The Conjugate Gradient Method for Solving Systems of Linear Equations

Ove Edlund

LTU

2017-04-19

Symmetric positive definite systems of linear equations

The topic of this talk is solution of massively large systems of linear equations, where the system matrix is symmetric and positive definite (SPD).

A matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is **symmetric** if $\mathbf{A}^T = \mathbf{A}$

and **positive definite** if $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$

Methods for positive definite systems of linear equations

- For moderately sized problems, use full matrix representation and Cholesky factorization.
- For large scale problems with possible sparse matrix representation, use sparse Cholesky factorization, with row and column reordering to reduce fill in.
- For huge problems where even the sparse solvers struggle, use iterative methods for approximate solutions. The most common one is the Conjugate Gradient method, the subject of this talk.

Some properties of the Conjugate Gradient method

- For an $n \times n$ system, the exact solution is found in n steps. Though the method is never used in that way. Instead only a fraction of those iteration are done, to get an approximation of the solution.
- The method does not need to know anything about the entries of the system matrix. It only needs to know how to make a matrix times vector multiplication.

The Cayley-Hamilton Theorem

Theorem (Cayley-Hamilton)

Given the characteristic polynomial of a matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$

$$p(\lambda) = \det(\lambda\mathbf{I} - \mathbf{A}) = \lambda^n + c_{n-1}\lambda^{n-1} + \cdots + c_2\lambda^2 + c_1\lambda + c_0 \ ,$$

if an equivalent matrix-polynomial is formed with the same coefficients, and the polynomial is evaluated with the matrix $\mathbf{A}$, the result is the zero matrix

$$p(\mathbf{A}) = \mathbf{A}^n + c_{n-1}\mathbf{A}^{n-1} + \cdots + c_2\mathbf{A}^2 + c_1\mathbf{A} + c_0\mathbf{I} = \mathbf{0}$$

A consequence of the Cayley-Hamilton theorem

As a consequence of the Cayley-Hamilton theorem, if $\mathbf{A}$ is invertible, then $c_0 \neq 0$, and we can solve for $\mathbf{I}$ in the equality

$$\mathbf{A}^n + c_{n-1}\mathbf{A}^{n-1} + \cdots + c_2\mathbf{A}^2 + c_1\mathbf{A} + c_0\mathbf{I} = \mathbf{0},$$

followed by multiplication with $\mathbf{A}^{-1}$, gives that $\mathbf{A}^{-1}$ can be expressed as a degree $n-1$ polynomial of $\mathbf{A}$

$$\mathbf{A}^{-1} = d_{n-1}\mathbf{A}^{n-1} + d_{n-2}\mathbf{A}^{n-2} + \cdots + d_2\mathbf{A}^2 + d_1\mathbf{A} + d_0\mathbf{I}$$

If a system of linear equations $\mathbf{Ax} = \mathbf{b}$ has an invertible coefficient matrix, its solution can be expressed as $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$, which translates to

$$\mathbf{x} = d_{n-1}\mathbf{A}^{n-1}\mathbf{b} + d_{n-2}\mathbf{A}^{n-2}\mathbf{b} + \cdots + d_2\mathbf{A}^2\mathbf{b} + d_1\mathbf{Ab} + d_0\mathbf{b}$$

Krylov subspace

A Krylov subspace of order k is defined by

$$K_k = \text{Span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \mathbf{A}^2\mathbf{b}, \ldots, \mathbf{A}^{k-1}\mathbf{b}, \mathbf{A}^k\mathbf{b}\}$$

From the previous discussion it obviously holds that the solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$ is in K_n, even for the case when K_n does not span $\mathbb{R}^n$.

