1. Conjugate Direction Methods

1.1. General Discussion. In this section we are again concerned with the problem of unconstrained optimization:

\[ \mathcal{P} : \text{ minimize } f(x) \]

subject to \( x \in \mathbb{R}^n \)

where \( f : \mathbb{R}^n \to \mathbb{R} \) is \( C^2 \). However, the emphasis will be on local quadratic approximations to \( f \). In particular, we study the problem \( \mathcal{P} \) when \( f \) has the form

\[ f(x) := \frac{1}{2} x^T Q x - b^T x, \]

where \( Q \) is a symmetric positive definite matrix. In this regard the notion of \( Q \)-conjugacy plays a key role.

Definition 1.1 (Conjugacy). Let \( Q \in \mathbb{R}^{n \times n} \) be symmetric and positive definite. We say that the vectors \( x, y \in \mathbb{R}^n \setminus \{0\} \) are \( Q \)-conjugate (or \( Q \)-orthogonal) if \( x^T Q y = 0 \).

Proposition 1.0.1 (Conjugacy implies Linear Independence). If \( Q \in \mathbb{R}^{n \times n} \) is positive definite and the set of nonzero vectors \( d_0, d_1, \ldots, d_k \) are (pairwise) \( Q \)-conjugate, then these vectors are linearly independent.

Proof. If \( 0 = \sum_{i=0}^{k} \alpha_i d_i \), then for \( i_0 \in \{0, 1, \ldots, k\} \)

\[ 0 = d_{i_0}^T Q \left[ \sum_{i=0}^{k} \alpha_i d_i \right] = \alpha_{i_0} d_{i_0}^T Q d_{i_0}, \]

Hence \( \alpha_i = 0 \) for each \( i = 0, \ldots, k \). \( \square \)

Observe that the unique solution to \( \mathcal{P} \) when \( f \) is given by (1.1) is

\[ x^* = Q^{-1} b. \]

If \( \{d_0, d_1, \ldots, d_{n-1}\} \) is a \( Q \)-conjugate basis for \( \mathbb{R}^n \), there are scalars \( \alpha_0, \ldots, \alpha_{n-1} \) such that

\[ x^* = \alpha_0 d_0 + \ldots + \alpha_{n-1} d_{n-1}. \]

Multiplying this expression through by \( Qd_i \) for each \( i = 0, \ldots, n-1 \) we find that

\[ \alpha_i = \frac{d_i^T Q x^*}{d_i^T Q d_i} = \frac{d_i^T b}{d_i^T Q d_i}, \]

for each \( i = 0, \ldots, n-1 \). Therefore

\[ x^* = \sum_{i=0}^{n-1} \frac{d_i^T b}{d_i^T Q d_i} = \left[ \sum_{i=0}^{n-1} \frac{d_i d_i^T}{d_i^T Q d_i} \right] b \]

so that

\[ Q^{-1} = \sum_{i=0}^{n-1} \frac{d_i d_i^T}{d_i^T Q d_i}. \]

It is important to note that the coefficients \( \alpha_i \) in the representation (1.2) can be computed without knowledge of \( x^* \). This observation is the basis of the following result.
Theorem 1.1 (Conjugate Direction Algorithm). Let \( \{d_i\}_{i=0}^{n-1} \) be a set of nonzero \( Q \)-conjugate vectors. For any \( x_0 \in \mathbb{R}^n \) the sequence \( \{x_k\} \) generated according to
\[
x_{k+1} := x_k + \alpha_k d_k, \quad k \geq 0
\]
with
\[
\alpha_k := \arg \min \{ f(x_k + \alpha d_k) : \alpha \in \mathbb{R} \}
\]
converges to the unique solution, \( x^* \) of \( P \) with \( f \) given by (1.1) after \( n \) steps, that is \( x_n = x^* \).

