# Convex Optimization CMU-10725

## Conjugate Direction Methods

Barnabás Póczos & Ryan Tibshirani

ML MACHINE LEARNING DEPARTMENT

Carnegie Mellon.
School of Computer Science

# Conjugate Direction Methods

# Books to Read

**David G. Luenberger, Yinyu Ye**: Linear and Nonlinear Programming

**Nesterov**: Introductory lectures on convex optimization

**Bazaraa, Sherali, Shetty**: Nonlinear Programming

**Dimitri P. Bestsekas**: Nonlinear Programming

# Motivation

Conjugate direction methods can be regarded as being between the method of steepest descent and Newton’s method.

**Motivation**:

- ❑ steepest descent is slow. Goal: Accelerate it!
- ❑ Newton method is fast… BUT:
  we need to calculate the inverse of the Hessian matrix…

**Something between steepest descent and Newton method?**

# Conjugate Direction Methods

**Goal:**

- Accelerate the convergence rate of steepest descent
- while avoiding the high computational cost of Newton's method

Originally developed for solving the quadratic problem:

$$\left( \min_{x \in \mathbb{R}^n} \frac{1}{2} x^T Q x - b^T x \right)$$

Assume matrix $Q \in \mathbb{R}^{n \times n}$ is positive definite

**Equivalently, our goal is to solve:** $\underline{Qx = b},\ x \in \mathbb{R}^n$

Conjugate direction methods can solve this problem at most n iterations (usually for large n less is enough)

# Conjugate Direction Methods

- ❑ Algorithm for the numerical solution of linear equations, whose matrix Q is symmetric and positive-definite.

- ❑ An iterative method, so it can be applied to systems that are too large to be handled by direct methods
  (such as the Cholesky decomposition.)

- ❑ Algorithm for seeking minima of nonlinear equations.

# Conjugate directions

**Definition** [Q-conjugate directions]

Let $Q$ be a symmetric matrix.

$\{d_1, d_2, \ldots, d_k\}$ vectors ($d_i \in \mathbb{R}^n$, $d_i \neq 0$) are $Q$-orthogonal (conjugate) w.r.t $Q$, if

$$d_i^T Q d_j = 0, \ \forall i \neq j$$

- □ In the applications that we consider, the matrix Q will be positive definite but this is not inherent in the basic definition.

- □ If Q = 0, any two vectors are conjugate.

- □ if Q = I, conjugacy is equivalent to the usual notion of orthogonality.

# Linear independence lemma

**Lemma** [Linear Independence]

Let $Q$ be positive definite.

If $\{d_1, d_2 \ldots, d_k\}$ vectors are $Q$-conjugate, then they are linearly independent.

**Proof:** [Proof by contradiction]

If $\underline{d_k} = \alpha_1 d_1 + \ldots + \alpha_{k-1} d_{k-1}$, then

$$0 < \underline{d_k^T Q d_k} = \underline{d_k^T Q(\alpha_1 d_1 + \ldots + \alpha_{k-1} d_{k-1}})$$

$$= \alpha_1 \underbrace{d_k^T Q d_1}_{0} + \ldots + \alpha_{k-1} \underbrace{d_k^T Q d_{k-1}}_{0} = 0 \quad ↯$$

# Why is Q-conjugacy useful?

# Quadratic problem

**Goal**: $$\min_{x\in\mathbb{R}^n} \frac{1}{2}x^TQx - b^Tx$$

Assume matrix $Q \in \mathbb{R}^{n\times n}$ is positive definite

the unique solution to this problem is also the unique solution to the linear equation:

$$Qx = b, \ x \in \mathbb{R}^n$$

Let $x^*$ denote the solution.

Let $\{d_0, d_1, \ldots, d_{n-1}\}$ vectors be $Q$-conjugate.

