# Conjugate Gradients explained

Cenek Albl

September 5, 2012

## 1 Motivation

Suppose we have a system of linear equations with a single solution described as

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{1}$$

where $\mathbf{A}$ is a known $n \times n$ real matrix, $\mathbf{b}$ is a known $n \times 1$ vector and $\mathbf{x}$ is an $n \times 1$ vector of unknowns. We wish to find $\mathbf{x}_*$ which is the solution to (1). To find $\mathbf{x}_*$, we can employ one of the direct methods, for example the Cholesky or LU decomposition, whose computation comp lexity is roughly $O(n^3)$. For smaller $\mathbf{A}$, this is a feasible approach. However, as $\mathbf{A}$ grows larger, finding the exact solution by direct methods becomes time consuming. In some cases we don't necessarily need the exact solution, but an estimate which is close to the solution is sufficient.

Bundle adjustment (BA) is one of such cases. In BA, solving of (1) is used at each iteration of the Levenberg-Marquardt algorithm which is by itself a local iterative algorithm working with an approximation to a non-linear function. By using an iterative method to solve (1) within BA, we can choose the number of iterations and therefore the precision of the approximation to the solution. Depending on the speed of convergence of the iterative method, the trade-off between quality of the solution and the speed of the computation can be highly in the favor of stopping the iterative method earlier.

A method of Conjugate Gradients (CG) is an iterative method, which seeks a minimum of a function. To find $\mathbf{x}_*$, we would like to have a function of $\mathbf{x}$, whose global minimum is the solution to (1). Such function can, for instance, be the following quadratic form

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

Now if $\mathbf{A}$ is symmetric and positive-definite, we can use CG to find $\mathbf{x}$ in $n$ iterations as will be shown next.

### 1.1 Why symmetric A?

The local extrema of (2) can be found by setting $f'(\mathbf{x}) = 0$. It can be derived that

$$f'(\mathbf{x}) = \frac{1}{2}\mathbf{A}^T\mathbf{x} + \frac{1}{2}\mathbf{A}\mathbf{x} - \mathbf{b} \tag{3}$$

If $\mathbf{A}$ is symmetric, i.e. $\mathbf{A}^T = \mathbf{A}$, then (3) becomes

$$f'(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b} \tag{4}$$

and by setting $f'(\mathbf{x})$ equal to zero we obtain our original system of linear equations (1).

### 1.2 Why positive-definite?

We saw that $f(\mathbf{x})$ has a critical point in $\mathbf{x}_*$. But when does $f(\mathbf{x})$ have a unique global minimum in $\mathbf{x}_*$? It is when $\mathbf{A}$ is positive definite, i.e. when

$$\mathbf{x}^T\mathbf{A}\mathbf{x} > 0 \tag{5}$$

for every non-zero $\mathbf{x}$. Let us see why. Let $\mathbf{e} = \mathbf{x} - \mathbf{x}_*$, i.e. it is the difference between the solution $\mathbf{x}_*$ and its current estimate $\mathbf{x}$. Then

$$\begin{aligned} f(\mathbf{x}_* + \mathbf{e}) &= \tfrac{1}{2}(\mathbf{x}_* + \mathbf{e})^T\mathbf{A}(\mathbf{x}_* + \mathbf{e}) - \mathbf{b}^T(\mathbf{x}_* + \mathbf{e}) + c \\ &= \tfrac{1}{2}\mathbf{x}_*^T\mathbf{A}\mathbf{x}_* + \mathbf{e}^T\mathbf{A}\mathbf{x}_* + \tfrac{1}{2}\mathbf{e}^T\mathbf{A}\mathbf{e} - \mathbf{b}^T\mathbf{x}_* - \mathbf{b}^T\mathbf{e} + c \\ &= \tfrac{1}{2}\mathbf{x}_*^T\mathbf{A}\mathbf{x}_* - \mathbf{b}^T\mathbf{x}_* + \mathbf{e}^T\mathbf{b} - \mathbf{b}^T\mathbf{e} + \tfrac{1}{2}\mathbf{e}^T\mathbf{A}\mathbf{e} + c \\ &= \tfrac{1}{2}\mathbf{x}_*^T\mathbf{A}\mathbf{x}_* - \mathbf{b}^T\mathbf{x}_* + c + \mathbf{e}^T\mathbf{b} + \tfrac{1}{2}\mathbf{e}^T\mathbf{A}\mathbf{e} - \mathbf{b}^T\mathbf{e} \\ &= f(\mathbf{x}_*) + \tfrac{1}{2}\mathbf{e}^T\mathbf{A}\mathbf{e} \end{aligned} \tag{6}$$

