**Numerical Analysis** **Autumn 2017**

# Conjugate Gradients: November 24, 2018

*Notes: Motivation and Introduction* *Authors: Jake Roth*

## Introduction[1]

The conjugate gradient (CG) method was perceived for some time as a direct method for solving systems of linear equations. In exact arithmetic, the method produces the exact solution in a finite number of steps. More precisely, for an $n \times n$ system, CG "converges" in at most $n$ steps. The advantages of conjugate gradients as an iterative method were not widely appreciated until much later. Not only does CG provide good approximate solutions with fewer iterations than what is required to produce an exact result, it can also be seen to be a more adaptive procedure than the stationary iterative methods studied previously in section 8.1.

We will develop the method in a sequence of steps to put it in context. It can be applied to symmetric, positive definite matrices, so we will limit our discussion to linear systems with such matrices.

## 1.1 CG Iteration

Previously, we saw that a downside of Gradient Descent was the possibility of repeating search directions. More precisely, at any iteration, the direction of steepest descent might be a direction that we had previously explored at a prior iteration. We will see that CG performs clever steps to avoid searching and stepping in previously traversed search directions. The intuition for this approach is largely based on Michael Zibulevsky's[2] lectures.

### 1.1.1 Motivation

Let's work on the same minimization problem

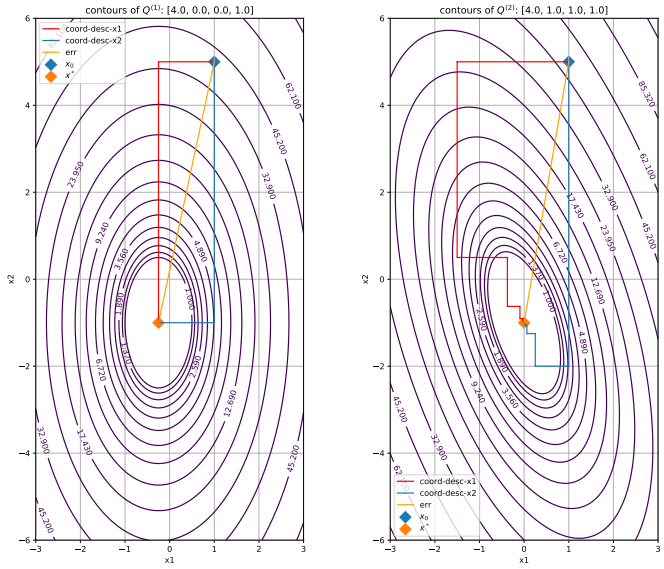
$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) = \tfrac{1}{2} x^T Q x + b^T x \tag{1.1}$$

as in Gradient Descent. Here we know that the optimal point is $x^* = -Q^{-1}b$. To consider two particular examples, take

$$Q^{(1)} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}, \quad b^{(1)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \text{and} \quad Q^{(2)} = \begin{bmatrix} 4 & 1 \\ 1 & 1 \end{bmatrix}, \quad b^{(2)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \text{with} \quad x_0 = \begin{bmatrix} 1 \\ 5 \end{bmatrix}.$$

[1] L. Ridgway Scott's NA Two Book
[2] Michael Zibulevsky's Optimization Course

Figure 1.1: Coordinate descent paths (along standard coordinate-axes) for diagonal $Q$ (left) and non-diagonal $Q$ (right). Note the steps along repeated directions in the right-hand plot.




## Diagonal Case

When $Q$ is diagonal, observe that the error $x^{(0)} - x^*$ decomposes into a component specified by the first coordinate axis and a component specified by the second coordinate axis. Intuitively, this happens because there is no interaction between $x_1$ and $x_2$ when computing $f$ (i.e., when viewing $Q^{(1)}$ as a covariance matrix, there is no correlation between $x_1$ and $x_2$).