The minimization problem

If $\mathbf{A} \in \mathbb{R}^{n\times n}$ is symmetric and positive definite (SPD) and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$, the solution to $\mathbf{Ax} = \mathbf{b}$ coincides with the minimum of the function

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

Let $\mathbf{x}^*$ be the solution to $\mathbf{Ax} = \mathbf{b}$. Then

$$\begin{aligned} f(\mathbf{x}) &= \frac{1}{2}((\mathbf{x}-\mathbf{x}^*)+\mathbf{x}^*)^T\mathbf{A}((\mathbf{x}-\mathbf{x}^*)+\mathbf{x}^*) - ((\mathbf{x}-\mathbf{x}^*)+\mathbf{x}^*)^T\mathbf{b} \\ &= \frac{1}{2}\underbrace{(\mathbf{x}-\mathbf{x}^*)^T\mathbf{A}(\mathbf{x}-\mathbf{x}^*)}_{\geq 0} + f(\mathbf{x}^*) \geq f(\mathbf{x}^*) \end{aligned}$$

so the minimal value of $f(\mathbf{x})$ is $f(\mathbf{x}^*)$ which is realized for $\mathbf{x} = \mathbf{x}^*$.

Krylov sequence

We will conduct our search for the solution of $\mathbf{Ax} = \mathbf{b}$ by sequentially increasing the order of the Krylov subspace used. For each Krylov subspace, the minimum of $f(\mathbf{x})$ with the constraint $\mathbf{x} \in K_k$ is found, forming the **Krylov sequence** $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots$ i.e.

$$\mathbf{x}_k = \arg\min_{\mathbf{x} \in K_k} f(\mathbf{x})$$

Once $\mathbf{x}_n$ is reached, the solution is guaranteed to be found, though it may happen earlier in the sequence than at step n.

Minimum of $f(\mathbf{x})$ in K_k, first attempt

Let

$$\mathbf{M}_k = \begin{bmatrix}\mathbf{b} & \mathbf{A}\mathbf{b} & \mathbf{A}^2\mathbf{b} & \dots & \mathbf{A}^{k-1}\mathbf{b}\end{bmatrix}$$

then any vector $\mathbf{x} \in K_k$ can be expressed as $\mathbf{x} = \mathbf{M}_k\mathbf{c}$ where $\mathbf{c} \in \mathbb{R}^k$ is the coordinate of $\mathbf{x}$. The function $f(\mathbf{x})$ becomes

$$f(\mathbf{x}) = g(\mathbf{c}) = \frac{1}{2}\mathbf{c}^T\mathbf{M}_k^T\mathbf{A}\mathbf{M}_k\mathbf{c} - \mathbf{c}^T\mathbf{M}_k^T\mathbf{b}$$

the minimum of which is found by $\nabla g(\mathbf{c}) = \mathbf{0}$ giving

$$\mathbf{M}_k^T\mathbf{A}\mathbf{M}_k\mathbf{c} = \mathbf{M}_k^T\mathbf{b}$$

Warning

To find $\mathbf{c}$ and subsequently $\mathbf{x}_k$, we need to solve a system of linear equations. So to solve a system of linear equations we need solve a system of linear equations. This is bad!

An idea: What if $\mathbf{M}_k^T\mathbf{A}\mathbf{M}_k$ was something simple, like a diagonal matrix. Is it possible to choose another base for K_k to make this happen?

A special inner product

If $\mathbf{A}$ is symmetric positive definite (SPD), we can form an inner product

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^T \mathbf{A} \mathbf{y}$$

If $\mathbf{x}$ and $\mathbf{y}$ fulfil $\mathbf{x}^T \mathbf{A} \mathbf{y} = 0$, we say that $\mathbf{x}$ and $\mathbf{y}$ are $\mathbf{A}$-*orthogonal*. This inner product also gives us the $\mathbf{A}$-norm

$$\|\mathbf{x}\|_{\mathbf{A}} = \sqrt{\mathbf{x}^T \mathbf{A} \mathbf{x}}$$

An identity involving the $\mathbf{A}$-norm

$$f(\mathbf{x}) - f(\mathbf{x}^*) = \frac{1}{2} \|\mathbf{x} - \mathbf{x}^*\|_{\mathbf{A}}^2 = \frac{1}{2} \|\mathbf{r}\|_{\mathbf{A}^{-1}}^2$$

where $\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x}$.