If $\mathbf{A}$ is positive-definite, the term $\frac{1}{2}\mathbf{e}^T\mathbf{A}\mathbf{e}$ will be always positive except when $\mathbf{e} = \mathbf{0}$ in which case we are at the solution already. We see that solving (1) for symmetric positive definite matrices $\mathbf{A}$ can be done by finding the global minimum of (2) by a local method.

## 2 Method

To solve (1), we will start at some arbitrary point $\mathbf{x}_0$ in $R^n$ and iteratively take updates to find the solution $\mathbf{x}_*$. Let's first define some useful quantities. Let $\mathbf{x}_0$ be an initial estimate of $\mathbf{x}_*$. At each step $k$, $k = 0, \ldots, n-1$ let $\mathbf{x}_k$ be our current estimate of $\mathbf{x}_*$. Then we have an error term

$$\mathbf{e}_k = \mathbf{x}_k - \mathbf{x}_* \tag{7}$$

and the residual

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k \tag{8}$$

which is also minus the gradient of (2). By substituting from (7) to (8) we obtain

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}(\mathbf{e}_k + \mathbf{x}_*) = \mathbf{A}\mathbf{x}_* + \mathbf{b} - \mathbf{A}\mathbf{e}_k = -\mathbf{A}\mathbf{e}_k \tag{9}$$

representing the relationship between the error and the residual.

## 2.1 Finding the solution equals eliminating the error

Let's define a sequence of $n$ linearly independent directions

$$\mathbf{D} = (\mathbf{d}_0, ..., \mathbf{d}_{n-1}) \tag{10}$$

in which we will take steps towards the solution. After each step we have taken in one of those directions, we arrive at a new value of $\mathbf{x}$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \beta_k \mathbf{d}_k, \quad k = 0, \ldots, n-1 \tag{11}$$

and because $\mathbf{D}$ forms a basis in $R^n$, there exists a sequence of coefficients

$$\mathbf{\Phi} = (\beta_0, ..., \beta_{n-1}) \tag{12}$$

such that we can write any $\mathbf{x}_*$ as

$$\mathbf{x}_* = \mathbf{x}_0 + \sum_{i=0}^{n-1} \beta_i \mathbf{d}_i \tag{13}$$

and by finding the coefficients $\mathbf{\Phi}$ we can also find the solution $\mathbf{x}_*$. From (7) and (13) we can express the error term as a linear combination of $\mathbf{D}$

$$\mathbf{e}_0 = -\sum_{i=0}^{n-1} \beta_i \mathbf{d}_i \tag{14}$$

Our goal is to eliminate the initial error $\mathbf{e_0}$, which we can do by finding appropriate $\mathbf{\Phi}$ and then taking steps $\beta_k$ to eliminate all components of $\mathbf{e_0}$ and therefore arriving at $\mathbf{x}_*$. It can be seen that in order to obtain $\mathbf{x}_*$ only $n$ components of error must be eliminated and we will show later that the algorithm finishes in $n$ iterations.

## 2.2 Orthogonal directions

To make our search for $\mathbf{\Phi}$ easier, we can choose the directions $\mathbf{d}_0, \mathbf{d}_1, ..., \mathbf{d}_{n-1}$ to be mutually orthogonal, i.e.

$$\mathbf{d}_i^T \mathbf{d}_j = 0 \tag{15}$$

for all $i \neq j$, and use the orthogonality to make the computation of each coefficient $\beta_k$ independent on the values of other coefficients. If we multiply both sides of (14) by $\mathbf{d}_k^T$ from the left we get

$$\mathbf{d}_k^T \mathbf{e}_0 = -\sum_{i=0}^{n-1} \beta_i \mathbf{d}_k^T \mathbf{d}_i, \quad k = 0, \ldots, n-1 \tag{16}$$

where the elements of the sum are zero for all $i \neq k$. Therefore

$$\mathbf{d}_k^T \mathbf{e}_0 = -\beta_k \mathbf{d}_k^T \mathbf{d}_k \tag{17}$$

$$\beta_k = -\frac{\mathbf{d}_k^T \mathbf{e}_0}{\mathbf{d}_k^T \mathbf{d}_k}, \quad k = 0, \ldots, n-1 \tag{18}$$