Proof. Let us first compute the value of the \( \alpha_k \)'s. Set
\[
\varphi_k(\alpha) = f(x_k + \alpha d_k) = \alpha d^T_k Qd_k + \alpha g_k^T d_k + f(x_k),
\]
where \( g_k = \nabla f(x_k) = Qx_k - b \). Then \( \varphi_k'(\alpha) = \alpha d^T_k Qd_k + g_k^T d_k \), hence
\[
\hat{\alpha}_k = -\frac{g_k^T d_k}{d_k^T Qd_k}.
\]
Now suppose \( x^* - x_0 \) has representation
\[
(1.3) \quad x^* - x_0 = \hat{\alpha}_0 d_0 + \hat{\alpha}_1 d_1 + \ldots + \hat{\alpha}_{n-1} d_{n-1}.
\]
Since \( x_n = x_0 + \alpha_0 d_0 + \ldots + \alpha_{n-1} d_{n-1} \), the result is established if we can show that \( \hat{\alpha}_k = \alpha_k \) for each \( k = 0, 1, \ldots, n-1 \). Multiplying (1.3) through by \( Qd_k \) yields
\[
(1.4) \quad \hat{\alpha}_k = \frac{d_k^T Q(x^* - x_0)}{d_k^T Qd_k}.
\]
But \( Qx^* = b \) and
\[
d_k^T Qx_0 = d_k^T Q(x_0 + \alpha_0 d_0 + \ldots + \alpha_{k-1} d_{k-1}) = d_k^T Qd_k.
\]
Therefore
\[
\hat{\alpha}_k = -\frac{d_k^T Q(x_0 - x^*)}{d_k^T Qd_k} = -\frac{d_k^T (Qx_k - b)}{d_k^T Qd_k} = -\frac{d_k^T g_k}{d_k^T Qd_k} = \alpha_k.
\]
\( \square \)

The following result provides further geometric insight into how the algorithm is proceeding.

Theorem 1.2. (Expanding Subspace Theorem)
Let \( \{d_i\}_{i=0}^{n-1} \) be a sequence of nonzero \( Q \)-conjugate vectors in \( \mathbb{R}^n \). Then for any \( x_0 \in \mathbb{R}^n \) the sequence \( \{x_k\} \) generated according to
\[
x_{k+1} = x_k + \alpha_k d_k
\]
with
\[
\alpha_k = -\frac{g_k^T d_k}{d_k^T Qd_k}
\]
has the property that \( f(x) = \frac{1}{2} x^T Qx - b^T x \) attains its minimum value on the affine set \( x_0 + \text{Span} \{d_0, \ldots, d_{k-1}\} \) at the point \( x_k \).
Proof. We establish the result by directly computing the solution to
\[
\min f(x) \\
\text{subject to } x - x_0 \in \text{Span } \{d_0, d_1, \ldots, d_{k-1}\}.
\]
By setting \( D_k = [d_0, d_1, \ldots, d_{k-1}] \) and \( z = x - x_0 \) we can rewrite (1.5) as
\[
\min f(z + x_0) \\
\text{subject to } z = D_k y,
\]
which can be written as
\[
\min_y f(D_k y + x_0).
\]
Writing \( \varphi(y) = f(D_k y + x_0) \)
\[
= \frac{1}{2} y^T D_k^T Q D_k Y + g_0^T D_k y + f(x_0),
\]
where \( g_0 = \nabla f(x_0) \), we see that the solution to (1.6) is obtained by setting
\[
0 = \nabla \varphi(y) = D_k^T Q D_k y + D_k^T g_0.
\]
Now
\[
D_k^T Q D_k = [d_i^T Q d_j]_{i,j=0}^{k-1}
\]
\[
= \text{diag}[d_i^T Q d_i]_{i=0}^{k-1},
\]
and
\[
D_k^T g_0 = [d_0^T g_0, d_1^T g_0, \ldots, d_{k-1}^T g_0]^T.
\]
Hence
\[
y_i = \frac{-d_i^T g_0}{d_i^T Q d_i} \text{ for } i = 0, \ldots, k - 1.
\]
Therefore, the solution (1.5) is
\[
x^*_k = x_0 + \sum_{i=0}^{k-1} \frac{-d_i^T g_0}{d_i^T Q d_i} d_i.
\]
Consequently, the result will be established if we can show that
\[
\frac{-d_i^T g_0}{d_i^T Q d_i} = \frac{-d_i^T g_i}{d_i^T Q d_i}
\]
for each \( i = 0, 1, \ldots, k - 1 \). But this follows immediately from (1.4) since
\[
\alpha_i = \frac{-d_i^T g_i}{d_i^T Q d_i} = \frac{d_i^T (Q x_i - b)}{d_i^T Q d_i}
\]
\[
= \frac{-d_i^T (Q(x_0 + \alpha_0 d_0 + \alpha_1 d_1 + \cdots + \alpha_{i-1} d_{i-1} - b))}{d_i^T Q d_i}
\]
\[
= \frac{-d_i^T g_0}{d_i^T Q d_i}
\]
where the global minimum value of \( f \) is attained at \( x^* \) and so satisfies \( Q x^* = b \). \qed
Corollary 1.2.1. [Subspace Orthogonality Condition]
In the method of Conjugate directions the gradients \( g_k, k = 0, 1, \ldots, n \) satisfy
\[
g_k^T d_i = 0 \quad \text{for } i < k.
\]