Minimum of $f(\mathbf{x})$ in K_k, second attempt

Let

$$\mathbf{N}_k = \begin{bmatrix}\mathbf{d}_1 & \mathbf{d}_2 & \mathbf{d}_3 & \dots & \mathbf{d}_k\end{bmatrix}$$

where the columns form an $\mathbf{A}$-orthogonal basis for K_k, then any vector $\mathbf{x} \in K_k$ can be expressed as $\mathbf{x} = \mathbf{N}_k\mathbf{c}$ where $\mathbf{c} \in \mathbb{R}^k$ is the coordinate of $\mathbf{x}$. The function $f(\mathbf{x})$ becomes

$$f(\mathbf{x}) = g(\mathbf{c}) = \frac{1}{2}\mathbf{c}^T\mathbf{N}_k^T\mathbf{A}\mathbf{N}_k\mathbf{c} - \mathbf{c}^T\mathbf{N}_k^T\mathbf{b}$$

where $\mathbf{N}_k^T\mathbf{A}\mathbf{N}_k$ is a diagonal matrix with diagonal entries $\mathbf{d}_i^T\mathbf{A}\mathbf{d}_i$. The function $g(\mathbf{c})$ can thus be written as

$$g(\mathbf{c}) = \sum_{i=1}^{k}\frac{1}{2}c_i^2\mathbf{d}_i^T\mathbf{A}\mathbf{d}_i - c_i\mathbf{d}_i^T\mathbf{b}$$

The entries of $\nabla g(\mathbf{c})$ are

$$\frac{\partial}{\partial c_i}g(\mathbf{c}) = c_i\mathbf{d}_i^T\mathbf{A}\mathbf{d}_i - \mathbf{d}_i^T\mathbf{b}$$

Minimum of $f(\mathbf{x})$ in K_k, second attempt, cont

Thus the optimum, where $\nabla g(\mathbf{c}) = \mathbf{0}$, is found by

$$c_i = \frac{\mathbf{d}_i^T \mathbf{b}}{\mathbf{d}_i^T \mathbf{A} \mathbf{d}_i}$$

and

$$\mathbf{x}_k = \sum_{i=1}^{k} c_i \mathbf{d}_i = \mathbf{x}_{k-1} + c_k \mathbf{d}_k$$

The **A**-orthogonal base $\mathbf{d}_i$ is found using the Gram-Schmidt orthogonalization process.

Algorithm 1

```
$\mathbf{q}_1 := \mathbf{b}$
$\mathbf{d}_1 := \mathbf{b}$
$c_1 := \frac{\mathbf{d}_1^T \mathbf{b}}{\mathbf{d}_1^T \mathbf{A} \mathbf{d}_1}$
$\mathbf{x}_1 := c_1 \mathbf{d}_1$
for $k := 2$ to $n$
    $\mathbf{q}_k := \mathbf{A} \mathbf{q}_{k-1}$
    $\mathbf{d}_k := \mathbf{q}_k - \frac{\mathbf{q}_k^T \mathbf{A} \mathbf{d}_1}{\mathbf{d}_1^T \mathbf{A} \mathbf{d}_1} \mathbf{d}_1 - \frac{\mathbf{q}_k^T \mathbf{A} \mathbf{d}_2}{\mathbf{d}_2^T \mathbf{A} \mathbf{d}_2} \mathbf{d}_2 - \cdots - \frac{\mathbf{q}_k^T \mathbf{A} \mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^T \mathbf{A} \mathbf{d}_{k-1}} \mathbf{d}_{k-1}$
    $c_k := \frac{\mathbf{d}_k^T \mathbf{b}}{\mathbf{d}_k^T \mathbf{A} \mathbf{d}_k}$
    $\mathbf{x}_k := \mathbf{x}_{k-1} + c_k \mathbf{d}_k$
end for
```

It looks neat, but the Gram-Schmidt process is as time consuming as solving systems of linear equations, or worse. So this is also bad!