At a step $k$, we already took steps $0, \ldots, k-1$ and the current error is now

$$\mathbf{e}_k = \mathbf{e}_0 + \sum_{i=0}^{k-1} \beta_i \mathbf{d}_i \quad k = 0, \ldots, n-1 \tag{19}$$

We can express $\mathbf{e}_0$ from (19) and substitute it into (18) to obtain

$$\beta_k = -\frac{\mathbf{d}_k^T (\mathbf{e}_k - \sum_{i=0}^{k-1} \beta_i \mathbf{d}_i)}{\mathbf{d}_k^T \mathbf{d}_k} \tag{20}$$

$$\beta_k = -\frac{\mathbf{d}_k^T \mathbf{e}_k - \sum_{i=0}^{k-1} \beta_i \mathbf{d}_k^T \mathbf{d}_i}{\mathbf{d}_k^T \mathbf{d}_k} \tag{21}$$

where the sum in (21) is zero because $i$ never equals $k$. We have derived an expression for each step length

$$\beta_k = -\frac{\mathbf{d}_k^T \mathbf{e}_k}{\mathbf{d}_k^T \mathbf{d}_k}, \quad k = 0, \ldots, n-1 \tag{22}$$

which depends only on the current direction $\mathbf{d}_k$ and the current value of error $\mathbf{e}_k$. Unfortunately, we don't know the value of $\mathbf{e}_k$, so how do we find $\beta_k$? Let's focus on what we do know and that is residual $\mathbf{r}_k$. From (9) we see that the residual is actually the error transformed by matrix $\mathbf{A}$.

## 2.3 Conjugate directions

We will now use matrix $\mathbf{A}$ to construct vectors $\mathbf{d}_0, \ldots, \mathbf{d}_{n-1}$ as $A$-orthogonal (conjugate) with respect to each other, i.e.

$$\mathbf{d}_j^T \mathbf{A} \mathbf{d}_i = 0 \tag{23}$$

for $i \neq j$. Let us see that we can construct such a set of vectors. Since $\mathbf{A}$ is symmetric and positive-definite, there exists a real nonsingular matrix $\mathbf{B}$ such that

$$\mathbf{B}\mathbf{B}^T = \mathbf{A} \tag{24}$$

Let's say we have two bases $\gamma$ and $\gamma'$. We will further denote vectors $\mathbf{d}_i$ in those two bases as $\mathbf{d}_{i\gamma}$ and $\mathbf{d}_{i\gamma'}$ respectively. Let $\mathbf{B}^T$ be a matrix transforming coordinates of a vector from basis $\gamma$ to basis $\gamma'$, e.g. $\mathbf{d}_{i\gamma'} = \mathbf{B}^T \mathbf{d}_{i\gamma}$. We can find $n$ orthogonal vectors in base $\gamma'$ such that $\mathbf{d}_{i\gamma'}^T \mathbf{d}_{j\gamma'} = 0$ for all $i \neq j$. If we write this equality in base $\gamma$ we see that

$$\mathbf{d}_{i\gamma'}^T \mathbf{d}_{j\gamma'} = 0$$
$$(\mathbf{B}^T \mathbf{d}_{i\gamma})^T \mathbf{B}^T \mathbf{d}_{j\gamma} = 0$$
$$\mathbf{d}_{i\gamma}^T \mathbf{B}\mathbf{B}^T \mathbf{d}_{j\gamma} = 0$$
$$\mathbf{d}_{i\gamma}^T \mathbf{A}\mathbf{d}_{j\gamma} = 0$$

