Conjugate gradients method

Daniil Merkulov

Optimization for ML. Faculty of Computer Science. HSE University

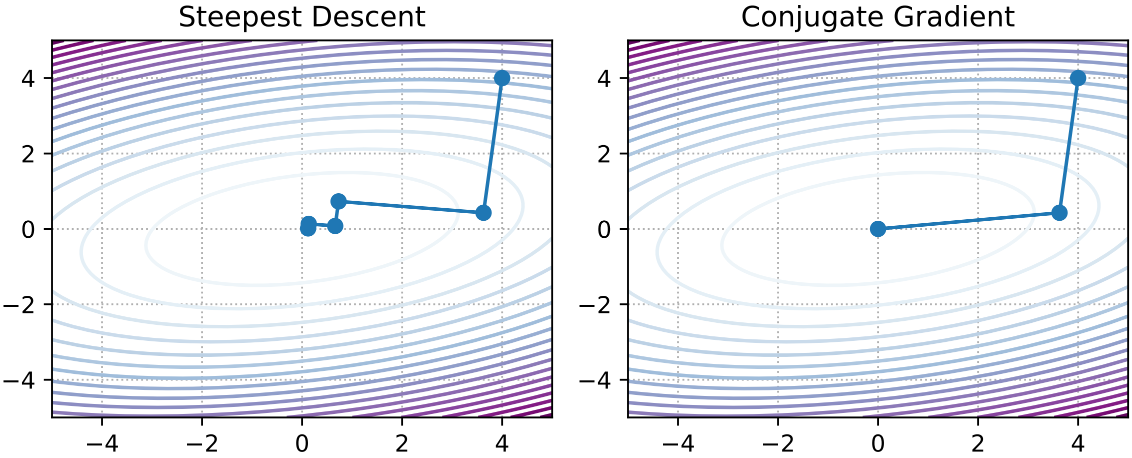
Strongly convex quadratics

Consider the following quadratic optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x) = \min_{x \in \mathbb{R}^n} \frac{1}{2} x^\top A x - b^\top x + c, \text{ where } A \in \mathbb{S}^n_{++}. \quad (1)$$

Optimality conditions

$$Ax^* = b$$



Exact line search aka steepest descent

$$\alpha_k = \text{arg}\min_{\alpha \in \mathbb{R}^+} f(x_{k+1}) = \text{arg}\min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

More theoretical than practical approach. It also allows you to analyze the convergence, but often exact line search can be difficult if the function calculation takes too long or costs a lot.

An interesting theoretical property of this method is that each following iteration is orthogonal to the previous one:

$$\alpha_k = \text{arg}\min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

Exact line search aka steepest descent

$$\alpha_k = \text{arg}\min_{\alpha \in \mathbb{R}^+} f(x_{k+1}) = \text{arg}\min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

More theoretical than practical approach. It also allows you to analyze the convergence, but often exact line search can be difficult if the function calculation takes too long or costs a lot. An interesting theoretical property of this method is that each following iteration is orthogonal to the previous one:

$$\alpha_k = \text{arg}\min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

Optimality conditions:

Exact line search aka steepest descent

$$\alpha_k = \text{arg}\min_{\alpha \in \mathbb{R}^+} f(x_{k+1}) = \text{arg}\min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

More theoretical than practical approach. It also allows you to analyze the convergence, but often exact line search can be difficult if the function calculation takes too long or costs a lot. An interesting theoretical property of this method is that each following iteration is orthogonal to the previous one:

$$\alpha_k = \text{arg}\min_{\alpha \in \mathbb{R}^+} f(x_k - \alpha \nabla f(x_k))$$

Optimality conditions:

$$\nabla f(x_k)^T \nabla f(x_{k+1}) = 0$$

Optimal value for quadratics

$$\nabla f(x_k)^\top A(x_k - \alpha \nabla f(x_k)) - \nabla f(x_k)^\top b = 0 \qquad \alpha_k = \frac{\nabla f(x_k)^T \nabla f(x_k)}{\nabla f(x_k)^T A \nabla f(x_k)}$$

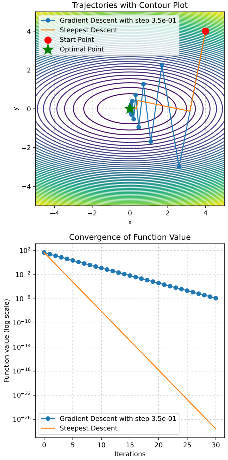
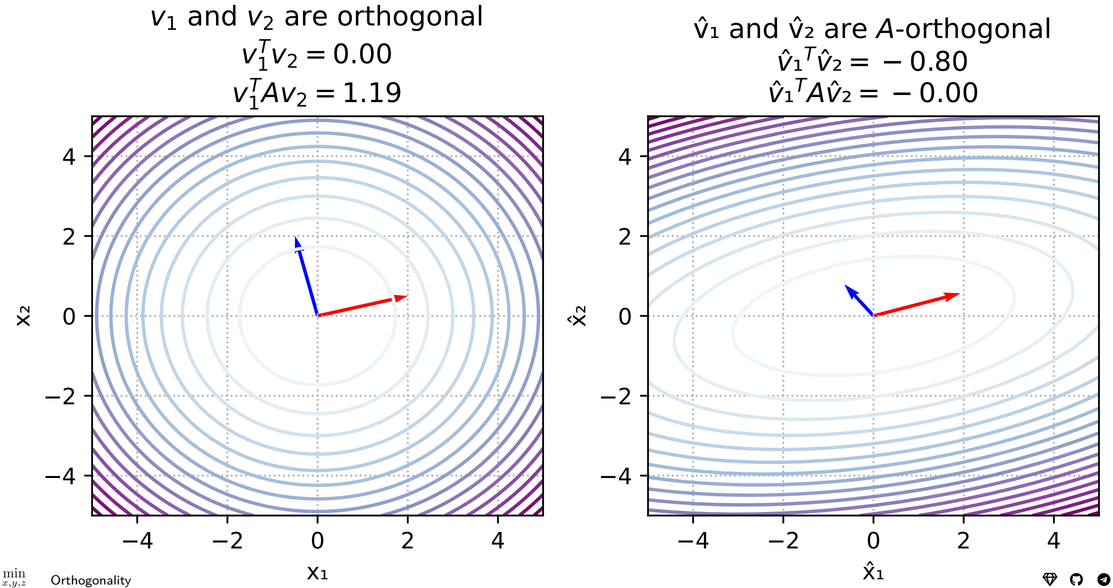


Figure 1: Steepest Descent

Open In Colab ♣

Conjugate directions. A-orthogonality.



Conjugate directions. A-orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T I x$ looks just like on the left part of Figure 2, while in other coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A \hat{x}$, where $A \in \mathbb{S}^n_{++}$.

$$\frac{1}{2}x^T I x \qquad\qquad \frac{1}{2}\hat{x}^T A \hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A \hat{x}$$

Conjugate directions. A-orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T I x$ looks just like on the left part of Figure 2, while in other coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A \hat{x}$, where $A \in \mathbb{S}^n_{++}$.

$$\frac{1}{2}x^T I x \qquad\qquad \frac{1}{2}\hat{x}^T A \hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x}$$

Conjugate directions. A-orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T I x$ looks just like on the left part of Figure 2, while in other coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A \hat{x}$, where $A \in \mathbb{S}^n_{++}$.

$$\frac{1}{2}x^T I x \qquad\qquad \frac{1}{2}\hat{x}^T A \hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}Q^T \hat{x}$$

Conjugate directions. A-orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T I x$ looks just like on the left part of Figure 2, while in other coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A \hat{x}$, where $A \in \mathbb{S}^n_{++}$.

$$\frac{1}{2}x^T I x \qquad\qquad \frac{1}{2}\hat{x}^T A \hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}} Q^T \hat{x} = \frac{1}{2}x^T I x$$

Conjugate directions. A-orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T Ix$ looks just like on the left part of Figure 2, while in other coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A\hat{x}$, where $A \in \mathbb{S}^n_{++}$.

$$\frac{1}{2}x^T Ix \qquad\qquad \frac{1}{2}\hat{x}^T A\hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A\hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T\hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}Q^T\hat{x} = \frac{1}{2}x^T Ix \qquad \text{if } x = \Lambda^{\frac{1}{2}}Q^T\hat{x}$$

Conjugate directions. A-orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T I x$ looks just like on the left part of Figure 2, while in other coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A \hat{x}$, where $A \in \mathbb{S}^n_{++}$.

$$\frac{1}{2}x^T I x \qquad\qquad \frac{1}{2}\hat{x}^T A \hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}} Q^T \hat{x} = \frac{1}{2}x^T I x \qquad \text{if } x = \Lambda^{\frac{1}{2}} Q^T \hat{x} \text{ and } \hat{x} = Q\Lambda^{-\frac{1}{2}} x$$

Conjugate directions. A-orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T I x$ looks just like on the left part of Figure 2, while in other coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A \hat{x}$, where $A \in \mathbb{S}^n_{++}$.

$$\frac{1}{2}x^T I x \qquad\qquad \frac{1}{2}\hat{x}^T A \hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}Q^T \hat{x} = \frac{1}{2}x^T I x \qquad \text{if } x = \Lambda^{\frac{1}{2}}Q^T \hat{x} \text{ and } \hat{x} = Q\Lambda^{-\frac{1}{2}}x$$

Conjugate directions. A-orthogonality.

Suppose, we have two coordinate systems and some quadratic function $f(x) = \frac{1}{2}x^T I x$ looks just like on the left part of Figure 2, while in other coordinates it looks like $f(\hat{x}) = \frac{1}{2}\hat{x}^T A \hat{x}$, where $A \in \mathbb{S}^n_{++}$.

$$\frac{1}{2}x^T I x \qquad\qquad \frac{1}{2}\hat{x}^T A \hat{x}$$

Since $A = Q\Lambda Q^T$:

$$\frac{1}{2}\hat{x}^T A \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda Q^T \hat{x} = \frac{1}{2}\hat{x}^T Q\Lambda^{\frac{1}{2}}\Lambda^{\frac{1}{2}}Q^T \hat{x} = \frac{1}{2}x^T I x \qquad \text{if } x = \Lambda^{\frac{1}{2}}Q^T \hat{x} \text{ and } \hat{x} = Q\Lambda^{-\frac{1}{2}}x$$

A-orthogonal vectors

Vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$ are called A-orthogonal (or A-conjugate) if

$$x^T A y = 0 \qquad \Leftrightarrow \qquad x \perp_A y$$

When $A = I$, A-orthogonality becomes orthogonality.

Gram–Schmidt process

Input: n linearly independent vectors $u_0, \dots, u_{n-1}$.

