Conjugate Gradient Method: Steepest Descent with Orthogonal Directions under Change of Coordinate

J.R

July 21, 2021

In this note, I want to guide you through the motivation and derivation of the bizarre Conjugate Gradient Method (CG Method) and show you that a method as intricate as CG Method is nothing more than a steepest descent with a change of coordinate.

Contents

1 Linear Equation and Quadratic Form

Let's begin with a linear equation

$$Ax = b \tag{1.1}$$

where A is symmetric and positive definite. In most situation, we can solve this linear equation by factoring A in to a product of a lower-triangle matrix and an upper-triangle matrix and doing backward/forward substitution. However, when A is huge, for example, 1 million by 1 million, the factoring seems to be intractable. Thus we have to come up with other methods for solving a large-scale linear system.

The quadratic form of (1.1) is

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c \tag{1.2}$$

We call (1.2) the quadratic form of (1.1) because the solution to (1.2) $x^* = A^{-1}b$ is the solution to (1.1). In other words, if we can solve optimization problem (1.2), we can solve the linear equation (1.1). Let's introduce some definition which will be used across this note.

$$\begin{aligned} x^* &= A^{-1}b = \text{argmin}_x f(x) \\ e_{(i)} &= x_{(i)} - x^* \\ r_{(i)} &= -Ae_{(i)} = b - Ax_{(i)} = -\nabla f(x_{(i)}) \end{aligned}$$

$e^{(i)}$ is called the error at $x_{(i)}$ and $r_{(i)}$ is called the residual at $x_{(i)}$.

And in this note, we will have a running example where $A = 2 \cdot \begin{pmatrix} 0.416 & 0.168 \\ 0.168 & 0.416 \end{pmatrix}$, $b = \begin{pmatrix} 5 \\ 2.5 \end{pmatrix}$ and $c = 6.25$, so that $f(x) = (x-u)^T P(x-u)$, where P has eigenvalue decomposition $P = \begin{pmatrix} 2/\sqrt{5} & -1\sqrt{5} \\ 1\sqrt{5} & 2\sqrt{5} \end{pmatrix} \begin{pmatrix} 1/4 & 0 \\ 0 & 1/25 \end{pmatrix} \begin{pmatrix} 2\sqrt{5} & -1\sqrt{5} \\ 1\sqrt{5} & 2\sqrt{5} \end{pmatrix}^T$ and $u = [2, 1]$. The contour lines are showed below, which are ellipsoids tilted and centered at $[2, 1]$.

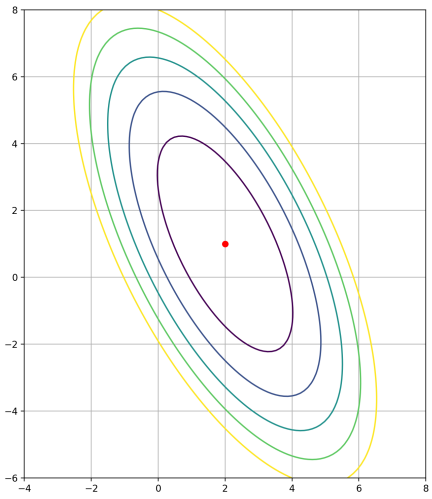


Figure 1: Contour lines of running example

2 Steepest Descent

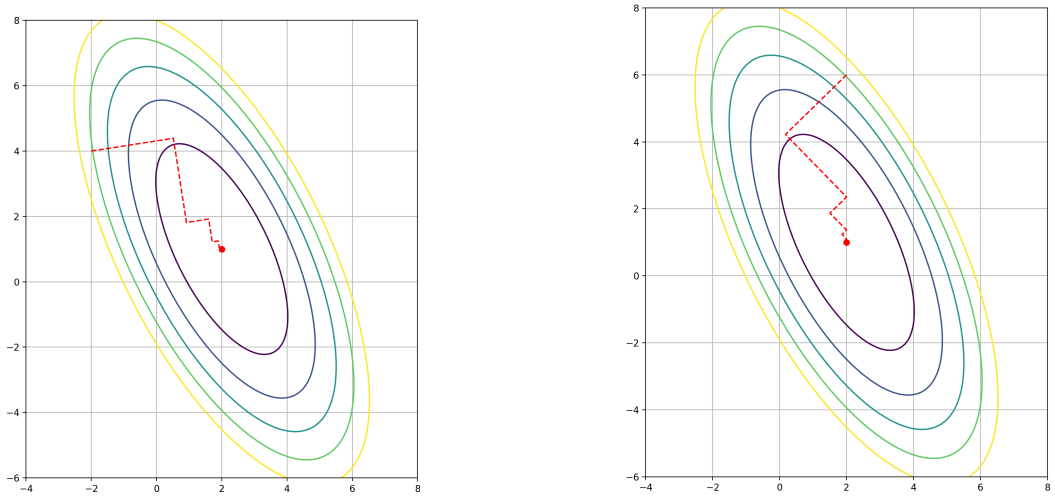
(1.2) is a convex optimization problem, we can solve it with many methods like newton method. However, in second-order methods, we need to invert A, which is too expansive so that we want to avoid. So we may want to use some first-order method. The first method coming to mind is gradient descent, but choosing step size in gradient descent can be tricky. For a problem has a simple quadratic form, we can easily do line search, which leads us to steepest descent. At each step in steepest descent, we evaluate the gradient, and go in the direction of negative gradient, which is $r_{(i)}$, until we reach a minimum along the direction of $r_{(i)}$. The step size is calculated by setting the gradient to 0

$$\frac{d}{d\alpha} f(x_{(i)} + \alpha_{(i)} r_{(i)}) = 0 \tag{2.1}$$

$$r_{(i)}^T [b - A(x_{(i)} + \alpha_{(i)} r_{(i)})] = 0 \tag{2.2}$$

$$\alpha_{(i)} = \frac{r_{(i)}^T r_{(i)}}{r_{(i)}^T A r_{(i)}} \tag{2.3}$$

Steepest descent on the running example with different starting point is shown in the following picture, stopping criterion is when $||r_{(i)}||_2 \leq 1e-6$.