The vectors which are orthogonal in basis $\gamma'$ are conjugate in basis $\gamma$. Assuming that the directions $\mathbf{d}_0, \ldots, \mathbf{d}_{n-1}$ are conjugate, we can multiply both sides of (14) by $\mathbf{A}$ from the left and then by $\mathbf{d}_k^T$ from the left to get

$$\mathbf{d}_k^T \mathbf{A}\mathbf{e}_0 = -\sum_{i=0}^{n-1} \beta_i \mathbf{d}_k^T \mathbf{A}\mathbf{d}_i \tag{25}$$

and following the same derivation as in (16)-(22) we get

$$\beta_k = -\frac{\mathbf{d}_k^T \mathbf{A}\mathbf{e}_k}{\mathbf{d}_k^T \mathbf{A}\mathbf{d}_k} \tag{26}$$

$$\beta_k = -\frac{\mathbf{d}_k^T (-\mathbf{r}_k)}{\mathbf{d}_k^T \mathbf{A}\mathbf{d}_k} \tag{27}$$

$$\beta_k = \frac{\mathbf{d}_k^T \mathbf{r}_k}{\mathbf{d}_k^T \mathbf{A}\mathbf{d}_k}, \quad k = 0, \ldots, n-1 \tag{28}$$

where everything is known once we create the set of directions $\mathbf{D}$.

## 2.4 Finding the conjugate directions

To find a set of conjugate directions, we can use *Gram-Schmidt orthogonalization* and slightly adjust it. In its original form, this method takes a linearly independent set $\mathbf{V} = \{\mathbf{v}_0, ..., \mathbf{v}_{n-1}\}$ in $R^n$ and produces an orthogonal set $\mathbf{U} = \{\mathbf{u}_0, ..., \mathbf{u}_{n-1}\}$ that spans $R^n$. To generate $\mathbf{U}$ we take $\mathbf{u}_0 = \mathbf{v}_0$ and calculate $\mathbf{u}_1, ..., \mathbf{u}_{n-1}$ as

$$\mathbf{u}_i = \mathbf{v}_i - \sum_{j=0}^{i-1} \delta_{ij} \mathbf{u}_j \quad i = 1, \ldots, n-1 \tag{29}$$

where $\delta_{ij}$ is not the Kronecker delta. This way we substract from each vector $\mathbf{v}_i$ all its components that are not orthogonal to the previously created vectors $\mathbf{u}_0, ..., \mathbf{u}_{i-1}$. To find the quantities $\delta_{ij}$, we can use the same approach as in section 2.2 and multiply each side of the transposed equation (29) by $\mathbf{u}_k$ from the right for all $k = 0, \ldots, i-1$

$$\mathbf{u}_i^T \mathbf{u}_k = \mathbf{v}_i^T \mathbf{u}_k - \sum_{j=0}^{i-1} \delta_{ij} \mathbf{u}_j^T \mathbf{u}_k \quad i = 1, \ldots, n-1 \text{ and } k = 0, \ldots, i-1 \tag{30}$$

$$0 = \mathbf{v}_i^T \mathbf{u}_k - \delta_{ik} \mathbf{u}_k^T \mathbf{u}_k \tag{31}$$

for all $i \neq k$ and

$$\delta_{ik} = \frac{\mathbf{v}_i^T \mathbf{u}_k}{\mathbf{u}_k^T \mathbf{u}_k} \quad i = 1, \ldots, n-1 \text{ and } k = 0, \ldots, i-1 \tag{32}$$

Ultimately, we obtain an orthogonal set of $n$ vectors. To get a set of $n$ $A$-orthogonal vectors, we simply impose this property in the calculation of $\delta_{ik}$, by multiplying by $\mathbf{A}\mathbf{u}_k$ instead of $\mathbf{u}_k$

$$\mathbf{u}_i^T \mathbf{A} \mathbf{u}_k = \mathbf{v}_i^T \mathbf{A} \mathbf{u}_k - \sum_{j=0}^{i-1} \delta_{ij} \mathbf{u}_j^T \mathbf{A} \mathbf{u}_k \tag{33}$$

$$\delta_{ik} = \frac{\mathbf{v}_i^T \mathbf{A} \mathbf{u}_k}{\mathbf{u}_k^T \mathbf{A} \mathbf{u}_k} \quad i = 1, \ldots, n-1 \text{ and } k = 0, \ldots, i-1 \tag{34}$$

The drawback of this method is that vectors $\mathbf{u}_0, \ldots, \mathbf{u}_{i-1}$ and $\mathbf{v}_0, \ldots, \mathbf{v}_{i-1}$ must be kept in memory in order to calculate $\mathbf{u}_i$ and the computational complexity for generating the full set is $O(n^3)$.

