# 1. Conjugate Direction Methods

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

$$\mathcal{P}: \quad \begin{array}{l} \text{minimize}\, f(x) \\ \text{subject to } x \in \mathbb{R}^n \end{array}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable. The underlying assumption of this section is that the dimension $n$ is very large, indeed, so large that the matrix $\nabla^2 f(x)$ cannot be held in storage. However, it will be assumed that the matrix vector product $\nabla^2 f(x)v$ can be computed for any vector $v$.

To better understand the numerical approach to be developed first consider a constrained version of $\mathcal{P}$:

$$\mathcal{P}_{(x_0,S)}: \quad \begin{array}{l} \text{minimize}\, f(x) \\ \text{subject to } x \in x_0 + S \end{array} ,$$

where $x_0 \in \mathbb{R}^n$ and $S \subset \mathbb{R}^n$ is a subspace with

$$x_0 + S = \{x_0 + s \; | s \in S\} \; .$$

The problem $\mathcal{P}_{(x_0,S)}$ is our first instance of a constrained problem. However, it is a constrained problem that is equivalent to an unconstrained problem. To see this let $v_0, v_1, \ldots, v_{k-1}$ be a basis for $S$. Then

$$S = \text{Span}(v_0, \ldots, v_{k-1}) = \text{Ran}(V), \quad \text{where} \quad V = [v_0,\ v_1,\ \ldots,\ v_{k-1}] \in \mathbb{R}^{n\times k}.$$

Hence

$$x_0 + S = x_0 + \text{Ran}(V) = \left\{x_0 + Vz \;\middle|\; z \in \mathbb{R}^k\right\}.$$

Therefore, the we can rewrite the problem $\mathcal{P}_{(x_0,S)}$ as

$$\mathcal{P}'_{(x_0,V)}: \quad \begin{array}{l} \text{minimize}\, f(x_0 + Vz) \\ \text{subject to } v \in \mathbb{R}^k \end{array} .$$

If $\bar{z}$ is a local solution to $\mathcal{P}'_{(x_0,V)}$, then $\bar{x} = x_0 + V\bar{z}$ is a local solution to $\mathcal{P}_{(x_0,S)}$, and conversely, if $\bar{x}$ is a local solution to $\mathcal{P}_{(x_0,S)}$, then any $\bar{z} \in \mathbb{R}^k$ for which $\bar{x} = x_0 + V\bar{z}$ is a local solution to $\mathcal{P}'_{(x_0,V)}$.

Due to this equivalence, it is possible to derive first-order necessary conditions for optimality in $\mathcal{P}_{(x_0,S)}$ from those for the unconstrained problem $\mathcal{P}'_{(x_0,V)}$.

**Theorem 1.1.** *(Subspace Optimality Theorem)*
*Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, $x_0 \in \mathbb{R}^n$, and $S$ a subspace of $\mathbb{R}^n$. If $\bar{x}$ is a local solution to $\mathcal{P}_{(x_0,S)}$, the $\nabla f(\bar{x}) \perp S$. If it is further assumed that $f$ is convex, then the condition $\nabla f(\bar{x}) \perp S$ is both necessary and sufficient for $\bar{x}$ to be a global solution to $\mathcal{P}_{(x_0,S)}$.*

*Proof.* As has already been observed, if $\bar{x}$ is a local solution to $\mathcal{P}_{(x_0,S)}$, then, since $\bar{x} \in x_0 + S$, there must exist $\bar{z} \in \mathbb{R}^k$ such that $\bar{x} = x_0 + V\bar{z}$ and any such $\bar{z}$ is a local solution to $\mathcal{P}'_{(x_0,V)}$.

Therefore, if $h(z) = f(x_0 + Vz)$, then

$$0 = \nabla h(\bar{z}) = V^T \nabla f(x_0 + V\bar{z}) = V^T \nabla f(\bar{x}) = \begin{bmatrix} v_1^T \nabla f(\bar{x}) \\ v_2^T \nabla f(\bar{x}) \\ \vdots \\ v_k^T \nabla f(\bar{x}) \end{bmatrix},$$

or equivalently, $\nabla f(\bar{x})^T v_i = 0,\ i = 1, 2, \ldots, k$, which is in turn equivalent to $\nabla f(\bar{x}) \perp S$.

If $f$ is assumed to be convex, then the final statement of the theorem follows from the convexity of the function $h$. □

1.2. **Conjugate Direction Methods.** In this section we focus on the problem $\mathcal{P}$ when $f$ has the form

$$f(x) := \frac{1}{2} x^T Q x - b^T x, \tag{1.1}$$

where $Q$ is a symmetric positive definite matrix. Our development in this section revolves around the notion of $Q$-conjugacy.