(a) (b)

Figure 2: Steepest descent on running example

Notice that search directions are orthogonal to the previous one. We can easily get this result from (2.1).

$$\begin{aligned} &\frac{d}{d\alpha} f(x_{(i)} + \alpha_{(i)} r_{(i)}) \\ =&r_{(i)}^T \nabla f(x_{(i)} + \alpha_{(i)} r_{(i)}) \\ =&- r_{(i)}^T r_{(i+1)} \end{aligned} \tag{2.4}$$

when we set $\frac{d}{d\alpha} f(x_{(i)} + \alpha_{(i)} r_{(i)})$ to 0, $r_{(i)}^T r_{(i+1)} = 0$ naturally follows.

One problem with this method is that we step in the same direction for many times. For example, in figure 2(a), we first take a step to the right(roughly speaking), then take a step down, and right, and down, ... Wouldn't it be nice if we can take a full step to the right and never go in that direction again. To be more concrete, after take a step in the direction of $d_{(i)}$ with step size $\alpha_{(i)}$, the error at $x^{(i)} + \alpha_{(i)} d_{(i)}$ is orthogonal to previous searching direction $d_{(i)}$ so that we don't need to go in direction of $d_{(i)}$ ever again, as illustrated in the following picture.

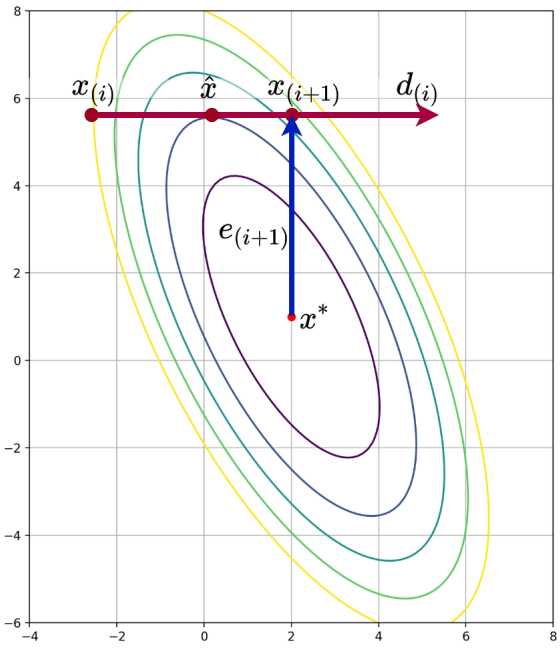


Figure 3

In figure 3, $\hat{x}$ is the minimizer subject to direction $d_{(i)}$. We can see that, to make $e_{(i+1)}$ orthogonal to the previous searching direction, we have to overshoot or undershoot compare to $\hat{x}$ depending on the shape of the ellipsoid. We can calculate the step size $\alpha_{(i)}$ by

$$d_{(i)}^T e^{(i+1)} = 0 \tag{2.5}$$

$$d_{(i)}^T (e^{(i)} + \alpha_{(i)} d_{(i)}) = 0$$

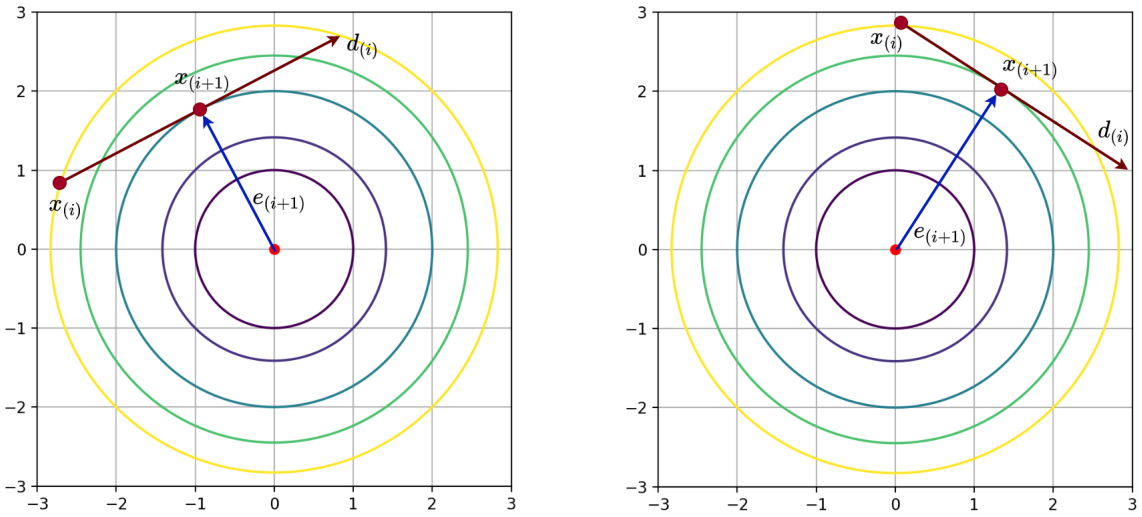
$$\alpha_{(i)} = -\frac{d_{(i)}^T e^{(i)}}{d_{(i)}^T d_{(i)}} \tag{2.6}$$

That means we cannot calculate $\alpha_{(i)}$ without knowing $e_{(i)}$, and the "orthogonal search directions" idea is infeasible. The problem with "orthogonal search directions" idea is that we have to "overshoot" or "undershoot". With that in mind, can we come up with something just like the "orthogonal search directions" but in which we don't have to overshoot? Fortunately, yes. And that is called "Conjugate Directions".

3 Conjugate Directions

In the previous section, we found that if we want $e_{(i+1)}$ to be orthogonal to previous search direction $d_{(i)}$, we have to "overshoot" or "undershoot".

The reason why we have to overshoot or undershoot is the asymmetry of ellipsoid. When we have a unit circle or unit ball, which is homogeneous in all direction, the resulting error after minimizing along a direction is guaranteed to be orthogonal to the search direction, as illustrated in the following picture.



(a) (b)

Figure 4: Orthogonal directions on unit circle

We can see from the picture that when minimize a quadratic function with $A = I$ along a direction, resulting error is orthogonal to error. Mathematical proof can be easily given.

Now the problem is, we have a quadratic function $f(x) = \frac{1}{2}x^T Ax - b^T x + c$, whose contour lines are ellipsoids, not unit balls. How can we make it a unit ball? The answer is, not surprisingly, yes and by changing coordinate.