Proof. This follows from a general property of minimization on affine sets. Consider the problem
\[
\min \varphi(x)
\text{ subject to } x \in x_0 + S,
\]
where \( \varphi : \mathbb{R}^n \to \mathbb{R} \) is \( C^1 \) and \( S \) is the subspace \( S := \text{span} \{v_1, \ldots, v_k\} \). If \( V \) is the matrix whose columns are given by \( v_1, \ldots, v_k \), then this problem is equivalent to the problem
\[
\min \varphi(x_0 + Vz)
\text{ subject to } z \in \mathbb{R}^k.
\]
Setting \( \hat{\varphi}(z) = \varphi(x_0 + Vz) \), we get that if \( \bar{z} \) solves the latter problem, then \( V^T \nabla \varphi(x_0 + V\bar{z}) = \nabla \hat{\varphi}(\bar{z}) = 0 \). Setting \( \bar{x} = x_0 + V\bar{z} \), we conclude that \( \bar{x} \) solves the original problem if and only if \( \bar{z} \) solves the latter problem in which case \( V^T \nabla \varphi(\bar{x}) = 0 \), or equivalently, \( v_i^T \nabla \varphi(\bar{x}) = 0 \) for \( i = 1, 2, \ldots, k \). \( \square \)

1.2. The Conjugate Gradient Algorithm. The conjugate direction algorithm of the previous section appears to be seriously flawed in that one must have on hand a set of conjugate directions \( \{d_0, \ldots, d_{n-1}\} \) in order to apply it. However, one builds a set of Q-conjugate directions as the algorithm proceeds. The example of such a procedure studied in this section is called the conjugate gradient algorithm.

The C-G Algorithm:

Initialization: \( x_0 \in \mathbb{R}^n, d_0 = -g_0 = -\nabla f(x_0) = b - Qx_0 \).

For \( k = 0, 1, 2, \ldots \)
\[
\alpha_k := -g_k^T d_k/d_k^T Q d_k
\]
\[
x_{k+1} := x_k + \alpha_k d_k
\]
\[
g_{k+1} := Qx_{k+1} - b
\]
\[
\beta_k := g_{k+1}^T Q d_k/d_k^T Q d_k
\]
\[
d_{k+1} := -g_{k+1} + \beta_k d_k
\]
\[
k := k + 1.
\]