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

**Proposition 1.1.1** (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 $i_0 \in \{0, 1, \ldots, k\}$

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

Hence $\alpha_i = 0$ for each $i = 0, \ldots, k$. □

Let $x_0 \in \mathbb{R}^n$ and suppose that the vectors $d_0, d_1, \ldots, d_{k-1} \in \mathbb{R}^n$ are $Q$-conjugate. Set $S = \text{Span}(d_0, d_1, \ldots, d_{k-1})$. Since $Q$ is positive definite, $f$ is both coercive and strictly convex. Therefore, a solution $x^*$ to $\mathcal{P}_{(x_0,S)}$ exists, is unique, and satisfies $0 = V^T \nabla f(x^*) = V^T (Qx^* - b)$ by the Subspace Optimality Theorem. Since $x^* \in x_0 + S$, there are scalars $\mu_0, \ldots,\ \mu_{n-1}$ such that

$$x^* = x_0 + \mu_0 d_0 + \ldots + \mu_{k-1} d_{k-1}. \tag{1.2}$$

Since $0 = V^T \nabla f(x^*) = V^T (Qx^* - b)$, for each $j = 0, 1, \ldots, k-1$ we have

$$\begin{aligned} 0 &= d_j^T (Qx^* - b) \\ &= d_j^T \left( Q(x_0 + \mu_0 d_0 + \ldots + \mu_{k-1} d_{k-1}) - b \right) \\ &= d_j^T (Qx_0 - b) + \mu_0 d_j^T Q d_0 + \ldots + \mu_{k-1} d_j^T Q d_{k-1} \\ &= d_j^T \nabla f(x_0) + \mu_j d_j^T Q d_j\ , \end{aligned}$$

so that

$$\mu_j = \frac{-d_j^T \nabla f(x_0)}{d_j^T Q d_j} \quad j = 0, 1 \ldots, k-1 \ . \tag{1.3}$$

This observation motivates the following theorem.

**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*

$$\alpha_k := \arg\min\{f(x_k + \alpha d_k) : \alpha \in \mathbb{R}\}$$

*has the property that $f(x) = \frac{1}{2} x^T Q x - b^T x$ attains its minimum value on the affine set $x_0 +$ Span $\{d_0, \ldots, d_{k-1}\}$ at the point $x_k$.*

*Proof.* Let us first compute the value of the $\alpha_k$'s. Set

$$\begin{aligned} \varphi_k(\alpha) &= f(x_k + \alpha d_k) \\ &= \tfrac{\alpha^2}{2} d_k^T Q d_k + \alpha g_k^T d_k + f(x_k), \end{aligned}$$

where $g_k = \nabla f(x_k) = Q x_k - b$. Then $\varphi_k'(\alpha) = \alpha d_k^T Q d_k + g_k^T d_k$. Since $f$ is strictly convex so is $\phi$, and so $\alpha_k$ is the unique solution to $\phi'(\alpha) = 0$ which is given by

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k}.$$

Therefore,

$$x_k = x_0 + \alpha_0 d_0 + \alpha_1 d_1 + \cdots + \alpha_k d_k$$

with

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k}, \ k = 0, 1, \ldots, k.$$

Preceding the theorem it was shown that if $x^*$ is the solution to the problem

$$\min \{ f(x) \ | x \in x_0 + \mathrm{Span}(d_0, d_1, \ldots, d_k) \} \ ,$$

then $x^*$ is given by (1.2) and (1.3). Therefore, If we can now show that $\mu_j = \alpha_j$, $j = 0, 1, \ldots, k$, then $x^* = x_k$ which proves the result. For each $j \in \{0, 1, \ldots, k\}$ we have

$$\begin{aligned} \nabla f(x_j)^T d_j &= (Q x_j - b)^T d_j \\ &= \left(Q(x_0 + \alpha_0 d_0 + \alpha_1 d_1 + \cdots + \alpha_{j-1} d_{j-1}) - b\right)^T d_j \\ &= (Q x_0 - b)^T d_j + \alpha_0 d_0^T Q d_j + \alpha_1 d_1^T Q d_j + \cdots + \alpha_{j-1} d_{j-1}^T Q d_j \\ &= (Q x_0 - b)^T d_j \\ &= \nabla f(x_0)^T d_j \ . \end{aligned}$$

Therefore, for each $j \in \{0, 1, \ldots, k\}$,

$$\alpha_j = \frac{-\nabla f(x_j)^T d_j}{d_j^T Q d_j} = \frac{-\nabla f(x_0)^T d_j}{d_j^T Q d_j} = \mu_j,$$

which proves the result. □

As an immediate consequence of this theorem we obtain the following result.