Let $u = A^{1/2}x$, we have

$$\begin{aligned} f(x) = \tilde{f}(u) &= \frac{1}{2}u^T u - (A^{-1/2}b)^T u + c \\ &= \frac{1}{2}\left[(u - A^{-1/2}b)^T (u - A^{-1/2}b)\right] - \frac{b^T A^{-1} b}{2} + c \end{aligned} \tag{3.1}$$

by dropping constant terms, we can see the contour lines of $\tilde{f}$ are unit circles centered at $A^{-1/2}b$, as illustrated in the following picture.

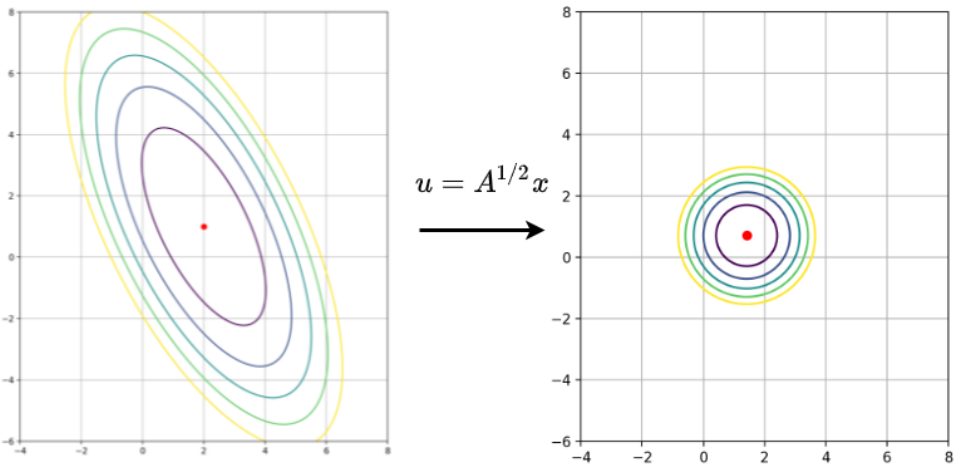


Figure 5: Change of coordinate $u = A^{1/2}x$

We can easily apply orthogonal directions as search direction in the new coordinate system, and when we find $u^* = A^{-1/2}b$, we can get $x^* = A^{-1/2}u^*$. (Actually we do not have to do this, but this is the idea.)

Thus, instead of orthogonal directions in the old system of x where the ellipsoids lie, we now need a group of direction which are orthogonal to each other in the new coordinate system.

Suppose $d_{(i)}$ and $d_{(j)}$ are two vectors in old coordinate system, under change of coordinate, we have $\tilde{d}_{(i)} = A^{1/2}d_{(i)}$ and $\tilde{d}_{(j)} = A^{1/2}d_{(j)}$. Since $d_{(i)}$ and $d_{(j)}$ are orthogonal under after of coordinate, we have

$$\begin{aligned}(A^{1/2}d_{(i)})^T A^{1/2}d_{(j)} &= 0 \\ d_{(i)}^T A d_{(j)} &= 0 \end{aligned} \tag{3.2}$$

If $d_{(i)}$ and $d_{(i)}$ satisfies (3.2), we say $d_{(i)}$ and $d_{(i)}$ are A-orthogonal, or in more fancy words, A-conjugate. Roughly speaking, in this context, conjugate means "orthogonal under linear transformation". The following pictures shows some conjugate vectors when $A = \begin{pmatrix} 1/4 & 0 \\ 0 & 1 \end{pmatrix}$.

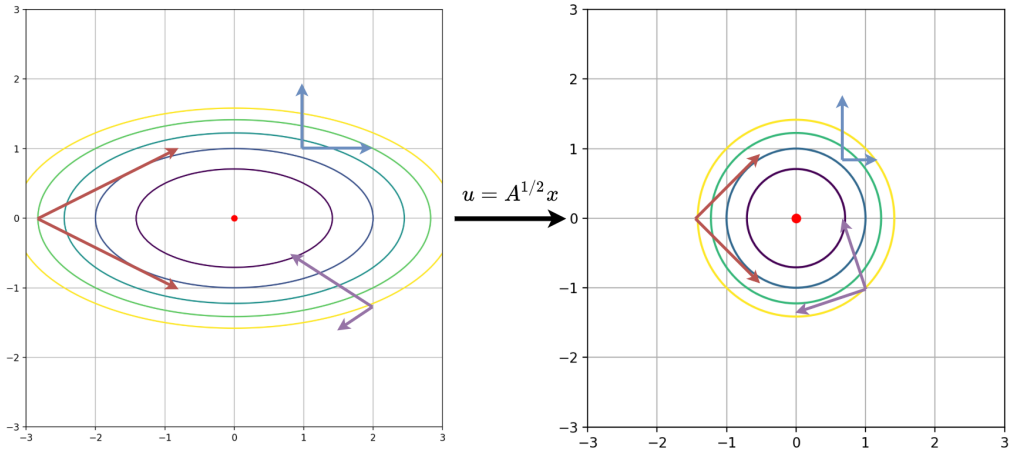


Figure 6: Conjugate vectors

4 Steepest Descent and Conjugate Directions

When applied to quadratic programming, steepest descent, combined with conjugate directions, has very interesting property and this can be seen with a change of coordinate. This section will be divided into two columns, the left column is what we do in the original coordinate system, the right column is what happens in the new coordinate system $u = A^{1/2}x$.