Since $\{d_0, d_1, \ldots, d_{n-1}\}$ vectors are independent,

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

# Importance of Q-conjugancy

$$\underline{x^*} = \underline{\alpha_0 d_0 + \ldots + \alpha_{n-1} d_{n-1}}$$

Therefore,

$$\boxed{\underline{d_i^T Q x^*}} = \underline{d_i^T Q(\alpha_0 d_0 + \ldots + \alpha_{n-1} d_{n-1})} = \underline{\alpha_i \boxed{d_i^T Q d_i}}$$

$$\Rightarrow \alpha_i = \frac{d_i^T \overbrace{Q x^*}^{b}}{d_i^T Q d_i} = \boxed{\frac{d_i^T b}{d_i^T Q d_i}}$$

We don't need to know $x^*$ to get $\alpha_i$!

$Qx^*=b$ is the step where standard orthogonality is not enough anymore,

We need to use Q-conjugacy.

# Importance of Q-conjugancy

$$\underline{x^*} = \underline{\alpha_0 d_0 + \ldots + \alpha_{n-1}} d_{n-1}$$

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

$$x_0 = 0$$

$$\alpha_i = \frac{d_i^T b}{d_i^T Q d_i}$$

$$\underline{x^*} = \sum_{i=0}^{d-1} \alpha_i d_i = \underline{\sum_{i=0}^{d-1} \frac{d_i^T b}{d_i^T Q d_i} d_i}$$

No need to do matrix inversion! We only need to calculate inner products.

The expansion for $x^*$ can be considered to be the result of an iterative process of $n$ steps where at the ith step $\alpha_i d_i$ is added.

This can be generalized further such a way that the *starting point of the iteration can be arbitrary $x_0$*

# Conjugate Direction Theorem

In the previous slide we had

$$\alpha_k = \frac{d_k^T b}{d_k^T Q d_k} \qquad x^* = \sum_{k=0}^{d-1} \alpha_k d_k = \sum_{k=0}^{d-1} \frac{d_k^T b}{d_k^T Q d_k} d_k$$

**Theorem** [Conjugate Direction Theorem]

Let $\{d_0, d_1, \ldots, d_{n-1}\}$ vectors be $Q$-conjugate.

$x_0 \in \mathbb{R}^n$ be an arbitrary starting point.

$$x_{k+1} = x_k + \alpha_k d_k \qquad \text{[update rule]}$$

$$g_k = Qx_k - b \qquad \text{[gradient of } f\text{]}$$

$f(x)=\frac{1}{2}x^TQx-bx$

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k} = -\frac{(Qx_k - b)^T d_k}{d_k^T Q d_k}$$

Then after n steps, $x_n = x^*$.

No need to do matrix inversion! We only need to calculate inner products.

# Proof

Since $\{d_0, d_1, \ldots, d_{n-1}\}$ vectors are independent,

$$\Rightarrow x^* - x_0 = \alpha_0 d_0 + \ldots + \alpha_{n-1} d_{n-1}$$

for some $\alpha_0, \ldots, \alpha_{n-1}$

Using the $x_{k+1} = x_k + \alpha_k d_k$ update rules, we have

$$x_1 = x_0 + \alpha_0 d_0$$

$$x_2 = x_0 + \alpha_0 d_0 + \alpha_1 d_1$$

$$x_k = x_0 + \alpha_0 d_0 + \alpha_1 d_1 + \ldots + \alpha_{k-1} d_{k-1}$$

$$x_n = x_0 + \alpha_0 d_0 + \alpha_1 d_1 + \ldots + \alpha_{n-1} d_{n-1} = x^*$$