Output: n linearly independent vectors, which are pairwise orthogonal $d_0, \dots, d_{n-1}$.

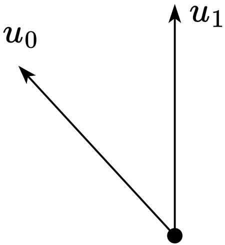


Figure 3: Illustration of Gram-Schmidt orthogonalization process

Gram–Schmidt process

Input: n linearly independent vectors $u_0, \ldots, u_{n-1}$.

Output: n linearly independent vectors, which are pairwise orthogonal $d_0, \ldots, d_{n-1}$.

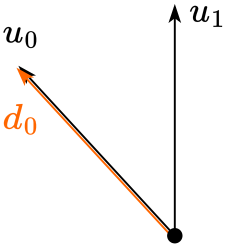


Figure 4: Illustration of Gram-Schmidt orthogonalization process

Gram–Schmidt process

Input: n linearly independent vectors $u_0, \ldots, u_{n-1}$.

Output: n linearly independent vectors, which are pairwise orthogonal $d_0, \ldots, d_{n-1}$.

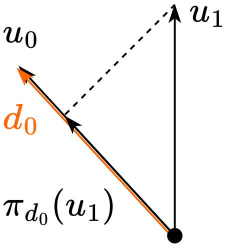


Figure 5: Illustration of Gram-Schmidt orthogonalization process

Gram–Schmidt process

Input: n linearly independent vectors $u_0, \ldots, u_{n-1}$.

Output: n linearly independent vectors, which are pairwise orthogonal $d_0, \ldots, d_{n-1}$.

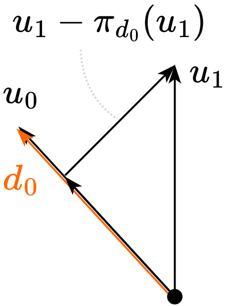


Figure 6: Illustration of Gram-Schmidt orthogonalization process

Gram–Schmidt process

Input: n linearly independent vectors $u_0, \ldots, u_{n-1}$.

Output: n linearly independent vectors, which are pairwise orthogonal $d_0, \ldots, d_{n-1}$.

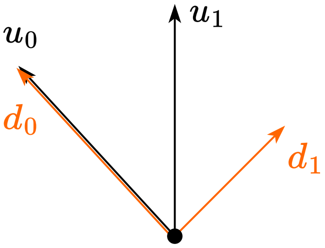
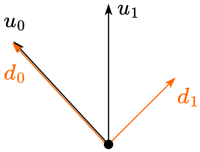
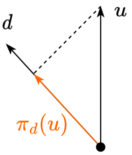


Figure 7: Illustration of Gram-Schmidt orthogonalization process

Gram–Schmidt process

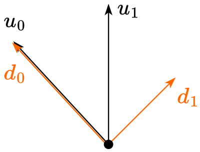
Input: n linearly independent vectors $u_0, \dots, u_{n-1}$.

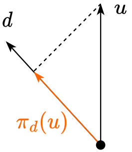




$$\pi_d(u) = \frac{\langle d, u \rangle}{\|d\|_2^2} d$$

Gram–Schmidt process



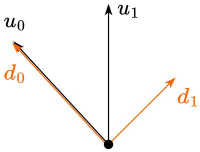


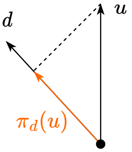
$$\pi_d(u) = \frac{\langle d, u \rangle}{\|d\|_2^2} d$$

Input: n linearly independent vectors $u_0, \ldots, u_{n-1}$.
Output: n linearly independent vectors, which are pairwise orthogonal $d_0, \ldots, d_{n-1}$.

$$d_0 = u_0$$

Gram–Schmidt process



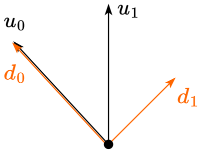


$$\pi_d(u) = \frac{\langle d, u\rangle}{\|d\|_2^2} d$$

Input: n linearly independent vectors $u_0, \ldots, u_{n-1}$.
Output: n linearly independent vectors, which are pairwise orthogonal $d_0, \ldots, d_{n-1}$.

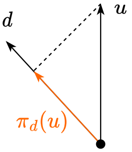
$$d_0 = u_0$$
$$d_1 = u_1 - \pi_{d_0}(u_1)$$

Gram–Schmidt process



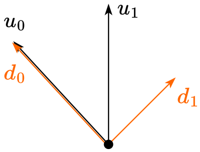
Input: n linearly independent vectors $u_0, \ldots, u_{n-1}$.
Output: n linearly independent vectors, which are pairwise orthogonal $d_0, \ldots, d_{n-1}$.

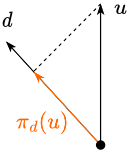
$$
\begin{aligned}
d_0 &= u_0 \\
d_1 &= u_1 - \pi_{d_0}(u_1) \\
d_2 &= u_2 - \pi_{d_0}(u_2) - \pi_{d_1}(u_2)
\end{aligned}
$$



$$
\pi_d(u) = \frac{\langle d, u \rangle}{\|d\|_2^2} d
$$

Gram–Schmidt process



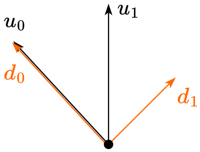


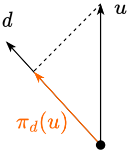
$$\pi_d(u) = \frac{\langle d, u \rangle}{\|d\|_2^2} d$$

Input: n linearly independent vectors $u_0, \ldots, u_{n-1}$.
Output: n linearly independent vectors, which are pairwise orthogonal $d_0, \ldots, d_{n-1}$.

$$
\begin{aligned}
d_0 &= u_0 \\
d_1 &= u_1 - \pi_{d_0}(u_1) \\
d_2 &= u_2 - \pi_{d_0}(u_2) - \pi_{d_1}(u_2) \\
&\vdots
\end{aligned}
$$

Gram–Schmidt process





$$\pi_d(u) = \frac{\langle d, u\rangle}{\|d\|_2^2} d$$

Input: n linearly independent vectors $u_0, \ldots, u_{n-1}$.
Output: n linearly independent vectors, which are pairwise orthogonal $d_0, \ldots, d_{n-1}$.

$$d_0 = u_0$$

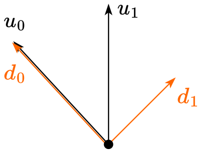
$$d_1 = u_1 - \pi_{d_0}(u_1)$$

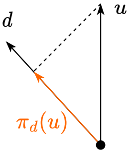
$$d_2 = u_2 - \pi_{d_0}(u_2) - \pi_{d_1}(u_2)$$

$$\vdots$$

$$d_k = u_k - \sum_{i=0}^{k-1} \pi_{d_i}(u_k)$$

Gram–Schmidt process



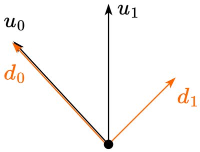


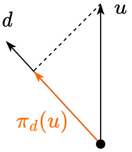
$$\pi_d(u) = \frac{\langle d, u\rangle}{\|d\|_2^2} d$$

Input: n linearly independent vectors $u_0, \ldots, u_{n-1}$.
Output: n linearly independent vectors, which are pairwise orthogonal $d_0, \ldots, d_{n-1}$.

$$d_0 = u_0$$
$$d_1 = u_1 - \pi_{d_0}(u_1)$$
$$d_2 = u_2 - \pi_{d_0}(u_2) - \pi_{d_1}(u_2)$$
$$\vdots$$
$$d_k = u_k - \sum_{i=0}^{k-1} \pi_{d_i}(u_k)$$

Gram–Schmidt process





$$\pi_d(u) = \frac{\langle d, u\rangle}{\|d\|_2^2} d$$

Input: n linearly independent vectors $u_0, \ldots, u_{n-1}$.
Output: n linearly independent vectors, which are pairwise orthogonal $d_0, \ldots, d_{n-1}$.

$$\begin{aligned} d_0 &= u_0 \\ d_1 &= u_1 - \pi_{d_0}(u_1) \\ d_2 &= u_2 - \pi_{d_0}(u_2) - \pi_{d_1}(u_2) \\ &\vdots \\ d_k &= u_k - \sum_{i=0}^{k-1} \pi_{d_i}(u_k) \end{aligned}$$

$$d_k = u_k + \sum_{i=0}^{k-1} \beta_{ik} d_i \qquad \beta_{ik} = -\frac{\langle d_i, u_k\rangle}{\langle d_i, d_i\rangle} \tag{2}$$

General idea

- In an isotropic $A = I$ world, the steepest descent starting from an arbitrary point in any n orthogonal linearly independent directions will converge in n steps in exact arithmetic. We attempt to construct the same procedure in the case $A \neq I$ using the concept of A-orthogonality.

General idea

- In an isotropic $A = I$ world, the steepest descent starting from an arbitrary point in any n orthogonal linearly independent directions will converge in n steps in exact arithmetic. We attempt to construct the same procedure in the case $A \neq I$ using the concept of A-orthogonality.
- Suppose, we have a set of n linearly independent A-orthogonal directions $d_0, \ldots, d_{n-1}$ (which will be computed with Gram-Schmidt process).

General idea

- In an isotropic $A = I$ world, the steepest descent starting from an arbitrary point in any n orthogonal linearly independent directions will converge in n steps in exact arithmetic. We attempt to construct the same procedure in the case $A \neq I$ using the concept of A-orthogonality.
- Suppose, we have a set of n linearly independent A-orthogonal directions $d_0, \ldots, d_{n-1}$ (which will be computed with Gram-Schmidt process).
- We would like to build a method, that goes from x_0 to the x^* for the quadratic problem with stepsizes α_i, which is, in fact, just the decomposition of $x^* - x_0$ to some basis:

$$x^* = x_0 + \sum_{i=0}^{n-1} \alpha_i d_i \qquad x^* - x_0 = \sum_{i=0}^{n-1} \alpha_i d_i$$

General idea

- In an isotropic $A = I$ world, the steepest descent starting from an arbitrary point in any n orthogonal linearly independent directions will converge in n steps in exact arithmetic. We attempt to construct the same procedure in the case $A \neq I$ using the concept of A-orthogonality.
- Suppose, we have a set of n linearly independent A-orthogonal directions $d_0, \ldots, d_{n-1}$ (which will be computed with Gram-Schmidt process).
- We would like to build a method, that goes from x_0 to the x^* for the quadratic problem with stepsizes α_i, which is, in fact, just the decomposition of $x^* - x_0$ to some basis:

$$x^* = x_0 + \sum_{i=0}^{n-1} \alpha_i d_i \qquad x^* - x_0 = \sum_{i=0}^{n-1} \alpha_i d_i$$

- We will prove, that α_i and d_i could be selected in a very efficient way (Conjugate Gradient method).