Fortunately, within the method of CG, the mentioned drawbacks can be solved. Let us consider using the residuals for creating the conjugate directions through the Gram-Schmidt process. From (9) we see that

$$\begin{aligned} \mathbf{r}_{k+1} &= -\mathbf{A}\mathbf{e}_{k+1} \\ &= -\mathbf{A}(\mathbf{e}_k + \beta_k \mathbf{d}_k) \\ &= \mathbf{r}_k - \beta_k \mathbf{A}\mathbf{d}_k \qquad k = 0, \ldots, n-1 \end{aligned} \tag{35}$$

This shows that after taking a step $\beta_k$, the new residual $\mathbf{r}_{k+1}$ will be a linear combination of the previous residual $\mathbf{r}_k$ and $\mathbf{A}\mathbf{d}_k$. Let's create an $A$-orthogonal direction $\mathbf{d}_k$ from the residuals $\mathbf{r}_0, \ldots, \mathbf{r}_k$, starting with $\mathbf{d}_0 = \mathbf{r}_0$. The Gram-Schmidt method guarantees that both $\mathbf{d}_0, \ldots, \mathbf{d}_k$ and $\mathbf{r}_0, \ldots, \mathbf{r}_k$ will span the same subspace for every k, i.e.

$$\mathbf{D}_k = \text{span}\{\mathbf{d}_0, \ldots, \mathbf{d}_k\} = \text{span}\{\mathbf{r}_0, \ldots, \mathbf{r}_k\} \quad k = 0, \ldots, n-1 \tag{36}$$

From (35) we see that $\mathbf{r}_{k+1} = \mathbf{r}_k - \beta_k \mathbf{A}\mathbf{d}_k$ and since $\mathbf{d}_k \in \mathbf{D}_k$ and $\mathbf{r}_k \in \mathbf{D}_k$ then

$$\begin{aligned} \mathbf{D}_{k+1} &= \text{span}\{\mathbf{D}_k, \mathbf{r}_{k+1}\} \\ &= \text{span}\{\mathbf{D}_k, \mathbf{r}_k - \beta_k \mathbf{A}\mathbf{d}_k\} \\ &= \text{span}\{\mathbf{D}_k, \mathbf{A}\mathbf{d}_k\} \end{aligned} \tag{37}$$

and since $\mathbf{D}_0 = \text{span}\{\mathbf{d}_0\} = \text{span}\{\mathbf{r}_0\}$ it follows that

$$\begin{aligned} \mathbf{D}_k &= \text{span}\{\mathbf{d}_0, \mathbf{A}\mathbf{d}_0, \mathbf{A}^2\mathbf{d}_0, \ldots, \mathbf{A}^{k-1}\mathbf{d}_0\} \\ &= \text{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0\} \quad k = 0, \ldots, n-1 \end{aligned} \tag{38}$$

which is called the *Krylov subspace*.

There is one more very important thing to notice. From equations (14) and (19) we can derive

$$\begin{aligned} \mathbf{e}_k &= \mathbf{e}_0 + \sum_{i=0}^{k-1} \beta_i \mathbf{d}_i \\ &= -\sum_{i=0}^{n-1} \beta_i \mathbf{d}_i + \sum_{i=0}^{k-1} \beta_i \mathbf{d}_i \\ &= -\sum_{i=k}^{n-1} \beta_i \mathbf{d}_i \quad k = 0, \ldots, n-1 \end{aligned} \tag{39}$$

which shows the error in terms of the remaining directions. We can use the $A$-orthogonality once again and multiply both sides of (39) by $\mathbf{d}_j^T \mathbf{A}$, $j = 1, \ldots, l-1$ from the left to get

$$\begin{aligned} -\mathbf{d}_j^T \mathbf{A} \mathbf{e}_k &= -\sum_{i=k}^{n-1} \beta_i \mathbf{d}_j^T \mathbf{A} \mathbf{d}_i \\ \mathbf{d}_j \mathbf{r}_k &= 0 \qquad j < k \end{aligned} \tag{40}$$