Therefore, it is enough to prove that with these $\alpha_k$ values we have

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

# Proof

We already know

- $x^* - x_0 = \alpha_0 d_0 + \ldots + \alpha_{n-1} d_{n-1}$
- $x_k - x_0 = \alpha_0 d_0 + \alpha_1 d_1 + \ldots + \alpha_{k-1} d_{k-1}$

$$g_k = Qx_k - b = Qx_k - Qx^* = Q(x_k - x^*)$$

Therefore,

$$d_k^T Q(x^* - x_0) = d_k^T Q(\alpha_0 d_0 + \ldots + \alpha_{n-1} d_{n-1}) = \alpha_k d_k^T Q d_k$$

$$\Rightarrow \alpha_k = \frac{d_k^T Q(x^* - x_0)}{d_k^T Q d_k}$$

$$d_k^T Q(x_k - x_0) = d_k^T Q(\alpha_0 d_0 + \alpha_1 d_1 + \ldots + \alpha_{k-1} d_{k-1}) = 0$$

$$d_k^T Q(x^* - x_0) = d_k^T Q(x^* - x_k + x_k - x_0) = d_k^T Q(x^* - x_k)$$

$$\alpha_k = \frac{d_k^T Q(x^* - x_0)}{d_k^T Q d_k} = \frac{d_k^T Q(x^* - x_k)}{d_k^T Q d_k} = -\frac{d_k^T g_k}{d_k^T Q d_k}$$

Q.E.D.

# Another motivation for Q-conjugacy

**Goal**:
$$\min_{x\in\mathbb{R}^n} \overbrace{\frac{1}{2}x^TQx - b^Tx}^{f(x)}$$

$$\underline{x - x_0} = \underline{\alpha_0 d_0 + \ldots + \alpha_{n-1}d_{n-1}} \text{ for some } \{\alpha_i\}_{i=0}^{n-1} \in \mathbb{R}$$

Therefore,

$$f(x) = \frac{1}{2}[x_0 + \sum_{j=0}^{n-1}\alpha_j d_j]^T Q[x_0 + \sum_{j=0}^{n-1}\alpha_j d_j] - b^T[x_0 + \sum_{j=0}^{n-1}\alpha_j d_j]$$

$$f(x) = c + \sum_{j=0}^{n-1}\frac{1}{2}\underline{[x_0+\alpha_j d_j]^T Q[x_0+\alpha_j d_j] - b^T[x_0+\alpha_j d_j]}$$

**n separate 1-dimensional optimization problems!**

# Expanding Subspace Theorem

# Expanding Subspace Theorem

Let $\mathcal{B}_k = \text{span}(d_0, \ldots, d_{k-1}) \subset \mathbb{R}^n$

[$k$-dimensional subspace of $\mathbb{R}^n$]

We will show as the method of conjugate directions progresses each $x_k$ minimizes the objective $f(x) = \frac{1}{2}x^TQx - b^Tx$ both over $x_0 + \mathcal{B}_k$ and $x_{k-1} + \alpha d_{k-1}$, $\alpha \in \mathbb{R}$

$$x_k = \underset{\substack{x = x_{k-1} + \alpha d_{k-1} \\ \alpha \in \mathbb{R}}}{\arg\min} \frac{1}{2}x^TQx - b^Tx$$

$$x_k = \underset{x = x_0 + \mathcal{B}_k}{\arg\min} \frac{1}{2}x^TQx - b^Tx$$

# Expanding Subspace Theorem

**Theorem** [Expanding Subspace Theorem]

$$\left.\begin{array}{l} \text{Let } \{d_i\}_{i=0}^{n-1} \text{ be a sequence of } Q\text{-conjugate vectors in } \mathbb{R}^n \\ \quad x_0 \in \mathbb{R}^n \text{ arbitrary} \\ \quad x_{k+1} = x_k + \alpha_k d_k \\ \quad \alpha_k = -\dfrac{g_k^T d_k}{d_k^T Q d_k} \end{array}\right\}$$

$$\Rightarrow \left\{\begin{array}{l} x_k = \underset{\substack{x = x_{k-1} + \alpha d_{k-1} \\ \alpha \in \mathbb{R}}}{\arg\min} \overbrace{\tfrac{1}{2} x^T Q x - b^T x}^{f(x)} \\ \\ x_k = \underset{x = x_0 + \mathcal{B}_k}{\arg\min} \tfrac{1}{2} x^T Q x - b^T x \end{array}\right.$$