Idea of Conjugate Directions (CD) method

Thus, we formulate an algorithm:

1. Let $k = 0$ and $x_k = x_0$, count $d_k = d_0 = -\nabla f(x_0)$.

Idea of Conjugate Directions (CD) method

Thus, we formulate an algorithm:

1. Let $k = 0$ and $x_k = x_0$, count $d_k = d_0 = -\nabla f(x_0)$.
2. By the procedure of line search we find the optimal length of step. Calculate α minimizing $f(x_k + \alpha_k d_k)$ by the formula

$$\alpha_k = -\frac{d_k^\top (Ax_k - b)}{d_k^\top A d_k} \tag{3}$$

Idea of Conjugate Directions (CD) method

Thus, we formulate an algorithm:

1. Let $k = 0$ and $x_k = x_0$, count $d_k = d_0 = -\nabla f(x_0)$.
2. By the procedure of line search we find the optimal length of step. Calculate α minimizing $f(x_k + \alpha_k d_k)$ by the formula

$$\alpha_k = -\frac{d_k^\top (Ax_k - b)}{d_k^\top A d_k} \tag{3}$$

3. We're doing an algorithm step:

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

Idea of Conjugate Directions (CD) method

Thus, we formulate an algorithm:

1. Let $k = 0$ and $x_k = x_0$, count $d_k = d_0 = -\nabla f(x_0)$.
2. By the procedure of line search we find the optimal length of step. Calculate α minimizing $f(x_k + \alpha_k d_k)$ by the formula
$$\alpha_k = -\frac{d_k^\top (Ax_k - b)}{d_k^\top A d_k} \tag{3}$$
3. We're doing an algorithm step:
$$x_{k+1} = x_k + \alpha_k d_k$$
4. Update the direction: $d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k$ in order to make $d_{k+1} \perp_A d_k$, where β_k is calculated by the formula:
$$\beta_k = \frac{\nabla f(x_{k+1})^\top A d_k}{d_k^\top A d_k}.$$

Idea of Conjugate Directions (CD) method

Thus, we formulate an algorithm:

1. Let $k = 0$ and $x_k = x_0$, count $d_k = d_0 = -\nabla f(x_0)$.
2. By the procedure of line search we find the optimal length of step. Calculate α minimizing $f(x_k + \alpha_k d_k)$ by the formula
$$\alpha_k = -\frac{d_k^\top (Ax_k - b)}{d_k^\top A d_k} \tag{3}$$
3. We're doing an algorithm step:
$$x_{k+1} = x_k + \alpha_k d_k$$
4. Update the direction: $d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k$ in order to make $d_{k+1} \perp_A d_k$, where β_k is calculated by the formula:
$$\beta_k = \frac{\nabla f(x_{k+1})^\top A d_k}{d_k^\top A d_k}.$$
5. Repeat steps 2-4 until n directions are built, where n is the dimension of space (dimension of x).

Conjugate Directions (CD) method

Lemma 1. Linear independence of A-conjugate vectors.

If a set of vectors $d_1, \ldots, d_n$ - are A-conjugate (each pair of vectors is A-conjugate), these vectors are linearly independent. $A \in \mathbb{S}^n_{++}$.

Conjugate Directions (CD) method

Lemma 1. Linear independence of A-conjugate vectors.

If a set of vectors $d_1, \ldots, d_n$ - are A-conjugate (each pair of vectors is A-conjugate), these vectors are linearly independent. $A \in \mathbb{S}^n_{++}$.

Proof

We'll show, that if $\sum\limits_{i=1}^n \alpha_i d_i = 0$, than all coefficients should be equal to zero:

Conjugate Directions (CD) method

Lemma 1. Linear independence of A-conjugate vectors.

If a set of vectors $d_1, \ldots, d_n$ - are A-conjugate (each pair of vectors is A-conjugate), these vectors are linearly independent. $A \in \mathbb{S}^n_{++}$.

Proof

We'll show, that if $\sum\limits_{i=1}^n \alpha_i d_i = 0$, than all coefficients should be equal to zero:

$$
0 = \sum_{i=1}^n \alpha_i d_i
$$

Conjugate Directions (CD) method

Lemma 1. Linear independence of A-conjugate vectors.

If a set of vectors $d_1, \ldots, d_n$ - are A-conjugate (each pair of vectors is A-conjugate), these vectors are linearly independent. $A \in \mathbb{S}^n_{++}$.

Proof

We'll show, that if $\sum\limits_{i=1}^n \alpha_i d_i = 0$, than all coefficients should be equal to zero:

$$0 = \sum_{i=1}^n \alpha_i d_i$$

$$\text{Multiply by } d_j^T A \cdot \qquad = d_j^\top A \left(\sum_{i=1}^n \alpha_i d_i \right)$$

Conjugate Directions (CD) method

Lemma 1. Linear independence of A-conjugate vectors.

If a set of vectors $d_1, \ldots, d_n$ - are A-conjugate (each pair of vectors is A-conjugate), these vectors are linearly independent. $A \in \mathbb{S}^n_{++}$.

Proof

We'll show, that if $\sum\limits_{i=1}^n \alpha_i d_i = 0$, than all coefficients should be equal to zero:

$$
\begin{aligned}
0 &= \sum_{i=1}^n \alpha_i d_i \\
\text{Multiply by } d_j^T A \cdot \qquad &= d_j^\top A \left(\sum_{i=1}^n \alpha_i d_i \right) = \sum_{i=1}^n \alpha_i d_j^\top A d_i
\end{aligned}
$$

Conjugate Directions (CD) method

Lemma 1. Linear independence of A-conjugate vectors.

If a set of vectors $d_1, \ldots, d_n$ - are A-conjugate (each pair of vectors is A-conjugate), these vectors are linearly independent. $A \in \mathbb{S}^n_{++}$.

Proof

We'll show, that if $\sum\limits_{i=1}^n \alpha_i d_i = 0$, than all coefficients should be equal to zero:

$$0 = \sum_{i=1}^n \alpha_i d_i$$

$$\text{Multiply by } d_j^T A \cdot \qquad = d_j^\top A \left(\sum_{i=1}^n \alpha_i d_i \right) = \sum_{i=1}^n \alpha_i d_j^\top A d_i$$

$$= \alpha_j d_j^\top A d_j + 0 + \ldots + 0$$

Conjugate Directions (CD) method

Lemma 1. Linear independence of A-conjugate vectors.

If a set of vectors $d_1, \ldots, d_n$ - are A-conjugate (each pair of vectors is A-conjugate), these vectors are linearly independent. $A \in \mathbb{S}^n_{++}$.

Proof

We'll show, that if $\sum\limits_{i=1}^n \alpha_i d_i = 0$, than all coefficients should be equal to zero:

$$
\begin{aligned}
0 &= \sum_{i=1}^n \alpha_i d_i \\
\text{Multiply by } d_j^T A \cdot \qquad &= d_j^\top A \left(\sum_{i=1}^n \alpha_i d_i \right) = \sum_{i=1}^n \alpha_i d_j^\top A d_i \\
&= \alpha_j d_j^\top A d_j + 0 + \ldots + 0
\end{aligned}
$$

Conjugate Directions (CD) method

Lemma 1. Linear independence of A-conjugate vectors.

If a set of vectors $d_1, \ldots, d_n$ - are A-conjugate (each pair of vectors is A-conjugate), these vectors are linearly independent. $A \in \mathbb{S}^n_{++}$.

Proof

We'll show, that if $\sum\limits_{i=1}^n \alpha_i d_i = 0$, than all coefficients should be equal to zero:

$$
\begin{aligned}
0 &= \sum_{i=1}^n \alpha_i d_i \\
\text{Multiply by } d_j^T A \cdot \qquad &= d_j^\top A \left(\sum_{i=1}^n \alpha_i d_i \right) = \sum_{i=1}^n \alpha_i d_j^\top A d_i \\
&= \alpha_j d_j^\top A d_j + 0 + \ldots + 0
\end{aligned}
$$

Thus, $\alpha_j = 0$, for all other indices one has to perform the same process

Proof of convergence

We will introduce the following notation:

- $r_k = b - Ax_k$ - residual,

Proof of convergence

We will introduce the following notation:

- $r_k = b - Ax_k$ - residual,
- $e_k = x_k - x^*$ - error.

Proof of convergence

We will introduce the following notation:

- $r_k = b - Ax_k$ - residual,
- $e_k = x_k - x^*$ - error.
- Since $Ax^* = b$, we have $r_k = b - Ax_k = Ax^* - Ax_k = -A(x_k - x^*)$

$$r_k = -Ae_k. \tag{4}$$

Proof of convergence

We will introduce the following notation:

- $r_k = b - Ax_k$ - residual,
- $e_k = x_k - x^*$ - error.
- Since $Ax^* = b$, we have $r_k = b - Ax_k = Ax^* - Ax_k = -A(x_k - x^*)$

$$r_k = -Ae_k. \tag{4}$$

- Note also, that since $x_{k+1} = x_0 + \sum\limits_{i=1}^{k} \alpha_i d_i$, we have

$$e_{k+1} = e_0 + \sum_{i=1}^{k} \alpha_i d_i. \tag{5}$$

Proof of convergence

Lemma 2. Convergence of conjugate direction method.

Suppose, we solve n-dimensional quadratic convex optimization problem (1). The conjugate directions method

$$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$

with $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, Ad_i \rangle}$ taken from the line search, converges for at most n steps of the algorithm.

Proof of convergence

Lemma 2. Convergence of conjugate direction method.

Suppose, we solve n-dimensional quadratic convex optimization problem (1). The conjugate directions method

$$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$

with $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, Ad_i \rangle}$ taken from the line search, converges for at most n steps of the algorithm.

Proof

1. We need to prove, that $\delta_i = -\alpha_i$:

$$e_0 = x_0 - x^* = \sum_{i=0}^{n-1} \delta_i d_i$$

Proof of convergence

Lemma 2. Convergence of conjugate direction method.

Suppose, we solve n-dimensional quadratic convex optimization problem (1). The conjugate directions method

$$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$

with $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, A d_i \rangle}$ taken from the line search, converges for at most n steps of the algorithm.

Proof

1. We need to prove, that $\delta_i = -\alpha_i$:

$$e_0 = x_0 - x^* = \sum_{i=0}^{n-1} \delta_i d_i$$

Proof of convergence

Lemma 2. Convergence of conjugate direction method.

Suppose, we solve n-dimensional quadratic convex optimization problem (1). The conjugate directions method

$$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$

with $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, Ad_i \rangle}$ taken from the line search, converges for at most n steps of the algorithm.