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

x^* solves $Ax = b$.

$$\begin{aligned} e_{(i)} &= x_{(i)} - x^* \\ \nabla f(x) &= Ax - b \\ r_{(i)} &= A(x^* - x_{(i)}) = -Ae_{(i)} \\ &= -Ax_{(i)} + b = -\nabla f(x_{(i)}) \end{aligned}$$

$$\tilde{f}(x) = \frac{1}{2}u^T u - (A^{-1/2}b)^T u + c$$

u^* solves $Iu^* = A^{-1/2}b$

$$\begin{aligned} \tilde{e}_{(i)} &= u_{(i)} - u^* \\ \nabla \tilde{f}(u) &= u - A^{-1/2}b \\ \tilde{r} &= I(u^* - u_{(i)}) = -I\tilde{e}_{(i)} \\ &= A^{-1/2}b - u_{(i)} = -\nabla \tilde{f}(u_{(i)}) \end{aligned}$$

When we search in direction of $d_{(i)}$ in the original coordinate system, we are searching in direction $A^{1/2}d_{(i)}$ in the new coordinate system. And we step in the searching direction with a step size $\alpha_{(i)}$ such that the resulting point minimize the function along this direction. The step size can be calculated by setting the gradient to 0.

$$\begin{aligned} x_{(i+1)} &= x_{(i)} + \alpha_{(i)}d_{(i)} \\ \frac{d}{d\alpha_{(i)}} f(x_{(i)} + \alpha_{(i)}d_{(i)}) &= 0 \\ \nabla f(x_{(i)} + \alpha_{(i)}d_{(i)})^T d_{(i)} &= 0 \\ e_{(i+1)}^T Ad^{(i)} &= 0 \end{aligned}$$

$$\begin{aligned} u_{(i+1)} &= u_{(i)} + \alpha_{(i)}A^{1/2}d_{(i)} \\ \frac{d}{d\alpha_{(i)}} \tilde{f}(u_{(i)} + \alpha_{(i)}A^{1/2}d_{(i)}) &= 0 \\ \nabla \tilde{f}(u_{(i)} + \alpha_{(i)}A^{1/2}d_{(i)})^T (A^{1/2}d_{(i)}) &= 0 \\ \tilde{e}_{(i+1)}^T A^{1/2}d_{(i)} &= 0 \end{aligned}$$

Let's translate the above math into English. When we go from $x_{(i)}$ in the direction $d_{(i)}$, we are going from $u_{(i)}$ in the direction $A^{1/2}d_{(i)}$ in the new coordinate system. If $x^{(i)} + \alpha_{(i)}d_{(i)}$ minimizes f in the direction of $d_{(i)}$, then $u_{(i)} + \alpha_{(i)}A^{1/2}d_{(i)}$ minimizes $\tilde{f}$ in the direction of $A^{1/2}d_{(i)}$ (since it's a linear transformation). However, in the new coordinate system, because the contour lines of $\tilde{f}$ are circles/balls, the resulting error $\tilde{e}_{(i+1)} = A^{1/2}e_{(i+1)}$ is guaranteed to be orthogonal to the previous searching direction $A^{1/2}d_{(i)}$. Thus we can know that $e_{(i+1)}^T Ad^{(i)} = 0$.

In other words, in the original coordinate system, when we minimize the function along a direction $d_{(i)}$, the resulting error $e_{(i+1)}$ and searching direction $d_{(i)}$ are A-orthogonal, or say, conjugate, as shown in the following picture.

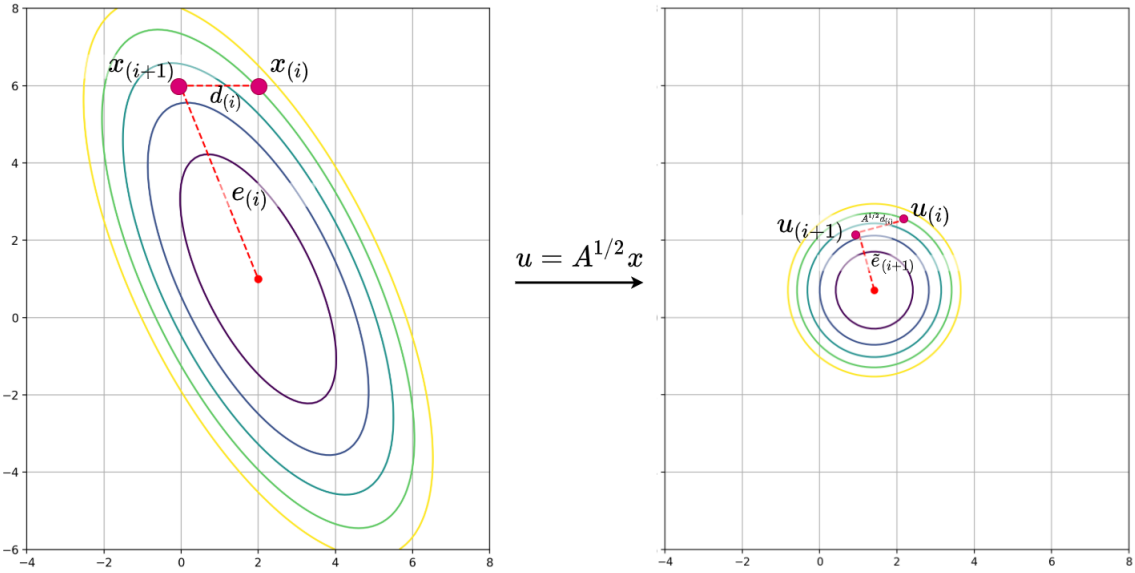


Figure 7: Steepest descent under linear transformation

5 Almost Conjugate Gradient Method

In the previous section, we see that when we apply steepest descent to a quadratic function, in the new coordinate system, the resulting error is orthogonal to the searching direction. That means, in the new coordinate system, after minimizing along a direction, we don't need to go into that direction again, or say, the later searching directions are gonna be orthogonal to the current searching direction.

Thus, if we have n orthogonal vectors that spans the space of new coordinate system, where n is the dimension of the space, then we can iteratively minimize over each of these directions. In other word, we need a group of vectors $d_{(i)}$ in the original coordinate system that are A-orthogonal.