To minimize $f$ starting from $x^{(0)} = (1, 5)^T$, we can then eliminate an entire component of the error when $Q = Q^{(1)}$ fig. 1.1 by stepping the appropriate amount along the $e_1$ or $e_2$ coordinate-axis directions. We have this nice procedure because the contours of $f$ are aligned with the standard coordinate axes and we have "good" orthogonal search directions $s_1 \coloneqq e_1$ and $s_2 \coloneqq e_2$ (without doing any work) simply by following the structure given by $Q^{(1)}$. Thus, in two steps, coordinate-descent will have exactly reached $x^*$. This motivates the merit of having orthogonal search directions in $n$ dimensions: given an orthogonal basis of dimension $n$, we can decompose the error into $n$ components and minimize each term sequentially.

## Non-Diagonal Case

When $Q$ is not diagonal, as in the case of $Q = Q^{(2)}$, $f$'s contours are still elliptical but are now skewed by the interaction between $x_1$ and $x_2$. In this way, $Q$ "warps" the standard coordinate system and creates a new "$Q$-geometry" so that the contours' principal axes (eigenvectors) are no longer aligned with the standard coordinate axes. We can summarize the new geometry through its inner product by defining $\langle x, x' \rangle_Q \coloneqq x^T Q x'$ (note that we could have done this for the diagonal case, but it is less informative there). We then observe the following relationship between the new inner product and the original inner product: $\langle x, x' \rangle_Q \coloneqq \langle x, Qx' \rangle$ where $\langle x, x' \rangle \coloneqq x^T I x' = x^T x'$ and $I$ the identity matrix.

If we now tried to minimize $f$ by following the same line-search procedure as in the diagonal case, we would observe a zig-zag path toward $x^*$ (like the right-hand plot of fig. 1.1). This hints at an inherent inconsistency

in the attempt to use the notion of orthogonality in the standard inner-product to define search directions in the new geometry.

To overcome this issue, we might try to focus our efforts on reframing the problem in $Q$'s geometry. If we tried to "undo" or unwarp the $Q$ inner-product, we could pre-multiply the vector in the second slot of the inner product by $Q^{-1}$, i.e. $\langle x, Q^{-1}y\rangle = \langle x, x'\rangle$ for $y = Qx'$. Writing $f$ in terms of the $Q$ inner product gives $f(x) = \frac{1}{2}\langle x, x\rangle_Q + \langle x, Q^{-1}b\rangle_Q$ and then unwarping by pre-multiplying the second slot by $Q^{-1}$[3] and minimizing the new function $\tilde{f}$ where

$$\tilde{f}(x) \coloneqq \tfrac{1}{2}\langle x, Q^{-1}x\rangle_Q + \langle x, Q^{-1}Q^{-1}b\rangle_Q = \tfrac{1}{2}x^Tx + b^TQ^{-1}x.$$

Then, performing a single step of Gradient Descent will give $x^* = -Q^{-1}b$ (this method of pre-multiplying by $Q^{-1}$ is Newton's method in higher dimensions). However, this required knowledge of $Q^{-1}$, which if we knew, would give us $x^* = -Q^{-1}b$ analytically.

As an alternative, we might try to diagonalize $Q$ to produce the Diagonal Case above so that we can apply sequential line-searches to minimize the error. In this case, what we need are directions $s_1$ and $s_2$ which are orthogonal in the $Q$ geometry (or "$Q$-orthogonal", i.e., $s_1^TQs_2 = 0$). This idea forms the foundation of the Conjugate Gradients method.

### 1.1.2 Formulation as a Minimization Problem

Consider the Taylor expansion around our minimization problem eq. (1.1)

$$f(x^{(0)} + s) \approx f(x^{(0)}) + (g^{(0)})^Ts + \tfrac{1}{2}s^TQs = f(x^{(0)}) + (g^{(0)})^Ts + \|s\|_Q^2 \tag{1.2}$$

where $g^{(0)} = Qx^{(0)} + b$ is the gradient of $f$ evaluated at $x^{(0)}$ and $s$ is our step.

Suppose now that we are given a set $\{d^1, \dots, d^n\}$ of $n$ vectors which are $Q$-orthogonal so that they span $\mathbb{R}^n$ and $(d^i)^TQd^j = 0$ for $i \neq j$. In our original space, these vectors form a basis, and in the space transformed by $Q$, these vectors form an orthogonal basis. Later, we will see how these vectors formally represent an "appropriate" basis for viewing the minimization problem's error.