Proof

1. We need to prove, that $\delta_i = -\alpha_i$:

$$e_0 = x_0 - x^* = \sum_{i=0}^{n-1} \delta_i d_i$$

2. We multiply both hand sides from the left by $d_k^T A$:

Proof of convergence

Lemma 2. Convergence of conjugate direction method.

Suppose, we solve n-dimensional quadratic convex optimization problem (1). The conjugate directions method

$$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$

with $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, Ad_i \rangle}$ taken from the line search, converges for at most n steps of the algorithm.

Proof

1. We need to prove, that $\delta_i = -\alpha_i$:

$$e_0 = x_0 - x^* = \sum_{i=0}^{n-1} \delta_i d_i$$

2. We multiply both hand sides from the left by $d_k^T A$:

$$d_k^T A e_0 = \sum_{i=0}^{n-1} \delta_i d_k^T A d_i$$

Proof of convergence

Lemma 2. Convergence of conjugate direction method.

Suppose, we solve n-dimensional quadratic convex optimization problem (1). The conjugate directions method

$$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$

with $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, Ad_i \rangle}$ taken from the line search, converges for at most n steps of the algorithm.

Proof

1. We need to prove, that $\delta_i = -\alpha_i$:

$$e_0 = x_0 - x^* = \sum_{i=0}^{n-1} \delta_i d_i$$

2. We multiply both hand sides from the left by $d_k^T A$:

$$d_k^T A e_0 = \sum_{i=0}^{n-1} \delta_i d_k^T A d_i = \delta_k d_k^T A d_k$$

Proof of convergence

Lemma 2. Convergence of conjugate direction method.

Suppose, we solve n-dimensional quadratic convex optimization problem (1). The conjugate directions method

$$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$

with $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, Ad_i \rangle}$ taken from the line search, converges for at most n steps of the algorithm.

Proof

1. We need to prove, that $\delta_i = -\alpha_i$:

$$e_0 = x_0 - x^* = \sum_{i=0}^{n-1} \delta_i d_i$$

2. We multiply both hand sides from the left by $d_k^T A$:

$$d_k^T A e_0 = \sum_{i=0}^{n-1} \delta_i d_k^T A d_i = \delta_k d_k^T A d_k$$

$$d_k^T A \left(e_0 + \sum_{i=0}^{k-1} \alpha_i d_i \right)$$

Proof of convergence

Lemma 2. Convergence of conjugate direction method.

Suppose, we solve n-dimensional quadratic convex optimization problem (1). The conjugate directions method

$$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$

with $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, Ad_i \rangle}$ taken from the line search, converges for at most n steps of the algorithm.

Proof

1. We need to prove, that $\delta_i = -\alpha_i$:

$$e_0 = x_0 - x^* = \sum_{i=0}^{n-1} \delta_i d_i$$

2. We multiply both hand sides from the left by $d_k^T A$:

$$d_k^T A e_0 = \sum_{i=0}^{n-1} \delta_i d_k^T A d_i = \delta_k d_k^T A d_k$$

$$d_k^T A \left(e_0 + \sum_{i=0}^{k-1} \alpha_i d_i \right) = d_k^T A e_k$$

Proof of convergence

Lemma 2. Convergence of conjugate direction method.

Suppose, we solve n-dimensional quadratic convex optimization problem (1). The conjugate directions method

$$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$

with $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, A d_i \rangle}$ taken from the line search, converges for at most n steps of the algorithm.

Proof

1. We need to prove, that $\delta_i = -\alpha_i$:

$$e_0 = x_0 - x^* = \sum_{i=0}^{n-1} \delta_i d_i$$

2. We multiply both hand sides from the left by $d_k^T A$:

$$d_k^T A e_0 = \sum_{i=0}^{n-1} \delta_i d_k^T A d_i = \delta_k d_k^T A d_k$$

$$d_k^T A \left(e_0 + \sum_{i=0}^{k-1} \alpha_i d_i \right) = d_k^T A e_k = \delta_k d_k^T A d_k \quad (A - \text{ orthogonality})$$

Proof of convergence

Lemma 2. Convergence of conjugate direction method.

Suppose, we solve n-dimensional quadratic convex optimization problem (1). The conjugate directions method

$$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$

with $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, Ad_i \rangle}$ taken from the line search, converges for at most n steps of the algorithm.

Proof

1. We need to prove, that $\delta_i = -\alpha_i$:

$$e_0 = x_0 - x^* = \sum_{i=0}^{n-1} \delta_i d_i$$

2. We multiply both hand sides from the left by $d_k^T A$:

$$d_k^T A e_0 = \sum_{i=0}^{n-1} \delta_i d_k^T A d_i = \delta_k d_k^T A d_k$$

$$d_k^T A \left(e_0 + \sum_{i=0}^{k-1} \alpha_i d_i \right) = d_k^T A e_k = \delta_k d_k^T A d_k \quad (A - \text{ orthogonality})$$

$$\delta_k = \frac{d_k^T A e_k}{d_k^T A d_k}$$

Proof of convergence

Lemma 2. Convergence of conjugate direction method.

Suppose, we solve n-dimensional quadratic convex optimization problem (1). The conjugate directions method

$$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$

with $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, Ad_i \rangle}$ taken from the line search, converges for at most n steps of the algorithm.

Proof

1. We need to prove, that $\delta_i = -\alpha_i$:

$$e_0 = x_0 - x^* = \sum_{i=0}^{n-1} \delta_i d_i$$

2. We multiply both hand sides from the left by $d_k^T A$:

$$d_k^T A e_0 = \sum_{i=0}^{n-1} \delta_i d_k^T A d_i = \delta_k d_k^T A d_k$$

$$d_k^T A \left(e_0 + \sum_{i=0}^{k-1} \alpha_i d_i \right) = d_k^T A e_k = \delta_k d_k^T A d_k \quad (A - \text{ orthogonality})$$

$$\delta_k = \frac{d_k^T A e_k}{d_k^T A d_k} = -\frac{d_k^T r_k}{d_k^T A d_k}$$

Proof of convergence

Lemma 2. Convergence of conjugate direction method.

Suppose, we solve n-dimensional quadratic convex optimization problem (1). The conjugate directions method

$$x_{k+1} = x_0 + \sum_{i=0}^{k} \alpha_i d_i$$

with $\alpha_i = \frac{\langle d_i, r_i \rangle}{\langle d_i, Ad_i \rangle}$ taken from the line search, converges for at most n steps of the algorithm.

Proof

1. We need to prove, that $\delta_i = -\alpha_i$:

$$e_0 = x_0 - x^* = \sum_{i=0}^{n-1} \delta_i d_i$$

2. We multiply both hand sides from the left by $d_k^T A$:

$$d_k^T A e_0 = \sum_{i=0}^{n-1} \delta_i d_k^T A d_i = \delta_k d_k^T A d_k$$

$$d_k^T A \left(e_0 + \sum_{i=0}^{k-1} \alpha_i d_i \right) = d_k^T A e_k = \delta_k d_k^T A d_k \quad (A - \text{ orthogonality})$$

$$\delta_k = \frac{d_k^T A e_k}{d_k^T A d_k} = -\frac{d_k^T r_k}{d_k^T A d_k} \Leftrightarrow \delta_k = -\alpha_k$$

Lemms for convergence

Lemma 3. Error decomposition

$$e_i = \sum_{j=i}^{n-1} -\alpha_j d_j \tag{6}$$

Lemms for convergence

Lemma 3. Error decomposition

$$e_i = \sum_{j=i}^{n-1} -\alpha_j d_j \tag{6}$$

Proof

By definition

$$e_i = e_0 + \sum_{j=0}^{i-1} \alpha_j d_j$$

Lemms for convergence

Lemma 3. Error decomposition

$$e_i = \sum_{j=i}^{n-1} -\alpha_j d_j \tag{6}$$

Proof

By definition

$$e_i = e_0 + \sum_{j=0}^{i-1} \alpha_j d_j = x_0 - x^* + \sum_{j=0}^{i-1} \alpha_j d_j$$

Lemms for convergence

Lemma 3. Error decomposition

$$e_i = \sum_{j=i}^{n-1} -\alpha_j d_j \tag{6}$$

Proof

By definition

$$e_i = e_0 + \sum_{j=0}^{i-1} \alpha_j d_j = x_0 - x^* + \sum_{j=0}^{i-1} \alpha_j d_j = -\sum_{j=0}^{n-1} \alpha_j d_j + \sum_{j=0}^{i-1} \alpha_j d_j$$

Lemms for convergence

Lemma 3. Error decomposition

$$e_i = \sum_{j=i}^{n-1} -\alpha_j d_j \tag{6}$$

Proof

By definition

$$e_i = e_0 + \sum_{j=0}^{i-1} \alpha_j d_j = x_0 - x^* + \sum_{j=0}^{i-1} \alpha_j d_j = -\sum_{j=0}^{n-1} \alpha_j d_j + \sum_{j=0}^{i-1} \alpha_j d_j = \sum_{j=i}^{n-1} -\alpha_j d_j$$

Lemms for convergence

Lemma 4. Residual is orthogonal to all previous directions for CD

Consider residual of the CD method at k iteration r_k, then for any $i < k$:

$$d_i^T r_k = 0 \tag{7}$$

Lemms for convergence

Lemma 4. Residual is orthogonal to all previous directions for CD

Consider residual of the CD method at k iteration r_k, then for any $i < k$:

$$d_i^T r_k = 0 \tag{7}$$

Proof

Let's write down (6) for some fixed index k:

Lemms for convergence

Lemma 4. Residual is orthogonal to all previous directions for CD

Consider residual of the CD method at k iteration r_k, then for any $i < k$:

$$d_i^T r_k = 0 \tag{7}$$

Proof

Let's write down (6) for some fixed index k:

$$e_k = \sum_{j=k}^{n-1} -\alpha_j d_j$$

Lemms for convergence

Lemma 4. Residual is orthogonal to all previous directions for CD

Consider residual of the CD method at k iteration r_k, then for any $i < k$:

$$d_i^T r_k = 0 \tag{7}$$

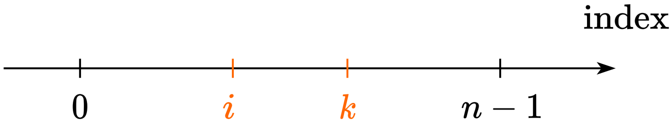
Proof

Let's write down (6) for some fixed index k:

$$e_k = \sum_{j=k}^{n-1} -\alpha_j d_j$$

Multiply both sides by $-d_i^T A \cdot$

$$-d_i^T A e_k = \sum_{j=k}^{n-1} \alpha_j d_i^T A d_j = 0$$



Thus, $d_i^T r_k = 0$ and residual r_k is orthogonal to all previous directions d_i for the CD method.