Instead of having n A-orthogonal vectors at the beginning, we can gradually build up a group of them by Gram-Schmidt conjugation. Let me brutally show you the "almost conjugate gradient method" and then explain it.

$$d_{(0)} = r_{(0)} \tag{5.1}$$

$$d_{(i)} = r_{(i)} - \sum_{j=0}^{i-1} \frac{d_{(j)}^T A r_{(i)}}{d_{(j)}^T A d_{(j)}} d_{(j)} \tag{5.2}$$

$$\alpha_{(i)} = \frac{r_{(i)}^T d_{(i)}}{d_{(i)}^T A d_{(i)}} \tag{5.3}$$

$$x_{(i+1)} = x^{(i)} + \alpha_{(i)} d_{(i)} \tag{5.4}$$

At first step, we perform steepest descent along the direction of negative gradient $r_{(0)}$. At each following step at $x_{(i)}$, we evaluate the negative gradient $r_{(i)}$ and build $d_{(i)}$ which is A-orthogonal to

all previous searching directions. First, we take $r_{(i)}$ and previous searching directions $d_{(j)}(j < i)$ to the new coordinate system, we get

$$\tilde{r}_{(i)} = A^{1/2} r_{(i)} \tag{5.5}$$

$$\tilde{d}_{(j)} = A^{1/2} d_{(j)} \qquad (j < i) \tag{5.6}$$

Then we project $\tilde{r}_{(i)}$ on each of $\tilde{d}_{(j)}$, the projection of $\tilde{r}_{(i)}$ on $\tilde{d}_{(j)}$ is

$$\begin{aligned} P_{\tilde{d}_{(j)}}(\tilde{r}_{(i)}) &= \tilde{d}_{(j)} \left[\tilde{d}_{(j)}^T \tilde{d}_{(j)} \right]^{-1} \tilde{d}_{(j)}^T \tilde{r}_{(i)} \\ &= A^{1/2} d_{(j)} \left[(A^{1/2} d_{(j)})^T (A^{1/2} d_{(j)}) \right]^{-1} (A^{1/2} d_{(j)})^T A^{1/2} r_{(i)} \\ &= \frac{d_{(j)}^T A r_{(i)}}{d_{(j)}^T A d_{(j)}} A^{1/2} d_{(j)} \end{aligned} \tag{5.7}$$

then we subtract $\tilde{r}_{(i)}$ by it's projection on all $\tilde{d}_{(j)}$ such that $j < i$, then the remaining part is surly orthogonal to all previous $\tilde{d}_{(j)}$ (in the new coordinate system)

$$\begin{aligned} \tilde{d}_{(i)} &= \tilde{r}_{(i)} - \sum_{j=0}^{i-1} \frac{d_{(j)}^T A r_{(i)}}{d_{(j)}^T A d_{(j)}} A^{1/2} d_{(j)} \\ &= A^{1/2} r_{(i)} - \sum_{j=0}^{i-1} \frac{d_{(j)}^T A r_{(i)}}{d_{(j)}^T A d_{(j)}} A^{1/2} d_{(j)} \end{aligned} \tag{5.8}$$

We have $\tilde{d}_{(i)}$ orthogonal to all previous $\tilde{d}_{(j)}$ in the new coordinate system. The next thing to do is to bring $\tilde{d}_{(i)}$ to the original coordinate system

$$\begin{aligned} d_{(i)} &= A^{-1/2} \tilde{d}_{(i)} \\ &= r_{(i)} - \sum_{j=0}^{i-1} \frac{d_{(j)}^T A r_{(i)}}{d_{(j)}^T A d_{(j)}} d_{(j)} \end{aligned}$$

thus we get (5.2).

Since $\tilde{d}_{(i)}$ is orthogonal to all $\tilde{d}_{(j)}$ in new coordinate system, $d_{(i)}$ is A-orthogonal to. all $d_{(j)}$ in the original system.

Now we have the searching direction $d_{(i)}$, what we need is the step size for steepest descent $\alpha_{(i)}$, which we can calculate by setting the gradient to 0.

$$\begin{aligned} \frac{d}{d\alpha_{(i)}} f(x_{(i)} + \alpha_{(i)} d_{(i)}) &= 0 \\ \nabla f(x_{(i)} + \alpha_{(i)} d_{(i)})^T d_{(i)} &= 0 \\ \left[A(x_{(i)} + \alpha_{(i)} d_{(i)}) - b \right]^T d_{(i)} &= 0 \\ -r_{(i)}^T d_{(i)} + \alpha_{(i)} d_{(i)}^T A d_{(i)} &= 0 \\ \alpha_{(i)} &= \frac{r_{(i)}^T d_{(i)}}{d_{(i)}^T A d_{(i)}} \end{aligned}$$

thus we get (5.3).

Figure 8 and Figure 9 show progress of the "almost conjugate gradient" with different starting points.