Theorem 1.3. [Conjugate Gradient Theorem]
The C-G algorithm is a conjugate direction method. If it does not terminate at \( x_k \), then
\[
(1) \text{ Span } [g_0, g_1, \ldots, g_k] = \text{span } [g_0, Qg_0, \ldots, Q^k g_0]
\]
\[
(2) \text{ Span } [d_0, d_1, \ldots, d_k] = \text{span } [g_0, Qg_0, \ldots, Q^k g_0]
\]
\[
(3) d_k^T Q d_i = 0 \text{ for } i \leq k - 1
\]
\[
(4) \alpha_k = g_k^T g_k/d_k^T Q d_k
\]
\[
(5) \beta_k = g_{k+1}^T g_{k+1}/g_k^T g_k.
\]

Proof. We first prove (1)-(3) by induction. The results are clearly true for \( k = 0 \). Now suppose they are true for \( k \), we show they are true for \( k + 1 \). First observe that
\[
g_{k+1} = g_k + \alpha_k Q d_k
\]
so that $g_{k+1} \in \text{Span}[g_0, \ldots, Q^{k+1}g_0]$ by the induction hypothesis on (1) and (2). Also $g_{k+1} \notin \text{Span} \left[ d_0, \ldots, d_k \right]$ otherwise $g_{k+1} = 0$ (by Theorem 1.2.1 since the method is a conjugate direction method up to step $k$ by the induction hypothesis. Hence $g_{k+1} \notin \text{Span} \left[ g_0, \ldots, Q^k g_0 \right]$ and so $\text{Span} \left[ g_0, g_1, \ldots, g_{k+1} \right] = \text{Span} \left[ g_0, \ldots, Q^{k+1}g_0 \right]$, which proves (1).

To prove (2) write

$$d_{k+1} = -g_{k+1} + \beta_k d_k$$

so that (2) follows from (1) and the induction hypothesis on (2).

To see (3) observe that

$$d^T_{k+1}Qd_i = -g^T_{k+1}Qd_i + \beta_k d^T_k Qd_i.$$

For $i = k$ the right hand side is zero by the definition of $\beta_k$. For $i < k$ both terms vanish. The term $g^T_{k+1}Qd_i = 0$ by Theorem 1.2 since $Qd_i \in \text{Span}[d_0, \ldots, d_k]$ by (1) and (2). The term $d^T_i Qd_i$ vanishes by the induction hypothesis on (3).

To prove (4) write

$$-g^T_k d_k = g^T_k g_k - \beta_k^{-1}g^T_k d_{k-1}$$

where $g^T_k d_{k-1} = 0$ by Theorem 1.2.

To prove (5) note that $g^T_{k+1}g_k = 0$ by Theorem 1.2 because $g_k \in \text{Span}[d_0, \ldots, d_k]$. Hence

$$g^T_{k+1}Qd_k = \frac{1}{\alpha_k} g^T_{k+1} [g_{k+1} - g_k] = \frac{1}{\alpha_k} g^T_{k+1} g_{k+1}.$$ 

Therefore,

$$\beta_k = \frac{1}{\alpha_k} \frac{g^T_{k+1} g_{k+1}}{d^T_k Qd_k} = \frac{g^T_{k+1} g_{k+1}}{g^T_k g_k}.$$

□

Remarks:

(1) The C–G method described above is a descent method since the values

$$f(x_0), f(x_1), \ldots, f(x_n)$$

form a decreasing sequence. Moreover, note that

$$\nabla f(x_k)^T d_k = -g^T_k g_k \quad \text{and} \quad \alpha_k > 0.$$

Thus, the C–G method behaves very much like the descent methods discussed previously.

(2) It should be observed that due to the occurrence of round-off error the C–G algorithm is best implemented as an iterative method. That is, at the end of $n$ steps, $f$ may not attain its global minimum at $x_n$ and the intervening directions $d_k$ may not be $Q$-conjugate. Consequently, at the end of the $n^{th}$ step one should check the value $\|\nabla f(x_n)\|$. If it is sufficiently small, then accept $x_n$ as the point at which $f$ attains its global minimum value; otherwise, reset $x_0 := x_n$ and run the algorithm again. Due to the observations in remark above, this approach is guaranteed to continue to reduce the function value if possible since the overall method is a descent method. In this sense the C–G algorithm is self correcting.
1.3. Extensions to Non-Quadratic Problems. If \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is not quadratic, then the Hessian matrix \( \nabla^2 f(x_k) \) changes with \( k \). Hence the C-G method needs modification in this case. An obvious approach is to replace \( Q \) by \( \nabla^2 f(x_k) \) everywhere it occurs in the C-G algorithm. However, this approach is fundamentally flawed in its explicit use of \( \nabla^2 f \). By using parts (4) and (5) of the conjugate gradient Theorem 1.3 and by trying to mimic the descent features of the C–G method, one can obtain a workable approximation of the C–G algorithm in the non–quadratic case.