The idea of the Conjugate Gradients (CG) method

- It is literally the Conjugate Direction method, where we have a special (effective) choice of $d_0, \ldots, d_{n-1}$.

The idea of the Conjugate Gradients (CG) method

- It is literally the Conjugate Direction method, where we have a special (effective) choice of $d_0, \ldots, d_{n-1}$.
- In fact, we use the Gram-Schmidt process with A-orthogonality instead of Euclidian orthogonality to get them from a set of starting vectors.

The idea of the Conjugate Gradients (CG) method

- It is literally the Conjugate Direction method, where we have a special (effective) choice of $d_0, \ldots, d_{n-1}$.
- In fact, we use the Gram-Schmidt process with A-orthogonality instead of Euclidian orthogonality to get them from a set of starting vectors.
- The residuals on each iteration $r_0, \ldots, r_{n-1}$ are used as starting vectors for Gram-Schmidt process.

The idea of the Conjugate Gradients (CG) method

- It is literally the Conjugate Direction method, where we have a special (effective) choice of $d_0, \ldots, d_{n-1}$.
- In fact, we use the Gram-Schmidt process with A-orthogonality instead of Euclidian orthogonality to get them from a set of starting vectors.
- The residuals on each iteration $r_0, \ldots, r_{n-1}$ are used as starting vectors for Gram-Schmidt process.
- The main idea is that for an arbitrary CD method, the Gramm-Schmidt process is quite computationally expensive and requires a quadratic number of vector addition and scalar product operations $\mathcal{O}\left(n^2\right)$, while in the case of CG, we will show that the complexity of this procedure can be reduced to linear $\mathcal{O}\left(n\right)$.

The idea of the Conjugate Gradients (CG) method

- It is literally the Conjugate Direction method, where we have a special (effective) choice of $d_0, \ldots, d_{n-1}$.
- In fact, we use the Gram-Schmidt process with A-orthogonality instead of Euclidian orthogonality to get them from a set of starting vectors.
- The residuals on each iteration $r_0, \ldots, r_{n-1}$ are used as starting vectors for Gram-Schmidt process.
- The main idea is that for an arbitrary CD method, the Gramm-Schmidt process is quite computationally expensive and requires a quadratic number of vector addition and scalar product operations $\mathcal{O}\left(n^2\right)$, while in the case of CG, we will show that the complexity of this procedure can be reduced to linear $\mathcal{O}\left(n\right)$.

The idea of the Conjugate Gradients (CG) method

- It is literally the Conjugate Direction method, where we have a special (effective) choice of $d_0, \ldots, d_{n-1}$.
- In fact, we use the Gram-Schmidt process with A-orthogonality instead of Euclidian orthogonality to get them from a set of starting vectors.
- The residuals on each iteration $r_0, \ldots, r_{n-1}$ are used as starting vectors for Gram-Schmidt process.
- The main idea is that for an arbitrary CD method, the Gramm-Schmidt process is quite computationally expensive and requires a quadratic number of vector addition and scalar product operations $\mathcal{O}\left(n^2\right)$, while in the case of CG, we will show that the complexity of this procedure can be reduced to linear $\mathcal{O}\left(n\right)$.

$$\text{CG} = \text{CD} + r_0, \ldots, r_{n-1} \text{ as starting vectors for Gram–Schmidt} + A\text{-orthogonality.}$$

Lemms for convergence

Lemma 5. Residuals are orthogonal to each other in the CG method

All residuals are pairwise orthogonal to each other in the CG method:

$$r_i^T r_k = 0 \qquad \forall i \neq k \tag{8}$$

Lemms for convergence

Lemma 5. Residuals are orthogonal to each other in the CG method

All residuals are pairwise orthogonal to each other in the CG method:

$$r_i^T r_k = 0 \qquad \forall i \neq k \tag{8}$$

Proof

Let's write down Gram-Schmidt process (2) with $\langle \cdot, \cdot \rangle$ replaced with $\langle \cdot, \cdot \rangle_A = x^T A y$

Lemms for convergence

Lemma 5. Residuals are orthogonal to each other in the CG method

All residuals are pairwise orthogonal to each other in the CG method:

$$r_i^T r_k = 0 \qquad \forall i \neq k \tag{8}$$

Proof

Let's write down Gram-Schmidt process (2) with $\langle \cdot, \cdot \rangle$ replaced with $\langle \cdot, \cdot \rangle_A = x^T A y$

$$d_i = u_i + \sum_{j=0}^{k-1} \beta_{ji} d_j \;\; \beta_{ji} = - \frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} \tag{9}$$

Lemms for convergence

Lemma 5. Residuals are orthogonal to each other in the CG method

All residuals are pairwise orthogonal to each other in the CG method:

$$r_i^T r_k = 0 \qquad \forall i \neq k \tag{8}$$

Proof

Let's write down Gram-Schmidt process (2) with $\langle \cdot, \cdot \rangle$ replaced with $\langle \cdot, \cdot \rangle_A = x^T A y$

$$d_i = u_i + \sum_{j=0}^{k-1} \beta_{ji} d_j \quad \beta_{ji} = - \frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} \tag{9}$$

Then, we use residuals as starting vectors for the process and $u_i = r_i$.

Lemms for convergence

Lemma 5. Residuals are orthogonal to each other in the CG method

All residuals are pairwise orthogonal to each other in the CG method:

$$r_i^T r_k = 0 \qquad \forall i \neq k \tag{8}$$

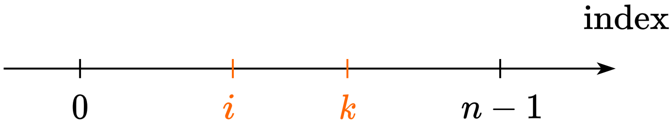
Proof

Let's write down Gram-Schmidt process (2) with $\langle \cdot, \cdot \rangle$ replaced with $\langle \cdot, \cdot \rangle_A = x^T A y$

$$d_i = u_i + \sum_{j=0}^{k-1} \beta_{ji} d_j \quad \beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} \tag{9}$$

Then, we use residuals as starting vectors for the process and $u_i = r_i$.

$$d_i = r_i + \sum_{j=0}^{k-1} \beta_{ji} d_j \quad \beta_{ji} = -\frac{\langle d_j, r_i \rangle_A}{\langle d_j, d_j \rangle_A} \tag{10}$$



Multiply both sides of (9) by $r_k^T \cdot$ for some index k:

$$r_k^T d_i = r_k^T u_i + \sum_{j=0}^{k-1} \beta_{ji} r_k^T d_j$$

Lemms for convergence

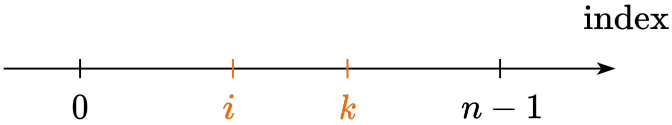
Lemma 5. Residuals are orthogonal to each other in the CG method

All residuals are pairwise orthogonal to each other in the CG method:

$$r_i^T r_k = 0 \qquad \forall i \neq k \tag{8}$$

Proof

Let's write down Gram-Schmidt process (2) with $\langle \cdot, \cdot \rangle$ replaced with $\langle \cdot, \cdot \rangle_A = x^T A y$



$$d_i = u_i + \sum_{j=0}^{k-1} \beta_{ji} d_j \quad \beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} \tag{9}$$

Then, we use residuals as starting vectors for the process and $u_i = r_i$.

$$d_i = r_i + \sum_{j=0}^{k-1} \beta_{ji} d_j \quad \beta_{ji} = -\frac{\langle d_j, r_i \rangle_A}{\langle d_j, d_j \rangle_A} \tag{10}$$

Multiply both sides of (9) by $r_k^T \cdot$ for some index k:

$$r_k^T d_i = r_k^T u_i + \sum_{j=0}^{k-1} \beta_{ji} r_k^T d_j$$

If $j < i < k$, we have the lemma 4 with $d_i^T r_k = 0$ and $d_j^T r_k = 0$. We have:

$$r_k^T u_i = 0 \text{ for CD} \quad r_k^T r_i = 0 \text{ for CG}$$

Lemms for convergence

Moreover, if $k = i$:

$$r_k^T d_k = r_k^T u_k + \sum_{j=0}^{k-1} \beta_{jk} r_k^T d_j$$

Lemms for convergence

Moreover, if $k = i$:

$$r_k^T d_k = r_k^T u_k + \sum_{j=0}^{k-1} \beta_{jk} r_k^T d_j = r_k^T u_k + 0,$$

Lemms for convergence

Moreover, if $k = i$:

$$r_k^T d_k = r_k^T u_k + \sum_{j=0}^{k-1} \beta_{jk} r_k^T d_j = r_k^T u_k + 0,$$

Lemms for convergence

Moreover, if $k = i$:

$$r_k^T d_k = r_k^T u_k + \sum_{j=0}^{k-1} \beta_{jk} r_k^T d_j = r_k^T u_k + 0,$$

and we have for any k (due to arbitrary choice of i):

$$r_k^T d_k = r_k^T u_k. \tag{11}$$

Lemms for convergence

Moreover, if $k = i$:

$$r_k^T d_k = r_k^T u_k + \sum_{j=0}^{k-1} \beta_{jk} r_k^T d_j = r_k^T u_k + 0,$$

and we have for any k (due to arbitrary choice of i):

$$r_k^T d_k = r_k^T u_k. \tag{11}$$

Lemma 6. Residual recalculation

$$r_{k+1} = r_k - \alpha_k A d_k \tag{12}$$

Lemms for convergence

Moreover, if $k = i$:

$$r_k^T d_k = r_k^T u_k + \sum_{j=0}^{k-1} \beta_{jk} r_k^T d_j = r_k^T u_k + 0,$$

and we have for any k (due to arbitrary choice of i):

$$r_k^T d_k = r_k^T u_k. \tag{11}$$

Lemma 6. Residual recalculation

$$r_{k+1} = r_k - \alpha_k A d_k \tag{12}$$

$$r_{k+1} = -Ae_{k+1} = -A\left(e_k + \alpha_k d_k\right) = -Ae_k - \alpha_k A d_k = r_k - \alpha_k A d_k$$

Finally, all these above lemmas are enough to prove, that $\beta_{ji} = 0$ for all i, j, except the neighboring ones.