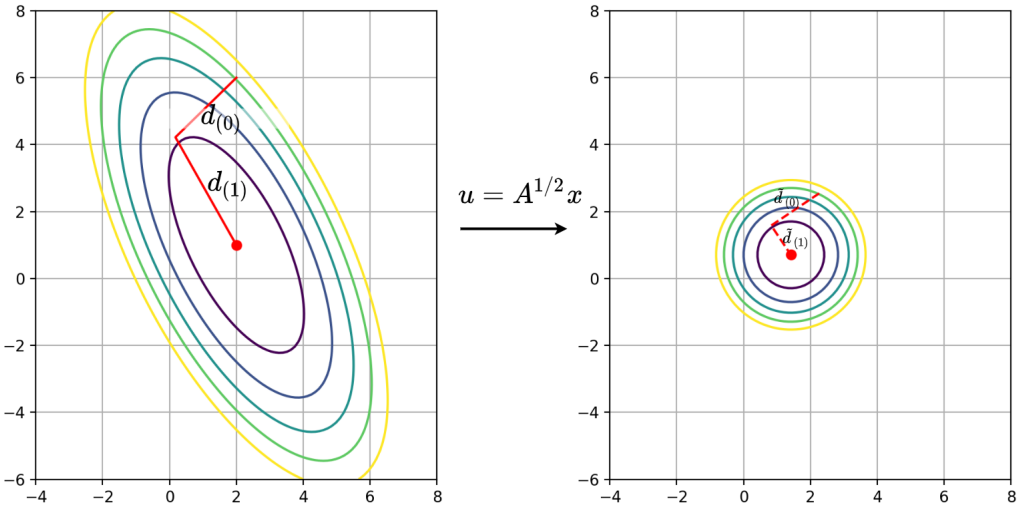


Figure 8: Almost conjugate gradient method

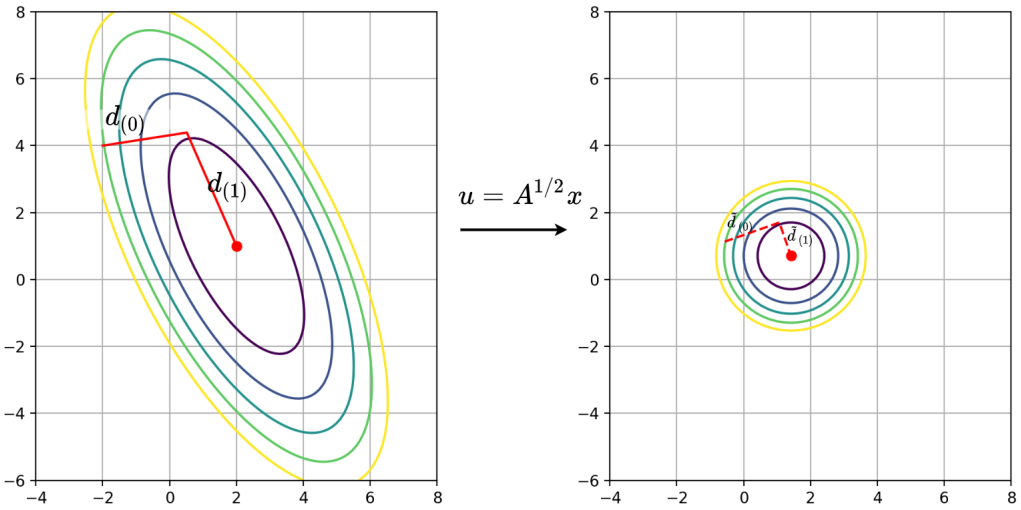


Figure 9: Almost conjugate gradient method

As you can see, applying steepest descent with conjugate directions in the original coordinate system is equivalent to applying steepest descent with orthogonal directions in the new coordinate system, and the reason we can do this in the new coordinate system is that the contour lines in the new coordinate system are circles and resulting error of steepest descent is guaranteed to be orthogonal to previous searching direction.

6 Simplifing "Almost CG Method"

If you have understand the "almost conjugate gradient method", you now get the core idea of the conjugate gradient method. Now we are one step away from the real conjugate gradient method, and that step is to reduce the complexity of the "almost conjugate gradient method". One thing you may notice that when we build $d_{(i)}$ according to (5.2), we must use all previous $d_{(j)}$ so that we must keep all $d_{(j)}$ in memory and that can be a large storage. However, by exploring the properties of spaces spanned by all $d_{(j)}$ and by $r_{(i)}$, we can simplify the process of building $d_{(i)}$.

Suppose $D_i = \text{span}\{d_{(0)}, d_{(1)}, ..., d_{(i)}\}$ is the space spanned by first i searching directions and $\text{span}\{r_{(0)}, r_{(1)}, ..., r_{(i)}\}$ is the space spanned by first i residuals. Now we can build the fact that $\text{span}\{r_{(0)}, r_{(1)}, ..., r_{(i)}\} = D_i$ using mathematical induction.

$$
\begin{aligned}
&\text{if } \ \text{span}\{r_0, ..., r_{(i)}\} = \text{span}\{d_{(0)}, d_{(1)}, ..., d_{(i)}\} = D_i \\
&\because d_{(i+1)} = r_{(i+1)} - \sum_{j=0}^{i} \frac{d_{(j)}^T A r_{(i+1)}}{d_{(j)}^T A d_{(j)}} d_{(j)} \\
&\therefore \text{span}\{d_{(0)}, ..., d_{(i+1)}\} = \text{span}\{d_{(0)}, ..., d_{(i)}, r_{(i+1)}\} \\
&\qquad\qquad\qquad\qquad = \text{span}\{D_i, r_{(i+1)}\} \\
&\qquad\qquad\qquad\qquad = \text{span}\{r_{(0)}, ..., r_{(i)}, r_{(i+1)}\} \\
&\because d_{(0)} = r_{(0)} \\
&\therefore \text{span}\{d_{(0)}\} = \text{span}\{r_{(0)}\} \\
&\therefore \text{span}\{d_{(0)}, d_{(1)}, ..., d_{(i)}\} = \text{span}\{r_0, ..., r_{(i)}\} \ \forall \ i
\end{aligned}
$$

Based on this conclusion, we can get more equations which can help us to simplify the "almost conjugate gradient method".

First, since all search directions in the transformed coordinate system $\tilde{d}_{(k)}$ are orthogonal to each other, the error in transformed coordinate system can be written as $\tilde{e}_{(j)} = \sum_{k=j}^{n-1} \delta_{(k)} \tilde{d}_{(k)}$. Taking this to the original coordinate system, we can write $e_{(j)} = \sum_{k=j}^{n-1} \delta_{(k)} d_{(k)}$. Multiply both sides by $d_{(i)}^T A$ where $i < j$, we have

$$
d_{(i)}^T A e_{(j)} = \sum_{k=j}^{n-1} \delta_{(k)} d_{(i)}^T A d_{(k)} = 0 \tag{6.1}
$$

since $d_{(i)}$'s are A-orthogonal, $d_{(i)}^T A d_{(k)}$ unless $i = k$. Thus we have

$$
\begin{aligned}
&d_{(i)}^T A e_{(j)} = 0 \quad (i < j) \\
&d_{(i)}^T r_{(j)} = 0 \quad (i < j) \qquad (6.2) \\
&\therefore r_{(j)} \perp \text{span}\{d_{(0)}, d_{(1)}, d_{(i)} | i < j\} \\
&\because \text{span}\{d_{(0)}, d_{(1)}, ..., d_{(i)}\} = \text{span}\{r_0, ..., r_{(i)}\} \ \forall \ i \\
&\therefore r_{(j)} \perp \text{span}\{r_{(0)}, r_{(1)}, r_{(i)} | i < j\} \\
&\therefore r_{(i)} \perp r_{(j)} \forall \ \ i \neq j \qquad (6.3)
\end{aligned}
$$