The Non-Quadratic C-G Algorithm

Initialization: \( x_0 \in \mathbb{R}^n, g_0 = \nabla f(x_0), d_0 = -g_0, 0 < c < \beta < 1. \) Having \( x_k \) obtain \( x_{k+1} \) as follows:

Check restart criteria. If a restart condition is satisfied, then reset \( x_0 = x_n, g_0 = \nabla f(x_0), d_0 = -g_0 \); otherwise, set

\[
\alpha_k \in \left\{ \lambda \mid \lambda > 0, \nabla f(x_k + \lambda d_k)^T d \geq \beta \nabla f(x_k)^T d, \text{ and } f(x_k + \lambda d_k) - f(x_k) \leq c \lambda \nabla f(x_k)^T d \right\}
\]

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

\[
g_{k+1} := \nabla f(x_{k+1})
\]

\[
\beta_k := \begin{cases} \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} & \text{Fletcher-Reeves} \\ \max \left\{ 0, \frac{g_{k+1}^T (g_{k+1} - g_k)}{g_k^T g_k} \right\} & \text{Polak-Ribiere} \end{cases}
\]

\[
d_{k+1} := -g_{k+1} + \beta_k d_k
\]

\[
k := k + 1.
\]

Remarks

(1) The Polak-Ribiere update for \( \beta_k \) has a demonstrated experimental superiority. One way to see why this might be true is to observe that

\[
g_{k+1}^T (g_{k+1} - g_k) \approx \alpha_k g_{k+1}^T \nabla^2 f(x_k) d_k
\]

thereby yielding a better second–order approximation. Indeed, the formula for \( \beta_k \) in the quadratic case is precisely

\[
\frac{\alpha_k g_{k+1}^T \nabla^2 f(x_k) d_k}{g_k^T g_k}.
\]

(2) Observe that the Hessian is never explicitly referred to in the above algorithm.

(3) At any given iteration the procedure requires the storage of only 2 vectors if Fletcher-Reeves is used and 3 vectors if Polak-Ribiere is used. This is of great significance if \( n \) is very large, say \( n = 50,000 \). Thus we see that one of the advantages of the C-G method is that it can be practically applied to very large scale problems.

(4) Aside from the cost of gradient and function evaluations the greatest cost lies in the line search employed for the computation of \( \alpha_k \).

We now consider appropriate restart criteria. Clearly, we should restart when \( k = n \) since this is what we do in the quadratic case. But there are other issues to take into consideration. First, since \( \nabla^2 f(x_k) \) changes with each iteration, there is no reason to think that we are preserving any sort of conjugacy relation from one iteration to the next. In order
to get some kind of control on this behavior, we define a *measure* of conjugacy and if this measure is violated, then we restart. Second, we need to make sure that the search directions $d_k$ are descent directions. Moreover, (a) the angle between these directions and the negative gradient should be bounded away from zero in order to force the gradient to zero, and (b) the directions should have a magnitude that is comparable to that of the gradient in order to prevent ill-conditioning. The precise restart conditions are given below.

**Restart Conditions**

1. $k = n$
2. $|g_k^T g_k| \geq 0.2 g_k^T g_k$
3. $-2g_k^T g_k \geq g_k^T d_k \geq -0.2 g_k^T g_k$

Conditions (2) and (3) above are known as the Powell restart conditions.