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A}$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j}$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j}$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$

$$\alpha_j \langle r_i, A d_j \rangle$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$
$$\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_j \rangle - \langle r_i, r_{j+1} \rangle$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$
$$\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_j \rangle - \langle r_i, r_{j+1} \rangle$$

1. If $i = j$: $\alpha_i \langle r_i, A d_i \rangle = \langle r_i, r_i \rangle - \langle r_i, r_{i+1} \rangle = \langle r_i, r_i \rangle$. This case is not of interest due to the GS process.

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$
$$\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_j \rangle - \langle r_i, r_{j+1} \rangle$$

1. If $i = j$: $\alpha_i \langle r_i, A d_i \rangle = \langle r_i, r_i \rangle - \langle r_i, r_{i+1} \rangle = \langle r_i, r_i \rangle$. This case is not of interest due to the GS process.
2. Neighboring case $i = j + 1$: $\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_{i-1} \rangle - \langle r_i, r_i \rangle = -\langle r_i, r_i \rangle$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$
$$\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_j \rangle - \langle r_i, r_{j+1} \rangle$$

1. If $i = j$: $\alpha_i \langle r_i, A d_i \rangle = \langle r_i, r_i \rangle - \langle r_i, r_{i+1} \rangle = \langle r_i, r_i \rangle$. This case is not of interest due to the GS process.
2. Neighboring case $i = j + 1$: $\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_{i-1} \rangle - \langle r_i, r_i \rangle = -\langle r_i, r_i \rangle$
3. For any other case: $\alpha_j \langle r_i, A d_j \rangle = 0$, because all residuals are orthogonal to each other.

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$
$$\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_j \rangle - \langle r_i, r_{j+1} \rangle$$

1. If $i = j$: $\alpha_i \langle r_i, A d_i \rangle = \langle r_i, r_i \rangle - \langle r_i, r_{i+1} \rangle = \langle r_i, r_i \rangle$. This case is not of interest due to the GS process.
2. Neighboring case $i = j + 1$: $\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_{i-1} \rangle - \langle r_i, r_i \rangle = -\langle r_i, r_i \rangle$
3. For any other case: $\alpha_j \langle r_i, A d_j \rangle = 0$, because all residuals are orthogonal to each other.

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$
$$\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_j \rangle - \langle r_i, r_{j+1} \rangle$$

1. If $i = j$: $\alpha_i \langle r_i, A d_i \rangle = \langle r_i, r_i \rangle - \langle r_i, r_{i+1} \rangle = \langle r_i, r_i \rangle$. This case is not of interest due to the GS process.
2. Neighboring case $i = j + 1$: $\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_{i-1} \rangle - \langle r_i, r_i \rangle = -\langle r_i, r_i \rangle$
3. For any other case: $\alpha_j \langle r_i, A d_j \rangle = 0$, because all residuals are orthogonal to each other.

Finally, we have a formula for $i = j + 1$:

$$\beta_{ji} = -\frac{r_i^T A d_j}{d_j^T A d_j}$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$

$$\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_j \rangle - \langle r_i, r_{j+1} \rangle$$

1. If $i = j$: $\alpha_i \langle r_i, A d_i \rangle = \langle r_i, r_i \rangle - \langle r_i, r_{i+1} \rangle = \langle r_i, r_i \rangle$. This case is not of interest due to the GS process.
2. Neighboring case $i = j + 1$: $\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_{i-1} \rangle - \langle r_i, r_i \rangle = -\langle r_i, r_i \rangle$
3. For any other case: $\alpha_j \langle r_i, A d_j \rangle = 0$, because all residuals are orthogonal to each other.

Finally, we have a formula for $i = j + 1$:

$$\beta_{ji} = -\frac{r_i^T A d_j}{d_j^T A d_j} = \frac{1}{\alpha_j} \frac{\langle r_i, r_i \rangle}{d_j^T A d_j}$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$
$$\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_j \rangle - \langle r_i, r_{j+1} \rangle$$

1. If $i = j$: $\alpha_i \langle r_i, A d_i \rangle = \langle r_i, r_i \rangle - \langle r_i, r_{i+1} \rangle = \langle r_i, r_i \rangle$. This case is not of interest due to the GS process.
2. Neighboring case $i = j + 1$: $\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_{i-1} \rangle - \langle r_i, r_i \rangle = -\langle r_i, r_i \rangle$
3. For any other case: $\alpha_j \langle r_i, A d_j \rangle = 0$, because all residuals are orthogonal to each other.

Finally, we have a formula for $i = j + 1$:

$$\beta_{ji} = -\frac{r_i^T A d_j}{d_j^T A d_j} = \frac{1}{\alpha_j} \frac{\langle r_i, r_i \rangle}{d_j^T A d_j} = \frac{d_j^T A d_j}{d_j^T r_j} \frac{\langle r_i, r_i \rangle}{d_j^T A d_j}$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$
$$\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_j \rangle - \langle r_i, r_{j+1} \rangle$$

1. If $i = j$: $\alpha_i \langle r_i, A d_i \rangle = \langle r_i, r_i \rangle - \langle r_i, r_{i+1} \rangle = \langle r_i, r_i \rangle$. This case is not of interest due to the GS process.
2. Neighboring case $i = j + 1$: $\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_{i-1} \rangle - \langle r_i, r_i \rangle = -\langle r_i, r_i \rangle$
3. For any other case: $\alpha_j \langle r_i, A d_j \rangle = 0$, because all residuals are orthogonal to each other.

Finally, we have a formula for $i = j + 1$:

$$\beta_{ji} = -\frac{r_i^T A d_j}{d_j^T A d_j} = \frac{1}{\alpha_j} \frac{\langle r_i, r_i \rangle}{d_j^T A d_j} = \frac{d_j^T A d_j}{d_j^T r_j} \frac{\langle r_i, r_i \rangle}{d_j^T A d_j} = \frac{\langle r_i, r_i \rangle}{\langle r_j, r_j \rangle}$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$

$$\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_j \rangle - \langle r_i, r_{j+1} \rangle$$

1. If $i = j$: $\alpha_i \langle r_i, A d_i \rangle = \langle r_i, r_i \rangle - \langle r_i, r_{i+1} \rangle = \langle r_i, r_i \rangle$. This case is not of interest due to the GS process.
2. Neighboring case $i = j + 1$: $\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_{i-1} \rangle - \langle r_i, r_i \rangle = -\langle r_i, r_i \rangle$
3. For any other case: $\alpha_j \langle r_i, A d_j \rangle = 0$, because all residuals are orthogonal to each other.

Finally, we have a formula for $i = j + 1$:

$$\beta_{ji} = -\frac{r_i^T A d_j}{d_j^T A d_j} = \frac{1}{\alpha_j} \frac{\langle r_i, r_i \rangle}{d_j^T A d_j} = \frac{d_j^T A d_j}{d_j^T r_j} \frac{\langle r_i, r_i \rangle}{d_j^T A d_j} = \frac{\langle r_i, r_i \rangle}{\langle r_j, r_j \rangle} = \frac{\langle r_i, r_i \rangle}{\langle r_{i-1}, r_{i-1} \rangle}$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$

$$\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_j \rangle - \langle r_i, r_{j+1} \rangle$$

1. If $i = j$: $\alpha_i \langle r_i, A d_i \rangle = \langle r_i, r_i \rangle - \langle r_i, r_{i+1} \rangle = \langle r_i, r_i \rangle$. This case is not of interest due to the GS process.
2. Neighboring case $i = j + 1$: $\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_{i-1} \rangle - \langle r_i, r_i \rangle = -\langle r_i, r_i \rangle$
3. For any other case: $\alpha_j \langle r_i, A d_j \rangle = 0$, because all residuals are orthogonal to each other.

Finally, we have a formula for $i = j + 1$:

$$\beta_{ji} = -\frac{r_i^T A d_j}{d_j^T A d_j} = \frac{1}{\alpha_j} \frac{\langle r_i, r_i \rangle}{d_j^T A d_j} = \frac{d_j^T A d_j}{d_j^T r_j} \frac{\langle r_i, r_i \rangle}{d_j^T A d_j} = \frac{\langle r_i, r_i \rangle}{\langle r_j, r_j \rangle} = \frac{\langle r_i, r_i \rangle}{\langle r_{i-1}, r_{i-1} \rangle}$$

Gram-Schmidt process in CG method

Consider the Gram-Schmidt process in the CG method

$$\beta_{ji} = -\frac{\langle d_j, u_i \rangle_A}{\langle d_j, d_j \rangle_A} = -\frac{d_j^T A u_i}{d_j^T A d_j} = -\frac{d_j^T A r_i}{d_j^T A d_j} = -\frac{r_i^T A d_j}{d_j^T A d_j}.$$

Consider the scalar product $\langle r_i, r_{j+1} \rangle$ using (12):

$$\langle r_i, r_{j+1} \rangle = \langle r_i, r_j - \alpha_j A d_j \rangle = \langle r_i, r_j \rangle - \alpha_j \langle r_i, A d_j \rangle$$
$$\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_j \rangle - \langle r_i, r_{j+1} \rangle$$

1. If $i = j$: $\alpha_i \langle r_i, A d_i \rangle = \langle r_i, r_i \rangle - \langle r_i, r_{i+1} \rangle = \langle r_i, r_i \rangle$. This case is not of interest due to the GS process.
2. Neighboring case $i = j + 1$: $\alpha_j \langle r_i, A d_j \rangle = \langle r_i, r_{i-1} \rangle - \langle r_i, r_i \rangle = -\langle r_i, r_i \rangle$
3. For any other case: $\alpha_j \langle r_i, A d_j \rangle = 0$, because all residuals are orthogonal to each other.

Finally, we have a formula for $i = j + 1$:

$$\beta_{ji} = -\frac{r_i^T A d_j}{d_j^T A d_j} = \frac{1}{\alpha_j} \frac{\langle r_i, r_i \rangle}{d_j^T A d_j} = \frac{d_j^T A d_j}{d_j^T r_j} \frac{\langle r_i, r_i \rangle}{d_j^T A d_j} = \frac{\langle r_i, r_i \rangle}{\langle r_j, r_j \rangle} = \frac{\langle r_i, r_i \rangle}{\langle r_{i-1}, r_{i-1} \rangle}$$

And for the direction

$$d_{k+1} = r_{k+1} + \beta_{k,k+1} d_k, \qquad \beta_{k,k+1} = \beta_k = \frac{\langle r_{k+1}, r_{k+1} \rangle}{\langle r_k, r_k \rangle}.$$

Conjugate gradients method

$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$

if $\mathbf{r}_0$ is sufficiently small, then return $\mathbf{x}_0$ as the result

$\mathbf{d}_0 := \mathbf{r}_0$

$k := 0$

repeat

$$\alpha_k := \frac{\mathbf{r}_k^\top \mathbf{r}_k}{\mathbf{d}_k^\top \mathbf{A}\mathbf{d}_k}$$

$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k \mathbf{d}_k$

$\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{d}_k$

if $\mathbf{r}_{k+1}$ is sufficiently small, then exit loop

$$\beta_k := \frac{\mathbf{r}_{k+1}^\top \mathbf{r}_{k+1}}{\mathbf{r}_k^\top \mathbf{r}_k}$$

$\mathbf{d}_{k+1} := \mathbf{r}_{k+1} + \beta_k \mathbf{d}_k$

$k := k + 1$

end repeat

return $\mathbf{x}_{k+1}$ as the result

Convergence

Theorem 1. If matrix A has only r different eigenvalues, then the conjugate gradient method converges in r iterations.