First, expand the problem's error $e = x^{(0)} - x^*$ in terms of the $d^i$s. Ideally we would choose our search direction $s = e$ for true error $e$, but since $e$ is unknown, we might consider expanding $s$ in terms of the $Q$-orthogonal directions. To build an intuition for why we'd want to expand in terms of these directions, recall that in the case where $Q = I$, $I$-orthogonal directions will ensure that we don't traverse previously explored directions. Analogously, with $Q \neq I$, we ensure that we take into account the natural curvature of the space induced by $Q$ inner product.

Thus, expanding the search direction $s$ in terms of $Q$-orthogonal directions, we have

$$s = \sum_{i=1}^n \alpha_i d^i, \quad \|s\|_Q^2 = \sum_{i=1}^n\sum_{j=1}^n (\alpha_i d^i)q_{ij}(\alpha_j d^j) = \sum_{i=1}^n \alpha_i^2\|d^i\|_Q^2. \tag{1.3}$$

for some $\alpha_i \in \mathbb{R}$, $i = 1, \dots, n$. Substituting these into eq. (1.2) gives

$$f(x^{(0)} + s) = f(x^{(0)}) + \sum_{i=1}^n \left[\alpha_i(g^{(0)})^Td^i + \alpha_i^2\|d^i\|_Q^2\right] =: \phi(\alpha) \tag{1.4}$$

[3]Note that we could also change coordinates by pre-multiplying $x$ by $Q^{-1/2}$ and minimize $\tilde{f}(y) \coloneqq y^Ty + b^TQ^{-1/2}y$ for $y = Q^{1/2}x$ to obtain $x^* = Q^{-1}b$ which is often how Newton's method is introduced

for $\alpha = (\alpha_1, \ldots, \alpha_n)^T \in \mathbb{R}^n$.

The key observation here is that minimizing $\phi(\alpha)$ is equivalent to minimizing $\phi_i(\alpha_i) = \alpha_i (g^{(0)})^T d^i + \alpha_i^2 \|d^i\|_Q^2$ separately for each $i \in \{1, \ldots, n\}$ since the $d^i$ are $Q$-orthogonal. That is, the minimization problem is *separable* and consists of $n$ 1-dimensional line-search problems of the sort we minimized during Gradient Descent. We find the analytic solution by differentiating with respect to $\alpha_i$, giving

$$\frac{d}{d\alpha_i} = 0 = (g^{(0)})^T d^i + 2\alpha_i \|d^i\|_Q^2 \implies \alpha_i = -\frac{(g^{(0)})^T d^i}{\|d^i\|_Q^2}. \tag{1.5}$$

This method is called *Conjugate Directions* (CD) since the directional components $d^i$ are $Q$-orthogonal.

### 1.1.3 Expanding Manifold Property

An important result from the above procedure is that at $k \leq n$ iterations, CD minimizes $f$ over the affine subspace $M^k = \{x^{(0)} + \text{span}\{d^1, \ldots, d^k\}\}$. To see this, we describe the $k^{th}$ iterate as

$$x^{(k)} = \underset{\alpha_1, \ldots, \alpha_k}{\text{argmin}} \left\{ f\left( x^{(0)} + \sum_{i=1}^{k} \alpha_i d^i \right) \right\} \tag{1.6}$$

$$= \underset{x \in M^k}{\text{argmin}} \left\{ f(x) \right\}. \tag{1.7}$$

From this representation, we make the following observation:

**Lemma 1.1 (Gradient-Manifold Orthogonality)** *The gradient of $f$ evaluated at the $k$-th iterate is orthogonal to the $k$-dimensional manifold spanned by the first $k$ directions. That is, $g^{(k)} \perp_I M^k$ (in the standard inner product denoted by identity $I$).*