The gradient and the residual

From the function $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{x}^T\mathbf{b}$, we see that

$$\nabla f(\mathbf{x}) = \mathbf{A}\mathbf{x} - \mathbf{b} = -\mathbf{r}$$

or

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x} = -\nabla f(\mathbf{x})$$

It is also clear that since $\mathbf{x}_k \in K_k$ it holds that

$$\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k \in K_{k+1}$$

so one possible use for the residual (i.e the negative gradient) is for expanding the Krylov subspace for the next iteration. Also note that

$$\begin{aligned}\mathbf{r}_k &= \mathbf{b} - \mathbf{A}(\mathbf{x}_{k-1} + c_k\mathbf{d}_k) = \mathbf{b} - \mathbf{A}\mathbf{x}_{k-1} - c_k\mathbf{A}\mathbf{d}_k \\ &= \mathbf{r}_{k-1} - c_k\mathbf{A}\mathbf{d}_k\end{aligned}$$

We use these observations to formulate a slightly modified algorithm

Algorithm 2

```
$\mathbf{d}_1 := \mathbf{b}$
$c_1 := \frac{\mathbf{d}_1^T\mathbf{b}}{\mathbf{d}_1^T\mathbf{A}\mathbf{d}_1}$
$\mathbf{x}_1 := c_1\mathbf{d}_1$
$\mathbf{r}_1 := \mathbf{b} - c_1\mathbf{A}\mathbf{d}_1$
for $k := 2$ to $n$
    $\mathbf{d}_k := \mathbf{r}_{k-1} - \frac{\mathbf{r}_{k-1}^T\mathbf{A}\mathbf{d}_1}{\mathbf{d}_1^T\mathbf{A}\mathbf{d}_1}\mathbf{d}_1 - \frac{\mathbf{r}_{k-1}^T\mathbf{A}\mathbf{d}_2}{\mathbf{d}_2^T\mathbf{A}\mathbf{d}_2}\mathbf{d}_2 - \cdots - \frac{\mathbf{r}_{k-1}^T\mathbf{A}\mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^T\mathbf{A}\mathbf{d}_{k-1}}\mathbf{d}_{k-1}$
    $c_k := \frac{\mathbf{d}_k^T\mathbf{b}}{\mathbf{d}_k^T\mathbf{A}\mathbf{d}_k}$
    $\mathbf{x}_k := \mathbf{x}_{k-1} + c_k\mathbf{d}_k$
    $\mathbf{r}_k := \mathbf{r}_{k-1} - c_k\mathbf{A}\mathbf{d}_k$
end for
```

Numerical experiments reveal that most of the terms in the calculation of $\mathbf{d}_k$ are zero with the exception of the first and last term. If this can be established in general, the calculation of $\mathbf{d}_k$ can be radically simplified, and we can formulate something that is *not bad*.

Three useful properties

1. $\mathbf{d}_j^T \mathbf{r}_k = 0$ for all $j \leq k$
2. $\mathbf{r}_j^T \mathbf{r}_k = 0$ for all $j < k$
3. $\mathbf{r}_k^T \mathbf{A} \mathbf{d}_j = 0$ for all $j < k$

From 3. it is clear that only the first and last term in the Gram-Schmidt orthoganalization is non-zero.

