rangle = (H_k)^{1/2}|x\rangle$ and $|q\rangle = (H_k)^{1/2}|\gamma_k\rangle$ with $|q\rangle \neq 0$, then

$$\langle x|H_k|x\rangle = \frac{\langle p|p\rangle\langle q|q\rangle - \langle p|q\rangle^2}{\langle q|q\rangle} + \frac{\langle x|\sigma_k\rangle^2}{\langle\sigma_k|\gamma_k\rangle}
 \tag{11.4}$$

with

$$\langle\sigma_k|\gamma_k\rangle = \langle\sigma_k|c_k|\sigma_k\rangle > 0.
 \tag{11.5}$$

By Schwartz's inequality we have

$$(11.6) \quad \langle p|p\rangle \langle q|q\rangle - \langle p|q\rangle^2 \geq 0,$$

and is equal to zero if and only if

$$(11.7) \quad |p\rangle = t|q\rangle$$

for some non-zero t . Since H_k is positive definite, (11.7) is equivalent to

$$|x\rangle = t|y_k\rangle.$$

If $\langle x|\sigma_k\rangle = 0$ for $|x\rangle \neq 0$, then $|x\rangle \neq t|y_k\rangle$ by (11.5). Hence (11.6) is a strict inequality and $\langle x|H_{k+1}|x\rangle > 0$. If $\langle x|\sigma_k\rangle \neq 0$ then $\langle x|\sigma_k\rangle^2 > 0$ and $\langle x|H_{k+1}|x\rangle > 0$.

From the definition of H_{l+1} we have

$$H_{l+1}C_l|\sigma_l\rangle = |\sigma_l\rangle$$

and

$$\langle p_{l+1}|C_l|p_l\rangle = 0$$

for each l . The definition of H_l here and in section 4 differ in that here $y_l = C_l\sigma_l$ and in section 4 $y_l = N\sigma_l$. Since C_l converges to C we have

$$\lim_{l \rightarrow \infty} H_l = C^{-1}.$$

Therefore $\{H_l\}$ is a sequence of positive definite matrices whose limit is positive definite.

The next theorem gives a result on the convergence of the FP-algorithm.

Theorem 11.3 If T_l is equal to H_l and algorithm (10.1)
does not terminate in a finite number of steps then there
exists positive real numbers v_l such that

$$(11.9) \quad |g_{l+1}| \leq v_l |g_l|$$

and

$$(11.10) \quad \lim_{l \rightarrow \infty} v_l = 0.$$

Proof. With the choice of $T_l = H_l$ for each l , the hypothesis of theorem 11.1 is satisfied. Therefore both (11.9) and (11.10) hold.

12. The cg-algorithm for non-quadratic functions.

In this section two cg-algorithms are given for non-quadratic functions. Both methods, however, generate the same direction vectors when the function to be minimized is quadratic.

We again have the sequence $\{C_l\}$ of positive definite matrices converging to the positive definite matrix C .

Theorem 12.1 Given a positive definite matrix K and non-zero vector g_0 then the algorithm

$$p_0 = Kg_0 \quad g_{l+1} = g_l - a_l C_l p_l$$

$$a_l = \frac{(g_l, p_l)}{(p_l, C_l p_l)}$$

(12.1)

$$p_{l+1} = Kg_{l+1} + b_l p_l$$

$$b_i = - \frac{(Kg_{i+1}, C_i p_i)}{(p_i, C_i p_i)}$$

generates non-zero vectors g_0, g_1, \dots and p_0, p_1, \dots such that

$$(12.2) \quad (g_{i+1}, p_i) = 0, \quad (p_{i+1}, C_i p_i) = 0$$

for all i . The algorithm terminates in r steps if for some $r, g_r = 0$; otherwise

$$(12.3) \quad \lim_{i \rightarrow \infty} g_i = 0$$

Proof. The scalars a_i and b_i are chosen so that (12.2) holds. If for some $r, g_r = 0$ then $b_{r-1} = 0$ and $p_r = 0$ and the algorithm terminates. The proof of (12.3) follows from the next lemma and theorem.

Lemma 12.1 There exists positive real numbers m and M such that

$$(12.4) \quad 0 < m \leq 1/a_i \leq M$$

for all i .