**Proof:** The lemma follows since at any minimum, the directional derivative over any subspace must not give a descent direction. In greater detail, let

$$\alpha = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} \in \mathbb{R}^k, \quad D = \begin{bmatrix} | & | & & | \\ d^1 & d^2 & \ldots & d^k \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times k}.$$

Then $x = x^{(0)} + D\alpha$. Let $\alpha^* \coloneqq \underset{\alpha}{\text{argmin}} \left\{ f(x^{(0)} + D\alpha) \right\}$. Then

$$\nabla_\alpha f = D^T \nabla_x f(x^{(0)} + D\alpha) \implies x^* = x^{(0)} + D\alpha^* \tag{1.8}$$

so $D^T \nabla_x f(x^*) = 0$ implies that

$$\begin{aligned} (d^{(0)})^T \nabla_x f(x^*) &= 0 \\ &\vdots \\ (d^{(k)})^T \nabla_x f(x^*) &= 0 \end{aligned}$$

or in other words, $(g^{(k)}) \perp_I M^k$. (**Note**: recall that the directional derivative is the inner product of the gradient with a direction.) ■

### 1.1.4 CG Algorithm

We first present the pseudocode for the Gram-Schmidt CG algorithm and then discuss its origins. There are three ingredients to get from one iterate to the next: the current position, the search direction, and the step length. Of these, computing the search direction is most costly, and we will focus on ways of minimizing the computational cost.

**Algorithm 1** Gram-Schmidt Conjugate Gradient

**procedure** GSCG($Q, b, x_0, \epsilon$)
  $d^{(0)} \leftarrow -g^{(0)} = -(Qx^{(0)} + b)$, $e \leftarrow \infty$, $k \leftarrow 1$, $n \leftarrow$length($x_0$)
  **if** $k < n$ and $e > \epsilon$ **then**
    $\alpha_k \leftarrow \frac{-\langle g^{(k)}, d^{(k)} \rangle_I}{\langle d^{(k)}, d^{(k)} \rangle_Q}$
    $x^{(k+1)} \leftarrow x^{(k)} + \alpha_k d^{(k)}$
    $d^{(k+1)} \leftarrow -g^{(k+1)} + \sum_{j=0}^{k} \frac{\langle g^{(k+1)}, d^{(j)} \rangle_Q}{\langle d^{(j)}, d^{(j)} \rangle_Q} d^{(j)}$
    $e \leftarrow \|d^{(k+1)}\|_2$
    $k \leftarrow k + 1$
  **end if**
**end procedure**

## $Q$-Orthogonalization and Conjugacy

As we saw, CD can express the true error $e$ as a linear combination of $Q$-orthogonal (or *conjugate*) search directions. If we are given $n$ such directions, it is possible to express the error fully in terms of these vectors by solving $n$ 1-$D$ optimization problems. The question then becomes how to efficiently compute the direction vectors, $d^{(i)}$. One approach would be to take an arbitrary vector $v$ as the first direction $d^{(0)}$ and use Gram-Schmidt to $Q$-orthogonalize the subsequent directions. It turns out that choosing $d^{(0)} = -g^{(0)}$ and orthogonalizing subsequent gradients yields an efficient Gram-Schmidt process with a three-term recurrence relation.

We modify traditional Gram-Schmidt so that each new direction is $Q$-orthogonal. We use the gradients at each step as our orthogonalization vectors so that we have

$$d^{(k+1)} = \text{proj}_{M^k}(g^{(k+1)}) = -g^{(k+1)} + \sum_{j=0}^{k} \frac{\langle g^{(k+1)}, d^{(j)} \rangle_Q}{\langle d^{(j)}, d^{(j)} \rangle_Q} d^{(j)}. \tag{1.9}$$

Performing an analytic line search on $f(x + \alpha_k d^{(k)}) = \frac{1}{2}(x + \alpha_k d^{(k)})^T Q(x + \alpha_k d^{(k)}) + b^T(x + \alpha_k d^{(k)})$ by differentiating w.r.t. $\alpha_k$ and setting equal to zero gives

$$\alpha_k = \frac{-(g^{(k)})^T d^k}{(d^{(k)})^T Q d^k}. \tag{1.10}$$

When we have the iterates $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$, we can reinterpret $\alpha_k$ as the though we had performed a single 1-$D$ Newton step of the form

$$x^{(k+1)} = x^{(k)} + f'_{d^{(k)}} / f''_{d^{(k)}}.$$

Here $f'_{d^{(k)}}$ denotes the directional derivative of $f$ in the direction $d^{(k)}$, and similarly $f''_{d^{(k)}}$ denotes the second directional derivative in the direction $d^{(k)}$.