Proofs of the properties

1. Proof by induction! Suppose that 1. holds for the case $\mathbf{d}_j^T \mathbf{r}_{k-1} = 0$ for all $j \leq k-1$. Then for $j \leq k-1$ it holds that
$$\mathbf{d}_j^T \mathbf{r}_k = \mathbf{d}_j^T(\mathbf{r}_{k-1} - c_k \mathbf{A}\mathbf{d}_k) = \mathbf{d}_j^T \mathbf{r}_{k-1} - \mathbf{d}_j^T \mathbf{A}\mathbf{d}_k = 0 - 0 = 0$$
and for $j = k$ we get
$$\begin{aligned}\mathbf{d}_k^T \mathbf{r}_k &= \mathbf{d}_k^T(\mathbf{b} - \mathbf{A}\mathbf{x}_k) = \mathbf{d}_k^T \mathbf{b} - \mathbf{d}_k^T \mathbf{A}(\mathbf{x}_{k-1} + c_k \mathbf{d}_k) \\ &= \mathbf{d}_k^T \mathbf{b} - \mathbf{d}_k^T \mathbf{A}\mathbf{x}_{k-1} - \frac{\mathbf{d}_k^T \mathbf{b}}{\mathbf{d}_k^T \mathbf{A}\mathbf{d}_k} \mathbf{d}_k^T \mathbf{A}\mathbf{d}_k = 0\end{aligned}$$
The term $\mathbf{d}_k^T \mathbf{A}\mathbf{x}_{k-1}$ is zero since $\mathbf{x}_{k-1}$ is a linear combination of $\mathbf{d}_1, \ldots, \mathbf{d}_{k-1}$ and thus $\mathbf{A}$-orthogonal to $\mathbf{d}_k$.
2. From the Gram-Schmidt step it is clear that there are constants a_i such that $\mathbf{r}_j = \sum_{i=1}^{j+1} a_i \mathbf{d}_i$, and since $j+1 \leq k$ it follows from 1.
$$\mathbf{r}_k^T \mathbf{r}_j = \sum_{i=1}^{j+1} a_i \mathbf{r}_k^T \mathbf{d}_i = 0$$
3. If the solution is found, $\mathbf{r}_k = \mathbf{0}$. Otherwise $c_j \neq 0$ and from 2. we get $\mathbf{r}_k^T \mathbf{A}\mathbf{d}_j = \mathbf{r}_k^T \frac{1}{c_j}(\mathbf{r}_{j-1} - \mathbf{r}_j) = \frac{1}{c_j}(\mathbf{r}_k^T \mathbf{r}_{j-1} - \mathbf{r}_k^T \mathbf{r}_j) = 0$

Algorithm 3

```
$\mathbf{d}_1 := \mathbf{b}$
$c_1 := \frac{\mathbf{d}_1^T \mathbf{b}}{\mathbf{d}_1^T \mathbf{A} \mathbf{d}_1}$
$\mathbf{x}_1 := c_1 \mathbf{d}_1$
$\mathbf{r}_1 := \mathbf{b} - c_1 \mathbf{A} \mathbf{d}_1$
for $k := 2$ to $n$
    $\mathbf{d}_k := \mathbf{r}_{k-1} - \frac{\mathbf{r}_{k-1}^T \mathbf{A} \mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^T \mathbf{A} \mathbf{d}_{k-1}} \mathbf{d}_{k-1}$
    $c_k := \frac{\mathbf{d}_k^T \mathbf{b}}{\mathbf{d}_k^T \mathbf{A} \mathbf{d}_k}$
    $\mathbf{x}_k := \mathbf{x}_{k-1} + c_k \mathbf{d}_k$
    $\mathbf{r}_k := \mathbf{r}_{k-1} - c_k \mathbf{A} \mathbf{d}_k$
end for
```

Algorithm 3 - Discussion

- This is starting to look really good, but there still is a major drawback in this algorithm. The computation intesive task here is to make matrix times vector multiplication, which is made four times in each iteration. Moreover, it is the same matrix vector multiplication that is repeated over and over again. Why not do it once in each iteration and place the result in a vector, $\mathbf{z}_k := \mathbf{A}\mathbf{d}_k$?
- Even though the exact solution is found after (at most) n steps in exact arithmetics, the method is not used in that way in practice. I good approximation is in general found in much fewer iterations. Also it is sensitive to rounding errors if too many iterations are used.
- The sensitivity to rounding errors go down if some of the quantities are calculated in a different way.

$$\mathbf{d}_k^T\mathbf{b} = \mathbf{d}_k^T(\mathbf{b} - \mathbf{A}\mathbf{x}_{k-1}) = (\mathbf{r}_{k-1} + k\mathbf{d}_{k-1})^T\mathbf{r}_{k-1} = \mathbf{r}_{k-1}^T\mathbf{r}_{k-1}$$