Proof. For non-zero vectors p_i and g_i we have by (12.1)

$$a_i = \frac{(p_i, g_i)}{(p_i, C_i p_i)} = \frac{(g_i, Kg_i)}{(p_i, C_i p_i)}.$$

We note that

$$(g_i, Kg_i) = (p_i, K^{-1} p_i) - b_{i-1}^2 (p_{i-1}, K^{-1} p_{i-1})$$

and hence

$$(g_l, Kg_l) \leq (p_l, K^{-1} p_l).$$

Therefore

$$a_l \leq \frac{(p_l, K^{-1} p_l)}{(p_l, C_l p_l)}$$

and a_l is bounded above by $1/\lambda_{\min}^l$ where λ_{\min}^l is the minimum eigenvalue of $C_l K$. Since C_l converges to C there exists $m > 0$ such that

$$(12.5) \quad a_l \leq 1/m$$

for all l .

To show a_l is bounded away from zero we note that

$$(12.6) \quad (p_l, C_l p_l) = (g_l, K C_l K g_l) + 2b_{l-1}(p_l, C_l p_{l-1}) - b_{l-1}^2 (p_{l-1}, C_l p_{l-1}).$$

Using (12.1) and (12.2) equation (12.6) becomes

$$(12.7) \quad (p_l, C_l p_l) = (g_l, K C_l K g_l) + b_{l-1} \frac{(g_l, K g_l)}{a_{l-1}} \left[\frac{(p_l, C_l C_{l-1}^{-1} g_{l-1})}{(p_l, g_{l-1})} - \frac{(p_l, C_l C_{l-1}^{-1} g_l)}{(p_l, g_l)} + \frac{(p_{l-1}, C_l p_{l-1})}{(p_{l-1}, C_{l-1} p_{l-1})} \right].$$

Since C_l converges to C the term in brackets converges to 1.

We note that

$$b_{i-1} = \frac{(g_i, Kg_i)}{(g_{i-1}, Kg_{i-1})} > 0,$$

hence for i sufficiently large equation (12.7) implies

$$(p_i, C_i p_i) \geq (g_i, KC_i Kg_i)$$

and

$$a_i \geq \frac{(g_i, Kg_i)}{(g_i, KCKg_i)}.$$

Thus, for i sufficiently large a_i is bounded below by $1/\lambda_{\max}^i$ where λ_{\max}^i is the maximum eigenvalue of $C_i K$. Hence there exists $M > 0$ such that

$$(12.8) \quad a_i \geq 1/M$$

for all i . From (12.5) and (12.8) we have

$$0 < m \leq 1/a_i \leq M$$

for all i .

Theorem 12.2 If algorithm (12.1) does not terminate
in a finite number of steps there exists positive real
numbers v and w such that $0 \leq v < 1$ and for $i > w$

$$(12.9) \quad (g_{i+1}, C^{-1} g_{i+1}) \leq v (g_i, C^{-1} g_i).$$

Proof. Using (12.1) we have

$$(12.10) \quad (g_{i+1}, C^{-1} g_{i+1}) = (g_i, C^{-1} g_i) - 2a_i (g_i, Kg_i) G_i(g_i, p_i)$$

where

$$G_i(x, y) = \frac{(x, C^{-1} C_i y)}{(x, y)} - \frac{1}{2} \frac{(y, C^{-1} C_i C_i y)}{(y, C_i y)}.$$

Since C_i converges to C ,

$$\lim_{i \rightarrow \infty} G_i(x, y) = 1/2$$

for all x and y such that

$$(x, y) = (x, Kx) \neq 0.$$

Hence there exists $w > 0$ such that for $i > w$

$$(12.11) \quad G_i(x, y) > 1/4.$$

From the lemma 12.1 we have

$$(12.12) \quad 0 < 1/M \leq a_i \leq 1/m.$$

Therefore from (12.10), (12.11), and (12.12)

$$(g_{i+1}, C^{-1}g_{i+1}) \leq (g_i, C^{-1}g_i) - \frac{a_i(g_i, Kg_i)}{2}$$

which becomes

$$(g_{i+1}, C^{-1}g_{i+1}) \leq \left[1 - \frac{a_i(g_i, Kg_i)}{2(g_i, C^{-1}g_i)} \right] (g_i, C^{-1}g_i)$$

with

$$1 - \frac{a_i(g_i, Kg_i)}{2(g_i, C^{-1}g_i)} \leq 1 - m/2M.$$

By choosing

$$v = 1 - m/2M,$$

we have $0 \leq v < 1$ and

$$(g_{i+1}, C^{-1}g_{i+1}) \leq v (g_i, C^{-1}g_i).$$

As a result of this theorem we have

$$\lim_{i \rightarrow \infty} (g_i, C^{-1}g_i) = 0$$

and hence

$$\lim_{i \rightarrow \infty} g_i = 0.$$

This completes the proof of theorem 12.1. Theorem 12.2 shows that for $l > w$ the sequence $\{(g_i, C^{-1}g_i)\}$ converges as fast as a geometric progression with ratio v .