## Efficient Orthogonalization

Using the gradient as the $Q$-orthogonalization vectors, we will see three simplifications in the formulas for computing the $n$ $Q$-orthogonal directions.

### Simplification 1

From the update formula for $x$ in algorithm 1, we can solve for $d^{(k)}$ in terms of the quantities $x^{(k)}, x^{(k+1)}$, and $\alpha_k$ so that

$$d^{(k)} = \frac{1}{\alpha_k}(x^{(k+1)} - x^{(k)}). \tag{1.11}$$

Taking a closer look at a particular transformed direction $Qd^{(j)}$ in eq. (1.9), observe that we can use the recurrence relation from eq. (1.11) it in terms of gradients as follows

$$Qd^{(j)} = \frac{1}{\alpha_j}Q(x^{(j+1)} - x^{(j)}) = \frac{1}{\alpha_j}(g^{(j+1)} - g^{(j)})$$

since the terms involving $b$ from the original eq. (1.1) cancel for each $g^{(i)}$ with $i \in \{j, j+1\}$ here.

Expressing the $Q$-transformed directions in terms of the gradients now allows us to use the expanding manifold property. Observing that $g^{(j+1)} \perp_I g^{(j)}, g^{(j-1)}, \ldots, g^{(0)}$ in the standard inner product, we will have significant cancellation in eq. (1.9). Specifically, we have

$$d^{(k+1)} = -g^{(k+1)} + \frac{(g^{(k+1)})^T(g^{(k+1)} - g^{(k)})}{(d^{(k)})^T(g^{(k+1)} - g^{(k)})}d^{(k)} = -g^{(k+1)} + \beta_k d^{(k)} \tag{1.12}$$

where

$$\beta_k = \frac{(g^{(k+1)})^T g^{(k+1)}}{(d^{(k)})^T(-g^{(k)})} \tag{1.13}$$

since the expanding manifold property also gives $d^{(k)} \perp_I g^{(k+1)}$.

### Simplification 2

We can further eq. (1.13) by expanding $d^{(k)} = -g^{(k)} + \beta_{k-1}d^{(k-1)}$ and again using the expanding manifold property noting that $d^{(k)} \perp_I \beta_{k-1}d^{(k-1)}$ so that

$$\beta_k = \frac{(g^{(k+1)})^T g^{(k+1)}}{(-g^{(k)})^T(-g^{(k)})} = \frac{\|g^{(k+1)}\|_2^2}{\|g^{(k)}\|_2^2}. \tag{1.14}$$

### Simplification 3

Finally, we can utilize quantities that we've previously computed to achieve an additional simplification in computing the gradient. Note that

$$g^{(k+1)} = Qx^{(k+1)} + b = Q\left(x^{(k)} + \alpha_k d^{(k)}\right) + b$$

and that $Qx^{(k)}$ was previously computed to when determining $g^{(k)}$. This means that at each iteration, the CG update only requires a matrix-vector product (in addition to less expensive vector-vector products).