Once we have $r_{(i)} \perp r_{(j)} \forall \;\; i \neq j$, we multiply both sides of (5.2) by $r_{(i)}^T$, we have

$$\begin{aligned} r_{(i)}^T d_{(i)} &= r_{(i)}^T r_{(i)} - \sum_{j=0}^{i-1} \frac{d_{(j)}^T A r_{(i)}}{d_{(j)}^T A d_{(j)}} r_{(i)}^T d_{(j)} \\ r_{(i)}^T d_{(i)} &= r_{(i)}^T r_{(i)} \end{aligned} \tag{6.4}$$

7 Finally, Conjugate Gradient Method

In the previous section, by exploiting the spaces spanned by $d_{(i)}$'s and by $r_{(i)}$'s, we get two identities

1. $r_{(i)} \perp r_{(j)} \forall \;\; i \neq j$
2. $r_{(i)}^T d_{(i)} = r_{(i)}^T r_{(i)}$

Now we use these two identities to simplify the "almost conjugate gradient method" and get the real conjugate gradient method. First, by the definition of the residual, we have

$$\begin{aligned} r_{(j+1)} &= b - A x_{(j+1)} \\ &= b - A(x_{(j)} + \alpha_{(j)} d_{(j)}) \\ &= r_{(j)} - \alpha_{(j)} A d_{(j)} \end{aligned} \tag{7.1}$$

and multiply both sides of (7.1) by $r_{(i)}^T$

$$\begin{aligned} r_{(i)}^T r_{(j+1)} &= r_{(i)}^T r_{(j)} - \alpha_{(j)} r_{(i)}^T A d_{(j)} \\ r_{(i)}^T A d_{(j)} &= \frac{r_{(i)}^T r_{(j)} - r_{(i)}^T r_{(j+1)}}{\alpha_{(j)}} \\ &= \begin{cases} \frac{r_{(i)}^T r_{(i)}}{\alpha_{(i)}} & i = j \\ -\frac{r_{(i)}^T r_{(i)}}{\alpha_{(i-1)}} & i = j+1 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{7.2}$$

From (7.2) we can see that most term in the summation in (5.2) appears to be 0. By plugging (7.2) into (5.2), we get

$$\begin{aligned} d_{(i)} &= r_{(i)} - \sum_{j=0}^{i-1} \frac{d_{(j)}^T A r_{(i)}}{d_{(j)}^T A d_{(j)}} d_{(j)} \\ &= r_{(i)} - \frac{d_{(i-1)}^T A r_{(i)}}{d_{(i-1)}^T A d_{(i-1)}} d_{(i-1)} \\ &= r_{(i)} + \frac{r_{(i)}^T r_{(i)}}{\alpha_{(i-1)} d_{(i-1)}^T A d_{(i-1)}} d_{(i-1)} \end{aligned} \tag{7.3}$$

then plug (5.3) into (7.3), we get

$$\begin{aligned} d_{(i)} &= r_{(i)} + \frac{r_{(i)}^T r_{(i)}}{\alpha_{(i-1)} d_{(i-1)}^T A d_{(i-1)}} d_{(i-1)} \\ &= \frac{r_{(i)}^T r_{(i)}}{r_{(i-1)}^T d_{(i-1)}} d_{(i-1)} \\ &= \frac{r_{(i)}^T r_{(i)}}{r_{(i-1)}^T r_{(i-1)}} d_{(i-1)} \end{aligned} \tag{7.4}$$

Thus we have a new update rule for $d_{(i)}$ and $\alpha_{(i)}$, and this lead us to the final conjugate gradient method

$$d_{(0)} = r_{(0)}$$

$$d_{(i)} = \frac{r_{(i)}^T r_{(i)}}{r_{(i-1)}^T r_{(i-1)}} d_{(i-1)} \tag{7.5}$$

$$\alpha_{(i)} = \frac{r_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}} \tag{7.6}$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)} \tag{7.7}$$

Notice that this is basicly the same thing as "almost conjugate gradient method" except that it save some computation by exploiting the innate property of the relationship between $d_{(i)}$'s and $r_{(i)}$'s. Figure 10 and Figure 11 shows the progress of conjugate gradient method on the running example with different starting points. They are almost the same as Figure 8 and Figure 9 because these two algorithms are doing the same thing, but with different complexities.