# Expanding Subspace Theorem

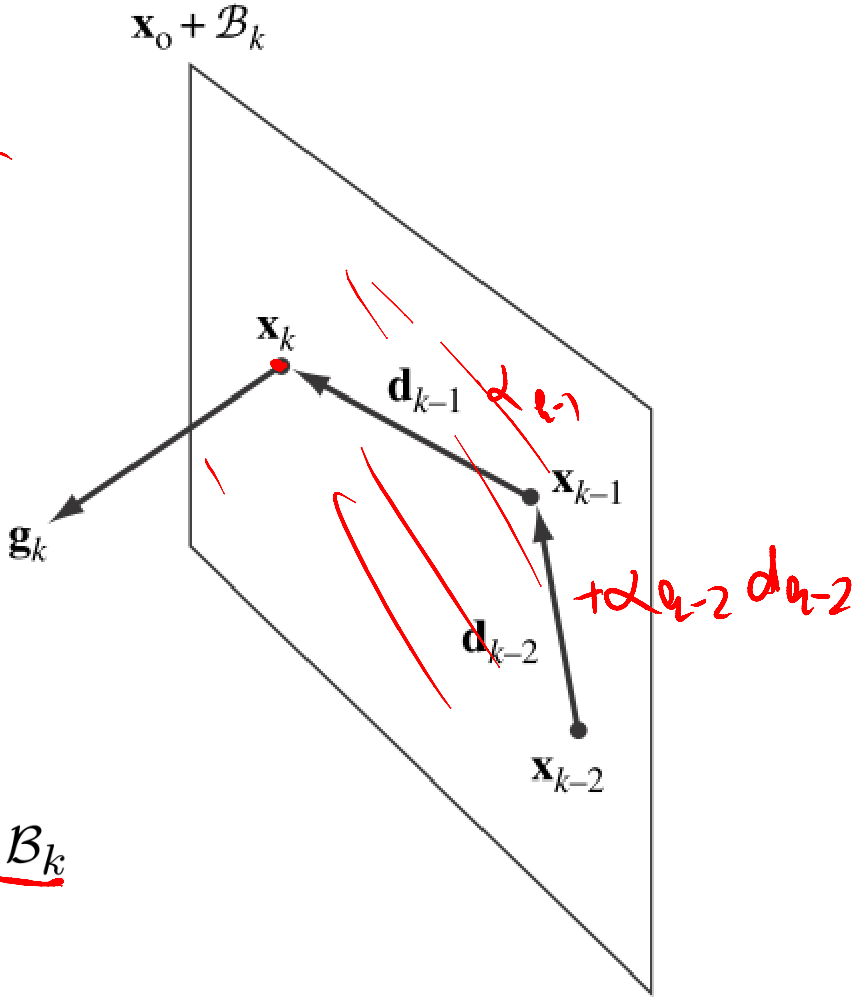
**Proof**

It is enough to show that $x_k$ minimizes $f$ on $x = x_0 + \mathcal{B}_k$ since it contains the line:

$x = x_{k-1} + \alpha d_{k-1}$

[By the definition of $\mathcal{B}_k$]

$f'(x_k) = g_k = Q x_k - b$



Since $f$ is stricly convex, it is enough to show that $g_k = f'(x_k)$ is orthogonal to $\mathcal{B}_k$

We prove $g_k \perp \mathcal{B}_k$ by induction.

# Proof

We prove $g_k \perp \mathcal{B}_k$ by induction.

$k = 0$ : $\mathcal{B}_0$ is empty set.