Theorem 2. The following convergence bound holds

$$\|x_k - x^*\|_A \leq 2\left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^k \|x_0 - x^*\|_A,$$

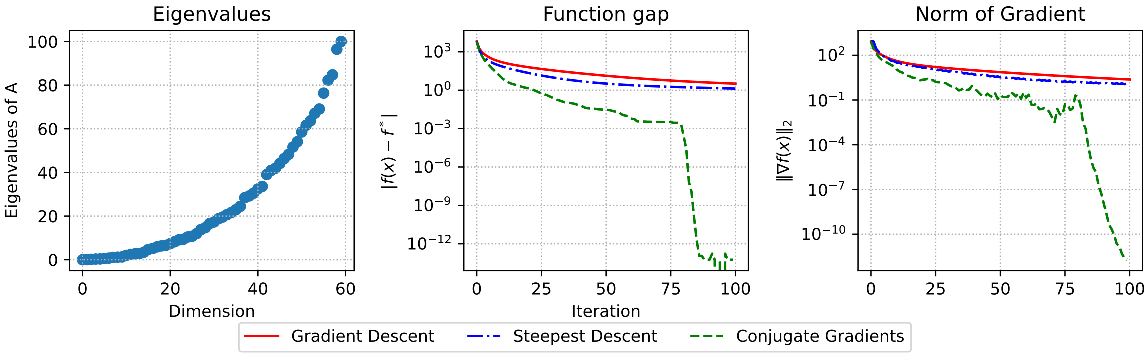
where $\|x\|_A^2 = x^\top A x$ and $\kappa(A) = \frac{\lambda_1(A)}{\lambda_n(A)}$ is the conditioning number of matrix A, $\lambda_1(A) \geq ... \geq \lambda_n(A)$ are the eigenvalues of matrix A

Note: Compare the coefficient of the geometric progression with its analog in gradient descent.

Numerical results

$$f(x) = \frac{1}{2}x^T Ax - b^T x \to \min_{x \in \mathbb{R}^n}$$

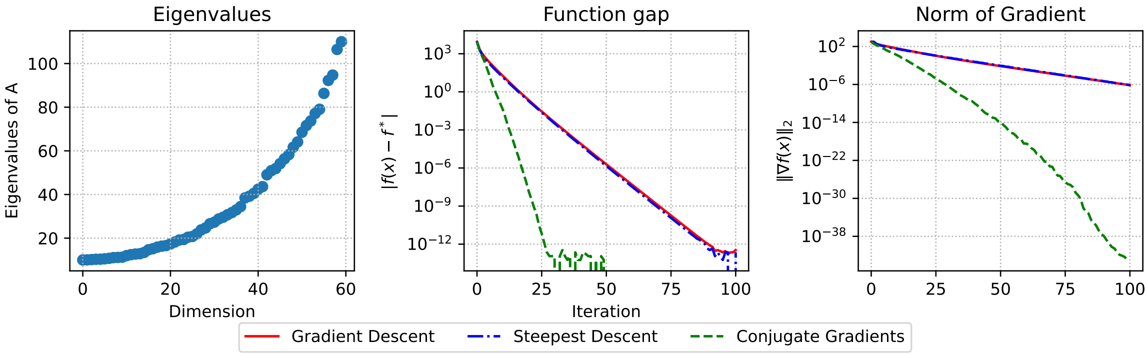
Convex quadratics. n=60, random matrix.



Numerical results

$$f(x) = \frac{1}{2}x^T A x - b^T x \to \min_{x \in \mathbb{R}^n}$$

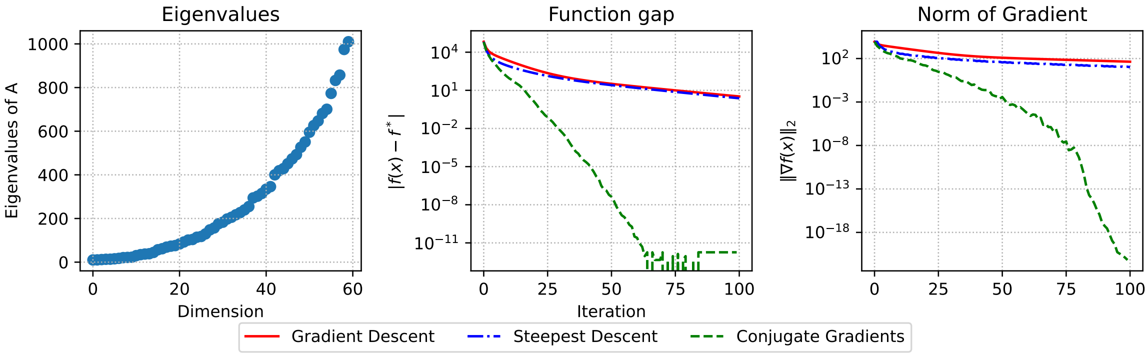
Strongly convex quadratics. n=60, random matrix.



Numerical results

$$f(x) = \frac{1}{2}x^T Ax - b^T x \to \min_{x \in \mathbb{R}^n}$$

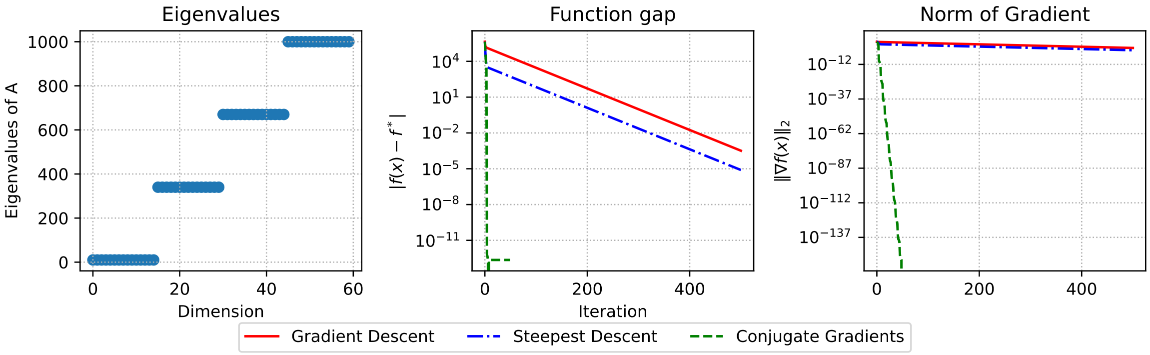
Strongly convex quadratics. n=60, random matrix.



Numerical results

$$f(x) = \frac{1}{2}x^T A x - b^T x \to \min_{x \in \mathbb{R}^n}$$

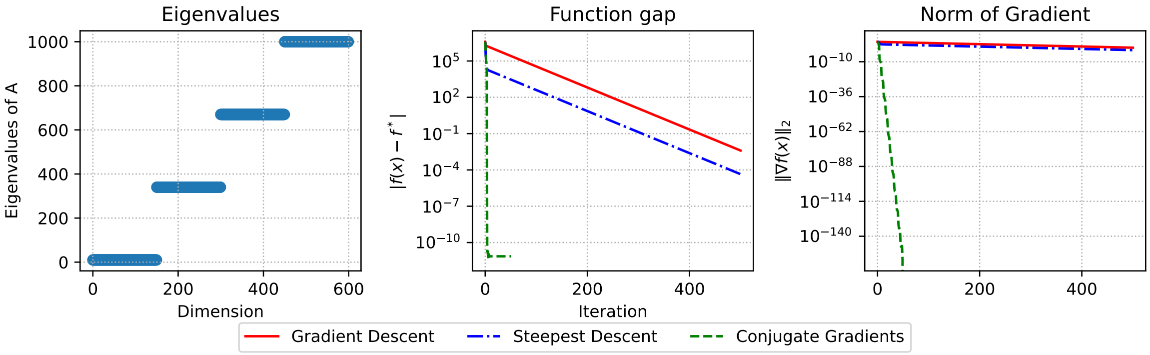
Strongly convex quadratics. n=60, clustered matrix.



Numerical results

$$f(x) = \frac{1}{2}x^T A x - b^T x \to \min_{x \in \mathbb{R}^n}$$

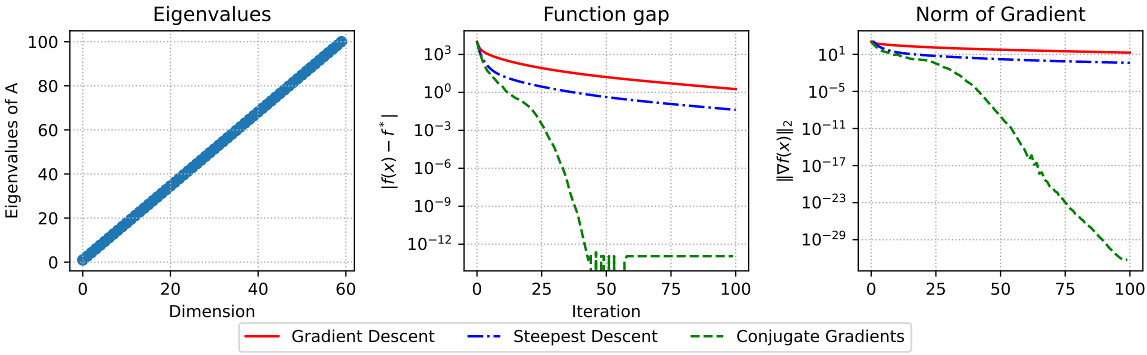
Strongly convex quadratics. n=600, clustered matrix.