$$\frac{\mathbf{r}_{k-1}^T\mathbf{A}\mathbf{d}_{k-1}}{\mathbf{d}_{k-1}^T\mathbf{A}\mathbf{d}_{k-1}} = \frac{1}{c_{k-1}}\frac{\mathbf{r}_{k-1}^T(\mathbf{r}_{k-2} - \mathbf{r}_{k-1})}{\mathbf{d}_{k-1}^T\mathbf{A}\mathbf{d}_{k-1}} = \frac{\mathbf{d}_{k-1}^T\mathbf{A}\mathbf{d}_{k-1}}{\mathbf{r}_{k-2}^T\mathbf{r}_{k-2}}\frac{-\mathbf{r}_{k-1}^T\mathbf{r}_{k-1}}{\mathbf{d}_{k-1}^T\mathbf{A}\mathbf{d}_{k-1}}$$
$$= -\frac{\mathbf{r}_{k-1}^T\mathbf{r}_{k-1}}{\mathbf{r}_{k-2}^T\mathbf{r}_{k-2}}$$

Algorithm 4 - the Conjugate Gradient algorithm, Hestenes & Stiefel (1952)

```
$\mathbf{d}_1 := \mathbf{b}$
$\mathbf{r}_0 := \mathbf{b}$
$\mathbf{z}_1 := \mathbf{A}\mathbf{d}_1$
$c_1 := \frac{\mathbf{r}_0^T \mathbf{r}_0}{\mathbf{d}_1^T \mathbf{z}_1}$
$\mathbf{x}_1 := c_1 \mathbf{d}_1$
$\mathbf{r}_1 := \mathbf{r}_0 - c_1 \mathbf{z}_1$
$k := 2$
while $\|r\|_2 > \varepsilon \|b\|_2$ do
    $\mathbf{d}_k := \mathbf{r}_{k-1} + \frac{\mathbf{r}_{k-1}^T \mathbf{r}_{k-1}}{\mathbf{r}_{k-2}^T \mathbf{r}_{k-2}} \mathbf{d}_{k-1}$
    $\mathbf{z}_k := \mathbf{A}\mathbf{d}_k$
    $c_k := \frac{\mathbf{r}_{k-1}^T \mathbf{r}_{k-1}}{\mathbf{d}_k^T \mathbf{z}_k}$
    $\mathbf{x}_k := \mathbf{x}_{k-1} + c_k \mathbf{d}_k$
    $\mathbf{r}_k := \mathbf{r}_{k-1} - c_k \mathbf{z}_k$
    $k := k + 1$
end while
```

Conjugate Gradient with warm start

If there is an initial guess $\mathbf{x}_0$ for the solution, we can find the true solution by calculating the delta $\Delta\mathbf{x}$ from the system $\mathbf{A}\Delta\mathbf{x} = \mathbf{r}_0$, where $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$, and forming $\mathbf{x}^* = \mathbf{x}_0 + \Delta\mathbf{x}$. This translates easily to the algorithm.

Algorithm 5 - Conjugate Gradient with warm start

```
$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{d}_1 := \mathbf{r}_0$
$\mathbf{z}_1 := \mathbf{A}\mathbf{d}_1$
$c_1 := \frac{\mathbf{r}_0^T\mathbf{r}_0}{\mathbf{d}_1^T\mathbf{z}_1}$
$\mathbf{x}_1 := \mathbf{x}_0 + c_1\mathbf{d}_1$
$\mathbf{r}_1 := \mathbf{r}_0 - c_1\mathbf{z}_1$
$k := 2$
while $\|r\|_2 > \varepsilon\|b\|_2$ do
    $\mathbf{d}_k := \mathbf{r}_{k-1} + \frac{\mathbf{r}_{k-1}^T\mathbf{r}_{k-1}}{\mathbf{r}_{k-2}^T\mathbf{r}_{k-2}}\mathbf{d}_{k-1}$
    $\mathbf{z}_k := \mathbf{A}\mathbf{d}_k$
    $c_k := \frac{\mathbf{r}_{k-1}^T\mathbf{r}_{k-1}}{\mathbf{d}_k^T\mathbf{z}_k}$
    $\mathbf{x}_k := \mathbf{x}_{k-1} + c_k\mathbf{d}_k$
    $\mathbf{r}_k := \mathbf{r}_{k-1} - c_k\mathbf{z}_k$
    $k := k + 1$
end while
```