Assume that $g_k \perp \mathcal{B}_k$, and prove that $g_{k+1} \perp \mathcal{B}_{k+1}$

By definition,

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

$$g_k = Qx_k - b$$

Therefore,

$$\begin{aligned} g_{k+1} &= Qx_{k+1} - b = Q(x_k + \alpha_k d_k) - b \\ &= (Qx_k - b) + \alpha_k Q d_k = g_k + \alpha_k Q d_k \end{aligned}$$

# Proof

First let us prove that $g_{k+1} \perp d_k$.

$d_i \quad i<k$

We have proved

- $g_{k+1} = g_k + \alpha_k Q d_k$

By definition,

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

Therefore,

$$d_k^T g_{k+1} = d_k^T (g_k + \alpha_k Q d_k) = d_k^T g_k - d_k^T g_k = 0$$

$$g_{k+1} \perp d_k$$

# Proof

Now let us prove that $\underline{g_{k+1}} \perp d_i,\ \underline{i < k}.$

Since

- $g_{k+1} = g_k + \alpha_k Q d_k$ [We have proved this]
- $g_k \perp \underline{\mathcal{B}_k} = \text{span}(d_0, \ldots, d_{k-1})$

[By induction assumption],

Therefore,

$$\underline{d_i^T g_{k+1}} = \underline{d_i^T (g_k + \alpha_k Q d_k)} = \underline{d_i^T g_k} + \alpha_k \underline{d_i^T Q d_k} = 0$$

$$g_{k+1} \perp d_i,\ \forall i < k$$

We have proved $\underline{g_{k+1} \perp \mathcal{B}_{k+1}}$ where $\mathcal{B}_{k+1} = \text{span}(d_0, \ldots, d_k)$

Q.E.D.

# Corollary of Exp. Subs. Theorem

**Corollary**

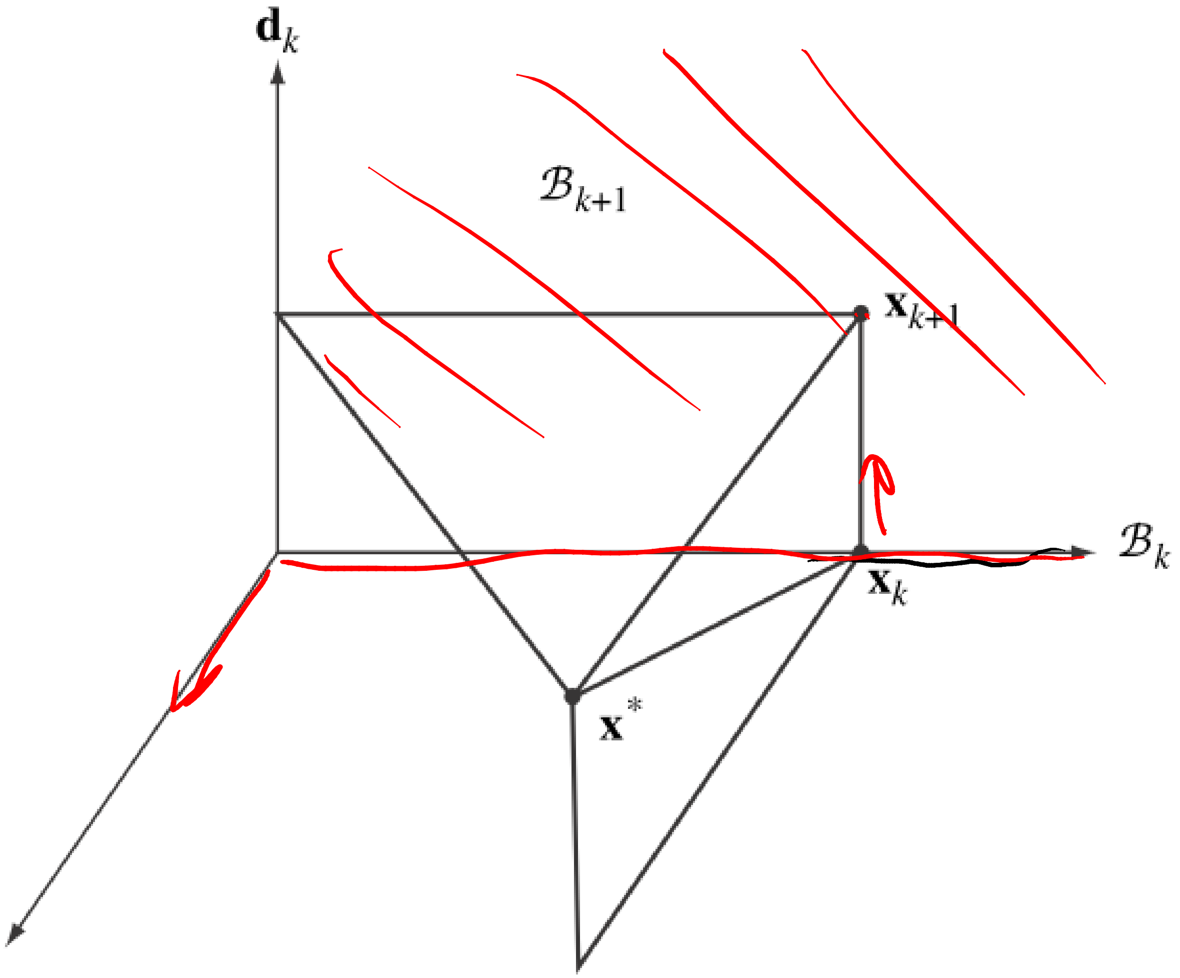
$$g_k \perp \mathcal{B}_k = \text{span}(\underline{d_0}, \dots, \underline{d_{k-1}})$$