Numerical results

$$f(x) = \frac{1}{2}x^T A x - b^T x \to \min_{x \in \mathbb{R}^n}$$

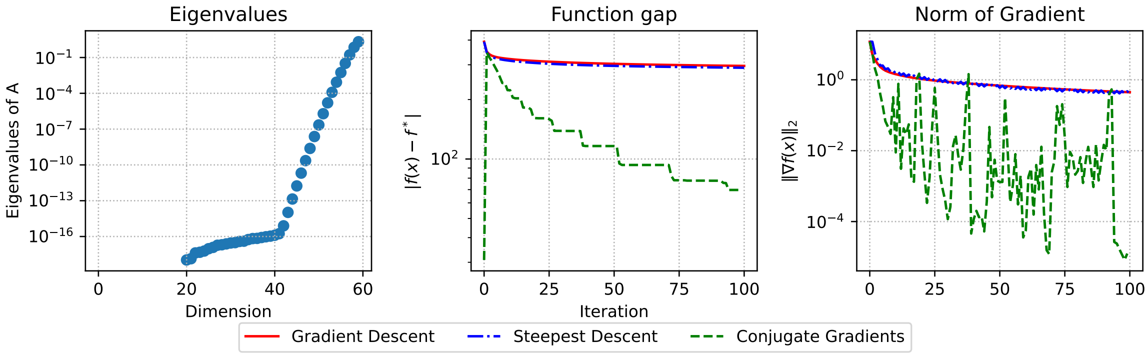
Strongly convex quadratics. n=60, uniform spectrum matrix.



Numerical results

$$f(x) = \frac{1}{2}x^T A x - b^T x \to \min_{x \in \mathbb{R}^n}$$

Strongly convex quadratics. n=60, Hilbert matrix.



Non-linear conjugate gradient method

In case we do not have an analytic expression for a function or its gradient, we will most likely not be able to solve the one-dimensional minimization problem analytically. Therefore, step 2 of the algorithm is replaced by the usual line search procedure. But there is the following mathematical trick for the fourth point:

For two iterations, it is fair:

$$x_{k+1} - x_k = cd_k,$$

where c is some kind of constant. Then for the quadratic case, we have:

$$\nabla f(x_{k+1}) - \nabla f(x_k) = (Ax_{k+1} - b) - (Ax_k - b) = A(x_{k+1} - x_k) = cAd_k$$

Expressing from this equation the work $Ad_k = \dfrac{1}{c}\left(\nabla f(x_{k+1}) - \nabla f(x_k)\right)$, we get rid of the "knowledge" of the function in step definition β_k, then point 4 will be rewritten as:

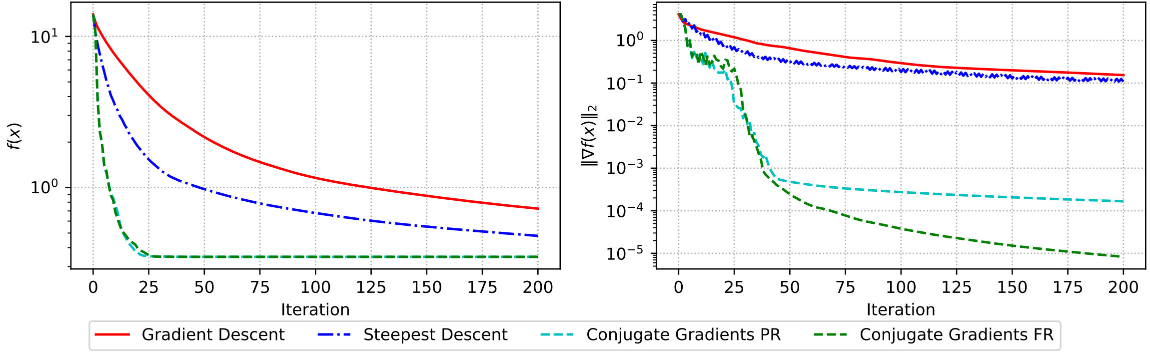
$$\beta_k = \frac{\nabla f(x_{k+1})^\top (\nabla f(x_{k+1}) - \nabla f(x_k))}{d_k^\top (\nabla f(x_{k+1}) - \nabla f(x_k))}.$$

This method is called the Polack-Ribier method.

Numerical results

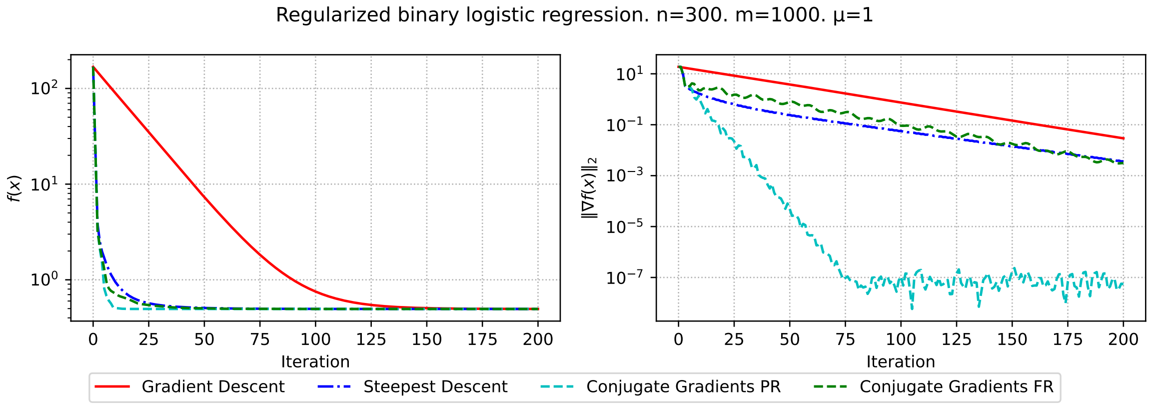
$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m} \log(1 + \exp(-y_i \langle a_i, x \rangle)) \to \min_{x \in \mathbb{R}^n}$$

Regularized binary logistic regression. n=300. m=1000. μ=0



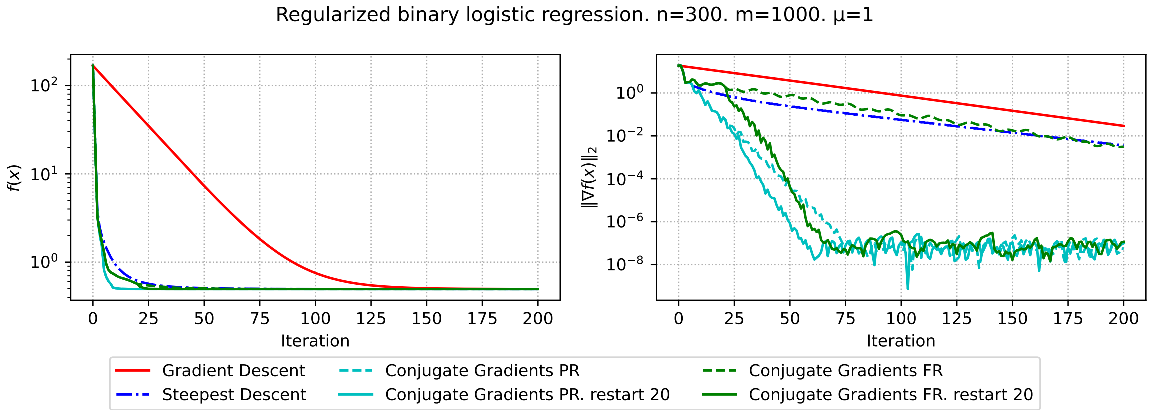
Numerical results

$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\log(1+\exp(-y_i\langle a_i, x\rangle)) \to \min_{x\in\mathbb{R}^n}$$



Numerical results

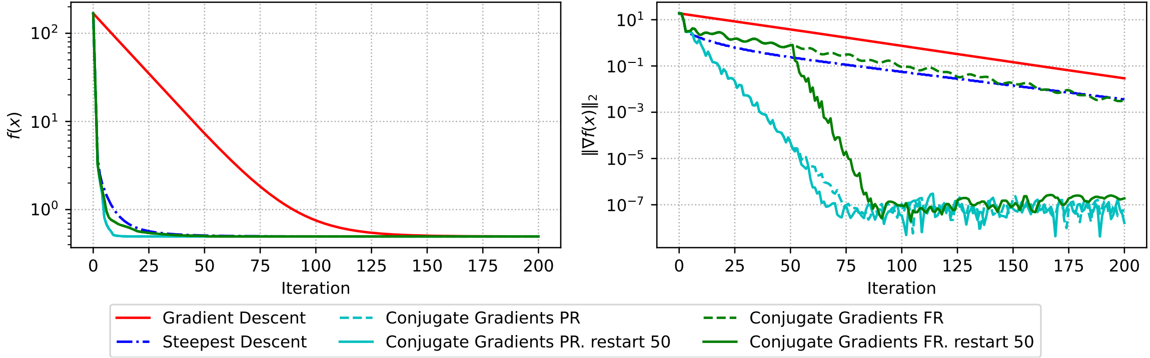
$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m} \log(1 + \exp(-y_i\langle a_i, x\rangle)) \to \min_{x\in\mathbb{R}^n}$$



Numerical results

$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\log(1+\exp(-y_i\langle a_i, x\rangle)) \to \min_{x\in\mathbb{R}^n}$$

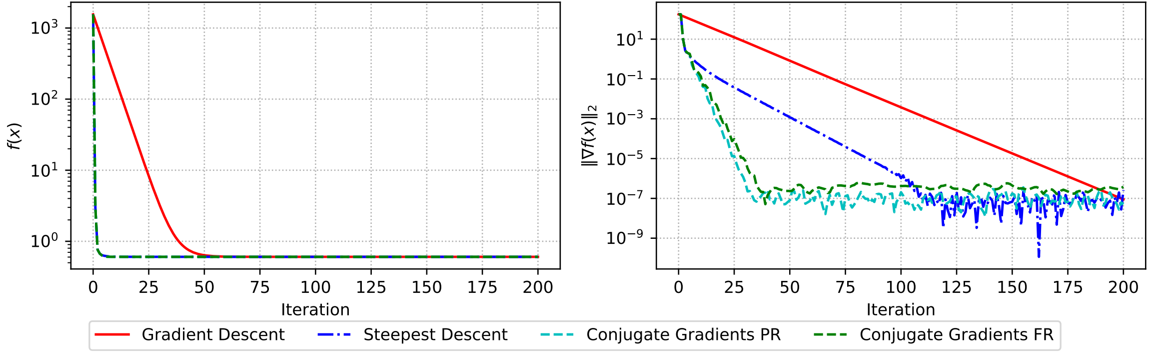
Regularized binary logistic regression. n=300. m=1000. μ=1



Numerical results

$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m}\log(1+\exp(-y_i\langle a_i, x\rangle)) \to \min_{x\in\mathbb{R}^n}$$

Regularized binary logistic regression. n=300. m=1000. μ=10



Numerical results

$$f(x) = \frac{\mu}{2}\|x\|_2^2 + \frac{1}{m}\sum_{i=1}^{m} \log(1 + \exp(-y_i\langle a_i, x\rangle)) \to \min_{x\in\mathbb{R}^n}$$