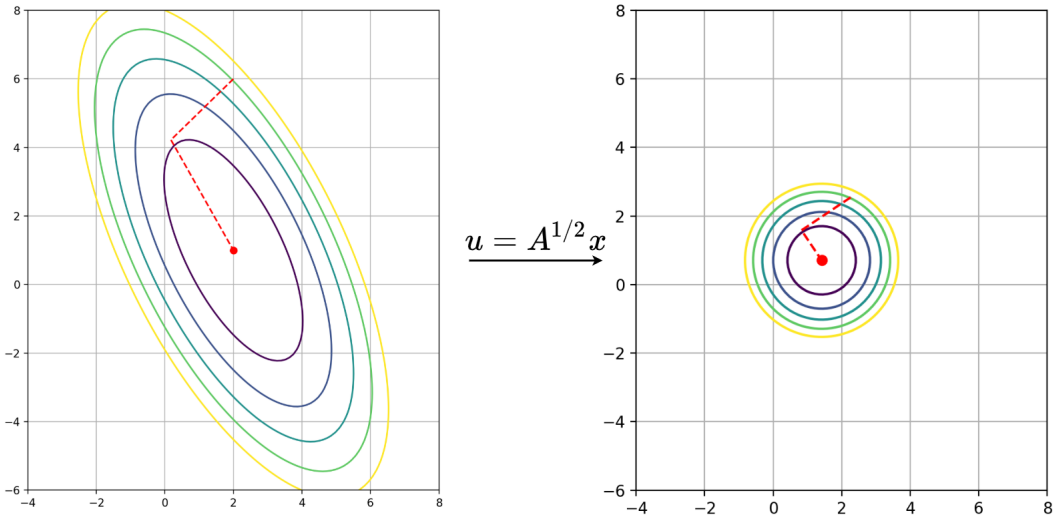


Figure 10: Conjugate gradient method

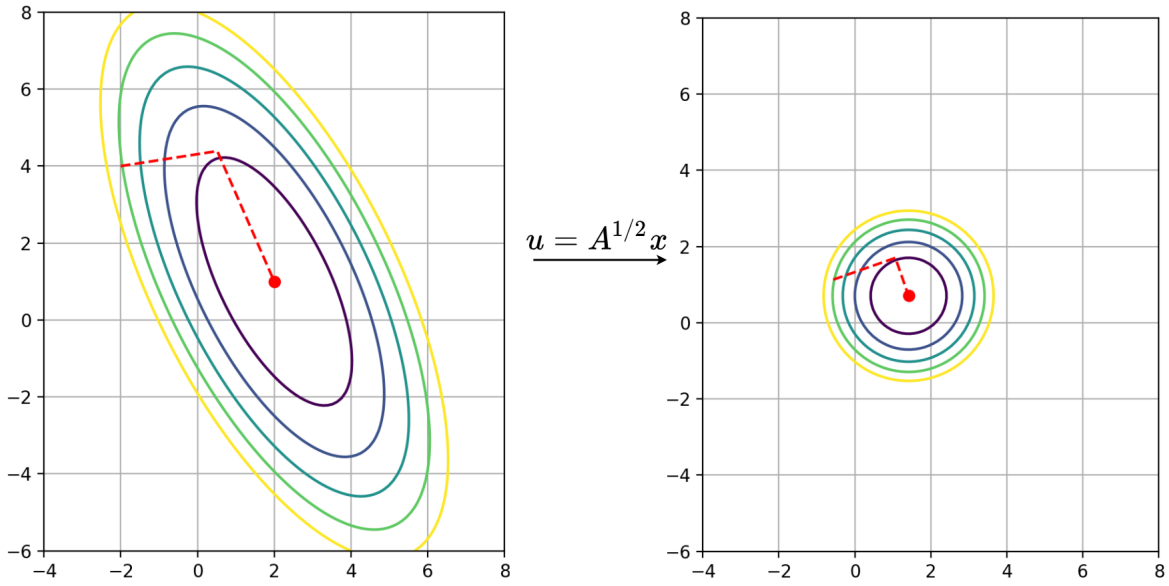


Figure 11: Conjugate gradient method

Figure 12 shows the residual norm and error norm when apply conjugate gradient method to a 2000x2000 system.

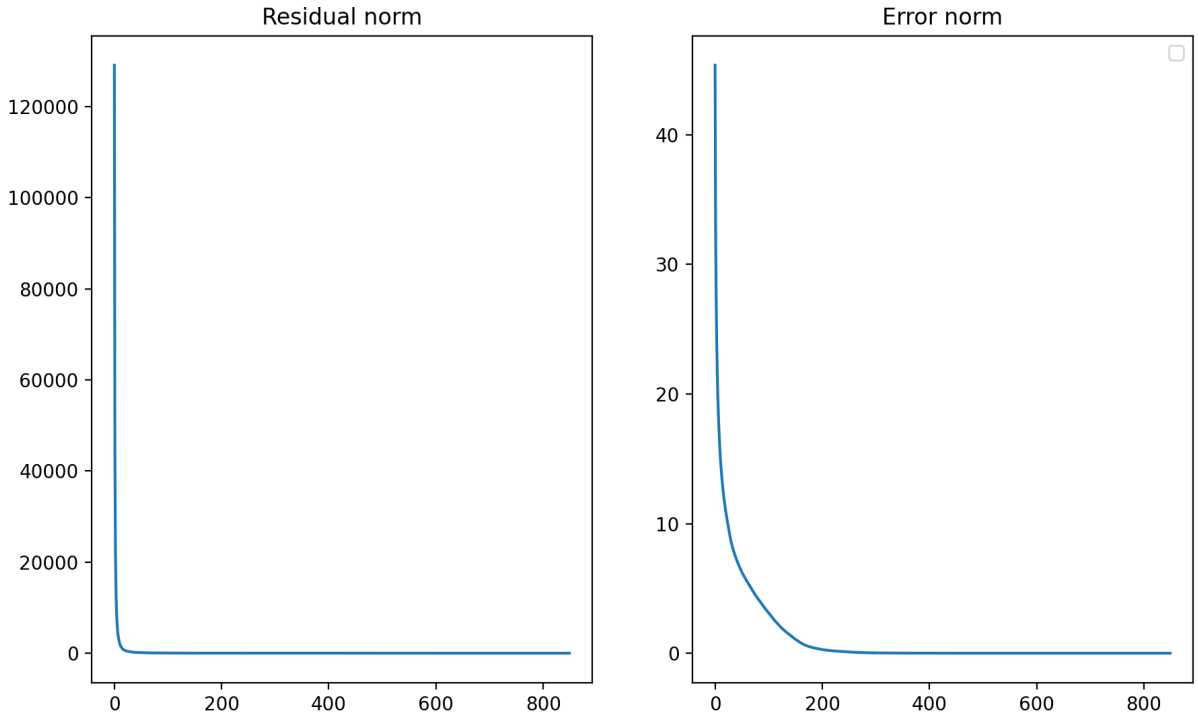


Figure 12: Conjugate gradient method on 2000x2000 system

8 Conclusion

This note is an introduction to the intricate conjugate gradient descent. Hopefully I have made the core idea of cg method clear to you. However, there are a lot of things about cg method that's not included in this note like convergence analysis of cg method, krylov space and spectral analysis of krylov space (this is available in EE364b).

As you can see, the awfully complicated conjugate gradient method is just steepest descent with orthogonal searching direction under a change of coordinate. Actually, not just in cg method, the technique of changing coordinate applies everywhere in math and engineering. Some examples are listed

- Gradient descent + changing coordinate (according to ellipsoid induced by local hessian) = Newton's method
- Cutting-plane method + changing coordinate (according to ignorance ellipsoid) = Ellipsoid method
- Fourier transform of radial function+ changing coordinate (cartesian to polar) = Radon Transform

The philosophy here is, many sophisticated things are just combination of many simple but clever things. The hard part is what simples things to use and how to assemble these simple things. The problem itself, in some sense, gives you hints about what technique to use(changing coordinates, Fourier analysis, etc) with its properties (asymmetric, radial, periodic, etc. In the example of cg method, we change coordinate to turn an "inhomogeneous" ellipsoid into a "homogeneous" circle/ball). In other words, the solution to a problem is right there in the problem itself, however, identifying those properties and correctly applying those techniques to the very specific problem is where being smart makes the difference.