The second cg-method is simply taking algorithm (12.1) and restarting it every n iterations. The algorithm is given in the following theorem.

Theorem 12.3 If K is a positive definite matrix and g_0 is a non-zero vector then the algorithm

$$p_0 = Kg_0 \quad g_{j+1+1} = g_{j+1} - a_{j+1}C_{j+1}p_{j+1}$$

$$a_{j+1} = \frac{(g_{j+1}, p_{j+1})}{(p_{j+1}, C_{j+1}p_{j+1})}$$

(12.13)

$$p_{j+1+1} = Kg_{j+1+1} + b_{j+1}p_{j+1} \quad i = 0, \dots, n-2$$

$$b_{j+1} = - \frac{(Kg_{j+1+1}, C_{j+1}p_{j+1})}{(p_{j+1}, C_{j+1}p_{j+1})}$$

$$p_{j+1+1} = Kg_{j+1+1}$$

$$i = n-1 ;$$

where $j = 0, n, 2n, 3n, \dots$ and for each j , $i = 0, \dots, n-1$; generates non-zero vectors g_0, g_1, \dots and p_0, p_1, \dots such that

$$(12.14) \quad (g_{j+i+1}, p_{j+i}) = 0$$

and

$$(12.15) \quad (p_{j+i+1}, C_{j+i} p_{j+i}) = 0 \quad i = 0, \dots, n-2.$$

The algorithm terminates in r steps if for some r , $g_r = 0$; otherwise

$$\lim_{i \rightarrow \infty} g_i = 0$$

Theorem 12.4 If algorithm (12.13) does not terminate in a finite number of steps there exists positive real numbers v and w such that $0 \leq v < 1$ and for $i > w$

$$(12.16) \quad (g_{i+1}, C^{-1} g_{i+1}) \leq v (g_i, C^{-1} g_i).$$

The proofs of theorems 12.3 and 12.4 duplicate the proofs of theorems 12.1 and 12.2.

Numerical tests [4] for minimizing non-quadratic functions have shown the second conjugate gradient method converges faster than the first method. The Fletcher-Powell formulation of the Davidon algorithm (F-P method) exhibits substantially faster convergence than either of the conjugate gradient methods.

13. Minimizing non-quadratic functions.

Let $f(x)$ be a non-quadratic function bounded below, and assume $f(x)$ has a positive definite Hessian matrix at every local minimum.

We indicate the matrix of second partial derivatives (Hessian matrix) at the point x by $f''(x)$. The gradient vector at x is indicated by $f'(x)$.

To minimize the function $f(x)$ we select an initial point x_0 and set $g_0 = -f'(x_0)$. Next select a vector p_0 that is in a direction of decent. This will be true if $(g_0, p_0) > 0$. The next point x_1 is found by minimizing $f(x)$ along the line

$$x(t) = x_0 + t p_0.$$

This point will be characterized by

$$(f'(x_1), p_0) = 0.$$

Having found x_1 , calculate g_1 by $g_1 = -f'(x_1)$ and select p_1 so that it is in a direction of decent. This will again be true if $(g_1, p_1) > 0$. The point x_{1+1} is found by minimizing $f(x)$ along the line

$$(13.1) \quad x(t) = x_1 + t p_1.$$

We again have $(f'(x_{1+1}), p_1) = 0$. The sequence $\{f(x_i)\}$ is a monotonic decreasing sequence of real numbers bounded below, and therefore converges. The sequence $\{x_i\}$ converges

to some point \bar{x} which is a local minimum to $f(x)$. Hence

$$(13.2) \quad \lim_{i \rightarrow \infty} x_i = \bar{x}$$

and

$$(13.3) \quad \lim_{i \rightarrow \infty} f(x_i) = f(\bar{x})$$

where $f(\bar{x}) \leq f(x)$ for all x in some neighborhood of \bar{x} . We are of course assuming $f(x)$ is sufficiently well behaved so that (13.2) and (13.3) hold.