which tells us, that each residual $\mathbf{r}_k$ is always orthogonal to all previous search directions $\mathbf{d}_0, ..., \mathbf{d}_{k-1}$. In other words, since $\mathbf{d}_0, ..., \mathbf{d}_{k-1}$ span $\mathbf{D}_k$ and $\mathbf{r}_k$ is orthogonal to $\mathbf{d}_0, ..., \mathbf{d}_{k-1}$, it follows that $\mathbf{r}_k$ is also orthogonal to $\mathbf{D}_k$. The fact that $\text{span}\{\mathbf{D}_k\} = \text{span}\{\mathbf{D}_{k-1}, \mathbf{A}\mathbf{D}_{k-1}\}$ and the residual $\mathbf{r}_k$ is orthogonal to $\mathbf{D}_k$ implies that $\mathbf{r}_k$ is also $A$-orthogonal to $\mathbf{D}_{k-1}$.

That is an important property. If the residual is conjugate to $\mathbf{D}_{k-1}$, it means that it is already conjugate to all $\mathbf{d}_0, ..., \mathbf{d}_{k-2}$. Therefore, at each step, by calculating the residual we generate a vector which is already $A$-orthogonal to all previous search directions except $\mathbf{d}_{k-1}$. To obtain a search direction $\mathbf{d}_k$, which is conjugate to all previous ones, we can take the residual $\mathbf{r}_k$ and make it $A$-orthogonal to $\mathbf{d}_{k-1}$ and we are done. We only need to find one coefficient $\delta_{ik}$ which corresponds to $\mathbf{d}_{k-1}$ in the equation (33) and we will call it simply $\delta$

$$\begin{aligned} \mathbf{d}_k^T \mathbf{A} \mathbf{d}_{k-1} &= \mathbf{r}_k^T \mathbf{A} \mathbf{d}_{k-1} - \delta \mathbf{d}_{k-1}^T \mathbf{A} \mathbf{d}_{k-1} \\ \delta &= \frac{\mathbf{r}_k^T \mathbf{A} \mathbf{d}_{k-1}}{\mathbf{d}_{k-1} \mathbf{A} \mathbf{d}_{k-1}} \end{aligned} \tag{41}$$

To simplify this equation and to get rid of unnecessary multiplication, we can combine (35), (40), (28) and (42) to get

$$\delta_k = \frac{\mathbf{r}_k^T \mathbf{r}_k}{\mathbf{r}_{k-1} \mathbf{r}_{k-1}} \tag{42}$$

Having obtained $\delta_k$, we can eliminate the only part of $\mathbf{r}_k$ which is not conjugate to $\mathbf{D}_k$ and get $\mathbf{d}_k$

$$\mathbf{d}_k = \mathbf{r}_k - \delta \mathbf{d}_{k-1} \tag{43}$$

## 2.5 Putting it together

We have obtained all the necessary pieces to find the solution to (1) by an iterative method. To begin we set $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{d}_0 = \mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$. In each step $k$ we take the following actions. Find the length of the step using (28)

$$\beta_k = -\frac{\mathbf{d}_k^T \mathbf{r}_k}{\mathbf{d}_k^T \mathbf{A} \mathbf{d}_k}$$

Take the step according to (11) to find new estimate of $\mathbf{x}_*$

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

Calculate the new residual according to (35)

$$r_{k+1} = \mathbf{r}_k - \beta_k \mathbf{A} \mathbf{d}_k$$

Find the size of the non-conjugate component of $\mathbf{r}_{k+1}$ in the direction of $\mathbf{d}_{k-1}$ according to (42)

$$\delta_{k+1} = \frac{\mathbf{r}_{k+1}^T \mathbf{r}_{k+1}}{\mathbf{r}_k \mathbf{r}_k}$$

Substract the component from $\mathbf{r}_k$ to find a new search direction $\mathbf{d}_k$ according to (43)

$$\mathbf{d}_{k+1} = \mathbf{r}_{k+1} - \delta_{k+1} \mathbf{d}_k$$

After $n$ steps, the algorithm eliminates all components of $\mathbf{e}_0$ and the solution $\mathbf{x}_*$ is found. We can set an early-stopping criteria to reduce the computation time.