## Revised Algorithm

In light of the simplifications above, we can rewrite algorithm 1 as

**Algorithm 2** Revised Conjugate Gradient

**procedure** CG($Q, b, x_0, \epsilon$)
  $d^{(0)} \leftarrow -g^{(0)} = -(Qx^{(0)} + b)$, $e \leftarrow \infty$, $k \leftarrow 1$, $n \leftarrow$length($x_0$)
  **if** $k < n$ and $e > \epsilon$ **then**
    $\alpha_k \leftarrow \frac{-\langle g^{(k)}, d^{(k)} \rangle_I}{\langle d^{(k)}, d^{(k)} \rangle_Q}$
    $x^{(k+1)} \leftarrow x^{(k)} + \alpha_k d^{(k)}$
    $g^{(k+1)} \leftarrow g^{(k)} + \alpha + k Q d^{(k)}$
    $\beta_{k+1} \leftarrow \frac{\langle g^{(k+1)}, g^{(k+1)} \rangle}{\langle g^{(k)}, g^{(k)} \rangle}$
    $d^{(k+1)} \leftarrow -g^{(k+1)} + \beta_{k+1} d^{(k)}$
    $e \leftarrow \|d^{(k+1)}\|_2$
    $k \leftarrow k + 1$
  **end if**
**end procedure**

Finally, we conclude with clarification on the etymology of the algorithm.

> The name "Conjugate Gradients" is a bit of a misnomer, because the gradients are not conjugate, and the conjugate directions are not all gradients. "Conjugated Gradients" would be more accurate. [4]

### 1.1.5 Problems

- Show that CG is not affine-invariant. Find an example in 3 dimensions where the path will depend on the starting point.
- Show that CG is orthogonal-invariant. That is, for any orthogonal matrix $R$, define $\bar{b} = R^T b$ and $\bar{Q} = R^T Q R$. Show that the CG iterates for minimizing $\tilde{f}(\bar{x}) = x^T \bar{Q} x + \bar{b}^T x$ are of the form $\bar{x} = Rx$.[5]
- Show that CG produces monotonically decreasing iterates (i.e., $f(x^{(k+1)}) \leq f(x^{(k)})$).

### 1.1.6 Writeups

- For $f(x) \coloneqq \frac{1}{2} x^T H x + b^T x$ where $H \succeq 0$ with $h(y) = Ay$ for $A \in GL(n, \mathbb{R})$, show that CG is not affine invariant

  We have $\nabla f(x) = Hx$ with $\nabla^2 f(x) = H$ and $g(y) = f(Ay)$.

  Note that the first step of CG begins with a step of GD. Thus, we may not have affine invariance from the first step. Since CG converges in $n$ steps, if we consider a $2 \times 2$ example without adding a constant term $b$, then while we might have $x^{(i)} \neq Ay^{(i)}$ for $i = 0, 1$, we will have it for $i = 2$ since the minimum is $(0, 0)^T$ which is affine invariant. So, consider adding $b = (1, 1)^T$ in the following problem

$$H = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \quad x^{(0)} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}.$$

[4] J.R. Shewchuck

[5] Orthogonal-invariance: p.562

Then we have

$$\begin{aligned} x^{(0)} &= (2,1)^T & y^{(0)} &= (1,2)^T & Ax^{(0)} &= (1,2)^T \\ x^{(1)} &= (-0.78,0.31)^T & y^{(1)} &= (-0.12,-0.60)^T & Ax^{(1)} &= (0.31,-0.78)^T \\ x^{(2)} &= (-2/3,-1)^T & y^{(2)} &= (-2,1/3)^T & Ax^{(2)} &= (-1,-2/3)^T \end{aligned}$$

and since $y^{(i)} \neq Ax^{(i)}$ for $i > 0$, we have a counter example.

- Similar to above but use conjugate directions.
- Let $f(x) \coloneqq \frac{1}{2}x^T Ax - b^T x$. We can follow my online communication [6]. Using $r_k = b - Ax_k$, the fact that $r_k^T p_k = r_k^T r_k$, and the definition of $\alpha_k = r_k^T r_k / p_k^T A p_k$, we have

$$f(x_{k+1}) = f(x_k) + \frac{1}{2}\alpha_k^2 p_k^T A p_k - \alpha_k r_k^T p_k = f(x_k) - \frac{1}{2}\frac{(r_k^T r_k)^2}{p_k^T A p_k}.$$

Since $A$ is symmetric positive definite,

$$f(x_k) - f(x_{k+1}) = \frac{1}{2}\frac{(r_k^T r_k)^2}{p_k^T A p_k} > 0.$$

[6] MSE communication