We have

$$(13.4) \quad g_{i+1} = g_i - f''(x_i + \theta_i \sigma_i) \sigma_i$$

where $0 \leq \theta_i < 1$ and

$$\sigma_i = x_{i+1} - x_i = a_i p_i.$$

Here a_i is the scalar that minimizes $f(x)$ along the line given by (13.1). The

$$\lim_{i \rightarrow \infty} (x_i + \theta_i \sigma_i) = \bar{x}$$

since σ_i converges to zero. Hence

$$\lim_{i \rightarrow \infty} f''(x_i + \theta_i \sigma_i) = f''(\bar{x}).$$

Since $f''(\bar{x})$ is positive definite there exists $J > 0$ such that for all $i > J$ $f''(x_i + \theta_i \sigma_i)$ is positive definite. For $i = 0, \dots, J$ we have

$$(g_i - g_{i+1}, \sigma_i) = a_i (g_i, p_i) > 0$$

since $a_i > 0$ and $(g_i, p_i) > 0$. Therefore there exists a positive definite matrix C_i such that

$$(13.5) \quad g_i - g_{i+1} = C_i \sigma_i$$

for $i = 0, \dots, J$. The vectors g_0, g_1, \dots therefore satisfy

$$g_{i+1} = g_i - a_i C_i p_i$$

where C_i is given by (13.5) for $i = 1, \dots, J$ and

$$C_i = f''(x_i + \theta_i \sigma_i)$$

for $i = J+1, \dots$. Hence $\{C_i\}$ is a sequence of positive definite matrices whose limit is $f''(\bar{x})$ which is also positive definite.

Now a_i satisfies the equation

$$a_i = \frac{(g_i, p_i)}{(p_i, C_i p_i)}$$

since $(g_{i+1}, p_i) = 0$.

With the above choice for C_i the algorithm for minimizing $f(x)$ can be written in the following form.

Given an initial point x_0 and setting $g_0 = -f'(x_0)$ we choose p_0 so that $(g_0, p_0) > 0$. Having found x_1, g_1 , and p_1 we calculate x_{i+1}, g_{i+1} , and p_{i+1} by the following formulas

$$g_{i+1} = g_i - a_i C_i p_i$$

(13.6)

$$a_1 = \frac{(g_1, p_1)}{(p_1, C_1 p_1)}$$

$$x_{i+1} = x_i + a_i p_i$$

and p_{i+1} is chosen so that $(g_{i+1}, p_{i+1}) > 0$.

The algorithms in sections 10, 11, and 12 differ from algorithm (13.6) in that they give specific methods for generating the vectors p_0, p_1, \dots . Hence the study of algorithms of the form given in sections 10, 11, and 12 is equivalent to studying algorithms for minimizing a large class of non-quadratic functions.

BIBLIOGRAPHY

BIBLIOGRAPHY

- [1] Hestenes, M. R. and E. Stiefel, "Methods of Conjugate Gradients for Solving Linear Systems." Report 1659. National Bureau of Standards, 1952.
- [2] Hestenes, M. R. "The Conjugate Gradient Method for Solving Linear Systems." Proceedings of Symposia in Applied Mathematics. Vol. 6. New York, N. Y.: McGraw-Hill Book Co., 1966.
- [3] Fletcher, R. and M. J. D. Powell. "A Rapidly Convergent Decent Method for Minimization." Computer Journal. July 1963.
- [4] Kelley, H. J. and Geraldine E. Myers. "Conjugate Direction Methods for Parameter Optimization." Presented at the 18th Congress of International Astronautical Federation, Belgrade, Yugoslavia, September 1967.
- [5] Myers, G. E. "Properties of the Conjugate Gradient and Davidson Methods." Westbury, New York: Analytic Mechanics Associates Inc.
- [6] Dirac, P. A. M. "The Principles of Quantum Mechanics." Oxford: Q. U. P., 1958.
- [7] Davidson, W. C. "Variable Metric Method for Minimization." A. E. C. Research and Development Report, ANL-5990, 1959.
- [8] Fletcher, R. and C. M. Reeves. "Function Minimization by Conjugate Gradients." Computer Journal. July 1964.
- [9] Hayes, R. M. "Iterative Methods of Solving Linear Problems in Hilbert Space." National Bureau of Standards Report 1733, May 1952.

MICHIGAN STATE UNIVERSITY LIBRARIES



3 1293 03145 4873