$$\underline{g_k^T d_i} = 0 \ \forall 0 \leq i < k$$

$$\emptyset = \mathcal{B}_0 \subset \dots \subset \mathcal{B}_k \subset \underline{\mathcal{B}_n} = \mathbb{R}^n$$

Since $\underline{x_k}$ minimizes $\underline{f}$ over $\underline{x_0 + \mathcal{B}_k}$,
$\Rightarrow$ $\underline{x_n}$ is the minimum of $\underline{f}$ in $\underline{\mathbb{R}^n}$.

# Expanding Subspace Theorem



# THE CONJUGATE GRADIENT METHOD

# THE CONJUGATE GRADIENT METHOD

Given $d_0,\dots, d_{n-1}$, we already have an update rule for $\alpha_k$

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

How should we choose vectors $d_0,\dots, d_{n-1}$?

**The conjugate gradient method**

- The conjugate gradient method is a conjugate direction method
- Selects the successive direction vectors as a conjugate version of the successive gradients obtained as the method progresses.
- The conjugate directions are not specified beforehand, but rather are determined sequentially at each step of the iteration.

# THE CONJUGATE GRADIENT METHOD

**Advantages**

❑ Simple update rule

❑ the directions are based on the gradients, therefore the process makes good uniform progress toward the solution at every step.

For arbitrary sequences of conjugate directions the progress may be slight until the final few steps

# Conjugate Gradient Algorithm

$$\left.\begin{aligned}
&\text{Let } \underline{x_0} \in \mathbb{R}^n \text{ be arbitrary.}\\
&\underline{d_0} = -g_0 = \underline{b - Qx_0}\\
&\underline{\alpha_k} = -\frac{g_k^T d_k}{d_k^T Q d_k}\\
&x_{k+1} = \underline{x_k + \alpha_k d_k}\\
&g_k = Qx_k - b\\
&d_{k+1} = -\underline{g_{k+1}} + \underline{\beta_k d_k}\\
&\underline{\beta_k = \frac{g_{k+1}^T Q d_k}{d_k^T Q d_k}}
\end{aligned}\right\} \text{Conjugate Gradient Algorithm}$$

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

$Qx - b$

- ❑ The CGA is only slightly more complicated to implement than the method of steepest descent but converges in a finite number of steps on quadratic problems.
- ❑ In contrast to Newton method, there is no need for matrix inversion.