**Theorem 1.3** (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 $\mathcal{P}$ with $f$ given by (1.1) after $n$ steps, that is $x_n = x^*$.*

1.3. **The Conjugate Gradient Algorithm.** The major drawback of the Conjugate Direction Algorithm of the previous section is that it seems to require that a set of $Q$-conjugate directions must be obtained before the algorithm can be implemented. This is in opposition to our working assumption that $Q$ is so large that it cannot be kept in storage since any set of $Q$-conjugate directions requires the same amount of storage as $Q$. However, it is possible to generate the directions $d_j$ one at a time and then discard them after each iteration of the algorithm. One example of such an algorithm is 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 $k = 0, 1, 2, \ldots$

$$\begin{array}{lll} \alpha_k & := -g_k^T d_k / d_k^T Q d_k & \\ x_{k+1} & := x_k + \alpha_k d_k & \\ g_{k+1} & := Qx_{k+1} - b & \qquad \text{(STOP if } g_{k+1} = 0\text{)} \\ \beta_k & := g_{k+1}^T Q d_k / d_k^T Q d_k & \\ d_{k+1} & := -g_{k+1} + \beta_k d_k & \\ k & := k + 1. & \end{array}$$

**Theorem 1.4.** *[*Conjugate Gradient Theorem*]*
*The C-G algorithm is a conjugate direction method. If it does not terminate at $x_k$ (i.e. $g_k \neq 0$), then*

(1) *Span* $[g_0, g_1, \ldots, g_k] =$ *span* $[g_0, Qg_0, \ldots, Q^k g_0]$
(2) *Span* $[d_0, d_1, \ldots, d_k] =$ *span* $[g_0, Qg_0, \ldots, Q^k g_0]$
(3) $d_k^T Q d_i = 0$ *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 $g_{k+1} \in$ Span$[g_0, \ldots, Q^{k+1} g_0]$ by the induction hypothesis on (1) and (2). Also $g_{k+1} \notin$ Span $[d_0, \ldots, d_k]$ otherwise $g_{k+1} = 0$ (by the Subspace Optimality Theorem ) since the method is a conjugate direction method up to step $k$ by the induction hypothesis. Hence

$g_{k+1} \notin$ Span $[g_0, \ldots, Q^k g_0]$ and so Span $[g_0, g_1, \ldots, g_{k+1}] =$ Span $[g_0, \ldots, Q^{k+1} g_0]$, 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

$$d_{k+1}^T Q d_i = -g_{k+1} Q d_i + \beta_k d_k^T Q d_i.$$

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

To prove (4) write

$$-g_k^T d_k = g_k^T g_k - \beta_{k-1} g_k^T d_{k-1}$$

where $g_k^T d_{k-1} = 0$ by Theorem 1.2.

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

$$g_{k+1}^T Q d_k = \frac{1}{\alpha_k} g_{k+1}^T [g_{k+1} - g_k] = \frac{1}{\alpha_k} g_{k+1}^T g_{k+1}.$$

Therefore,

$$\beta_k = \frac{1}{\alpha_k} \frac{g_{k+1}^T g_{k+1}}{d_k^T Q d_k} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}.$$

□

**Remarks:**

(1) The C–G method decribed 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

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

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

(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 guarenteed 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.

1.4. **Extensions to Non-Quadratic Problems.** If $f : \mathbb{R}^n \to \mathbb{R}$ 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 1.4 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 $x_k$ otain $x_{k+1}$ 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

$$\begin{array}{ll}
\alpha_k & \in \left\{ \lambda \;\middle|\; \begin{array}{c} \lambda > 0, \nabla f(x_k + \lambda d_k)^T d \geq \beta \nabla f(x_k)^T d_k, \text{ and} \\ f(x_k + \lambda d_k) - f(x_k) \leq c \lambda \nabla f(x_k)^T d_k \end{array} \right\} \\
x_{k+1} & := x_k + \alpha_k d_k \\
g_{k+1} & := \nabla f(x_{k+1}) \\
\beta_k & := \left\{ \begin{array}{rl} \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} & \text{Fletcher-Reeves} \\ \max\left\{0, \frac{g_{k+1}^T (g_{k+1} - g_k)}{g_k^T g_k}\right\} & \text{Polak-Ribiere} \end{array} \right. \\
d_{k+1} & := -g_{k+1} + \beta_k d_k \\
k & := k + 1.
\end{array}$$

**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 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 refered 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) $|g_{k+1}^T g_k| \geq 0.2 g_k^T g_k$
(3) $-2 g_k^T g_k \geq g_k^T d_k \geq -0.2 g_k^T g_k$

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