# Conjugate Gradient Theorem

To verify that the algorithm is a conjugate direction algorithm, all we need is to verify that the vectors $d_0,\ldots,d_k$ are Q-orthogonal.

**Theorem** [Conjugate Gradient Theorem]

The conjugate gradient algorithm is a conjugate direction method.

a) $\text{span}(g_0, g_1, \ldots, g_k) = \text{span}(g_0, Qg_0, \ldots, Q^k g_0)$

b) $\text{span}(d_0, d_1, \ldots, d_k) = \text{span}(g_0, Qg_0, \ldots, Q^k g_0)$

c) $d_k^T Q d_i = 0, \ \forall i < k$

d) $\alpha_k = \frac{g_k^T g_k}{d_k^T Q d_k}$

Only $\alpha_k$ needs matrix Q in the algorithm!

e) $\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$

# Proofs

Would be nice to discuss it, but no time…

# EXTENSION TO NONQUADRATIC PROBLEMS

# EXTENSION TO NONQUADRATIC PROBLEMS

**Goal:** $$\min_{x \in \mathbb{R}^n} \underline{f(x)}$$

**Do quadratic approximation**

$$\underline{g_k} = \nabla f(x_k) \quad \underline{Q\ } = \nabla^2 f(x_k)$$

This is similar to Newton’s method.
[f is approximated by a quadratic function]

- ❑ When applied to nonquadratic problems, conjugate gradient methods will not usually terminate within n steps.
- ❑ After n steps, we can restart the process from this point and run the algorithm for another n steps…

# Conjugate Gradient Algorithm for nonquadratic functions

**Step 1**

Starting at $x_0$ compute $g_0 = \nabla f(x_0)$ and set $d_0 = -g_0$.

**Step 2**

For $k = 0, 1, \ldots, n-1$ :

a) Set $x_{k+1} = \alpha_k d_k$ where $\alpha_k = \underline{-\frac{g_k^T d_k}{d_k^T [\nabla^2 f(x_k)] d_k}}$

b) Compute $g_{k+1} = \nabla f(x_{k+1})$

c) Unless $k = n-1$, set $\underline{d_{k+1}} = -g_{k+1} + \beta d_k$ where

$$\underline{\beta_k = \frac{\overline{g_{k+1}^T [\nabla^2 f(x_k)]} d_k}{d_k^T [\nabla^2 f(x_k)] d_k}}$$

End for

**Step 3**

Replace $x_0$ by $x_n$ and go back to Step 1.

# Properties of CGA

- An attractive feature of the algorithm is that, just as in the pure form of Newton’s method, no line searching is required at any stage.

- The algorithm converges in a finite number of steps for a quadratic problem.

- The undesirable features are  that Hessian matrix must be evaluated at each point. The algorithm is not, in this form, globally convergent.

# LINE SEARCH METHODS

# Fletcher–Reeves method

**Step 1**

Starting at $x_0$ compute $g_0 = \nabla f(x_0)$ and set $d_0 = -g_0$.

**Step 2**

For $k = 0, 1, \ldots, n-1$ : $\alpha_k = \arg\min_\alpha f(x_k + \alpha d_k)$

a) Set $x_{k+1} = x_k + \alpha_k d_k$ where

b) Compute $g_{k+1} = \nabla f(x_{k+1})$

c) Unless $k = n-1$, set $d_{k+1} = -g_{k+1} + \beta d_k$ where

$$\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^t g_k}$$

End for

**Step 3**

Replace $x_0$ by $x_n$ and go back to Step 1.

# Fletcher–Reeves method

Line search method

Hessian is not used in the algorithm

In the quadratic case it is identical to the original conjugate direction algorithm

# Polak–Ribiere method

Same as Fletcher–Reeves method, BUT:

$$\beta_k = \frac{(g_{k+1} - g_k)^T g_{k+1}}{g_k^T g_k}$$

Again this leads to a value identical to the standard formula in the quadratic case.

Experimental evidence seems to favor the Polak–Ribiere method over other methods of this general type.

# Convergence rate

Under some conditions the line search method is globally convergent.

Under some conditions, the rate is

$$\|x_{k+n} - x^*\| \leq c\|x_k - x^*\|^2$$

[since one complete n step cycle solves a quadratic problem similarly

To the Newton method]

# Numerical Experiments

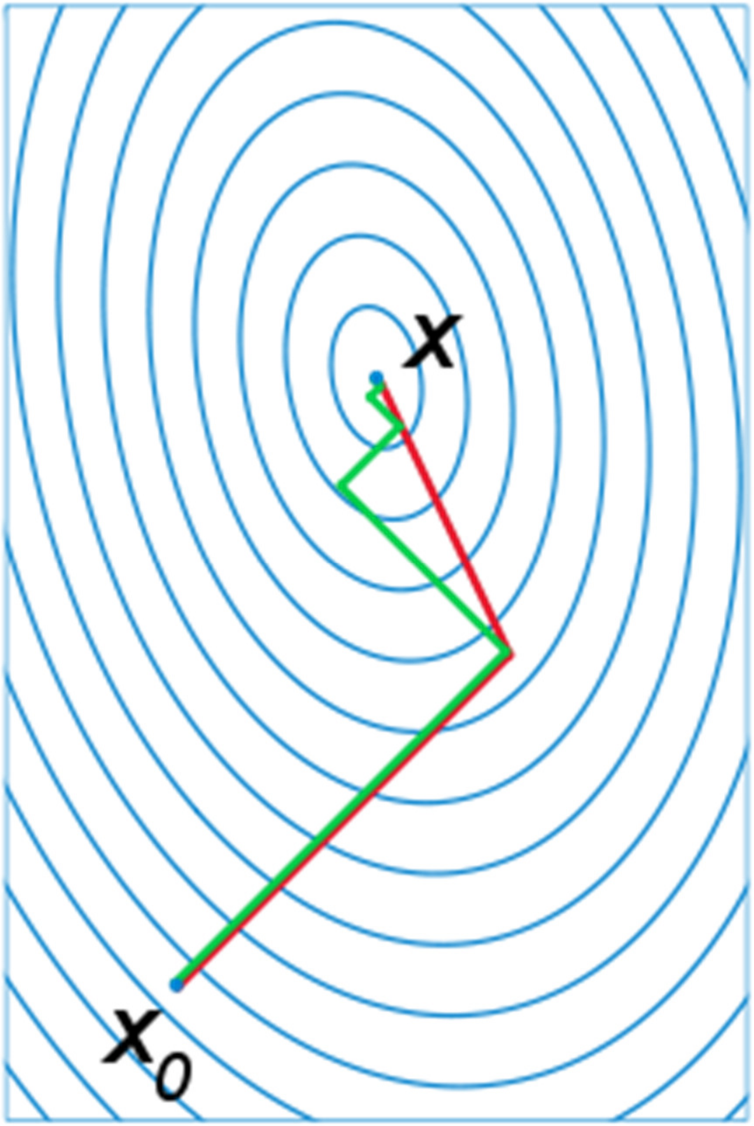
A comparison of

* gradient descent with optimal step size (in green) and

* conjugate vector (in red)

for minimizing a quadratic function.

Conjugate gradient converges in at most n steps (here n=2).



# Summary

❑ Conjugate Direction Methods

- conjugate directions

❑ Minimizing quadratic functions

❑ Conjugate Gradient Methods for nonquadratic functions

- Line search methods
    * Fletcher–Reeves
    * Polak–Ribiere