15 Conjugate Gradients

This method for symmetric positive definite matrices is considered to be the "original" Krylov subspace method. It was proposed by Hestenes and Stiefel in 1952, and is motivated by the following theorem.

Theorem 15.1 *If A is symmetric positive definite, then solving $A\boldsymbol{x} = \boldsymbol{b}$ is equivalent to minimizing the quadratic form*

$$\varphi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b}.$$

Proof We will consider changes of φ along a certain ray $\boldsymbol{x} + \alpha\boldsymbol{p}$ with $\alpha \in \mathbb{R}$, and fixed direction vector $\boldsymbol{p} \neq \boldsymbol{0}$.

First we show that $\varphi(\boldsymbol{x} + \alpha\boldsymbol{p}) > \varphi(\boldsymbol{x})$ if A symmetric positive definite.

$$\begin{aligned}
\varphi(\boldsymbol{x} + \alpha\boldsymbol{p}) &= \frac{1}{2}(\boldsymbol{x} + \alpha\boldsymbol{p})^T A(\boldsymbol{x} + \alpha\boldsymbol{p}) - (\boldsymbol{x} + \alpha\boldsymbol{p})^T\boldsymbol{b} \\
&= \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T A(\alpha\boldsymbol{p}) + \frac{1}{2}(\alpha\boldsymbol{p})^T A\boldsymbol{x} + \frac{1}{2}(\alpha\boldsymbol{p})^T A(\alpha\boldsymbol{p}) - \boldsymbol{x}^T\boldsymbol{b} - \alpha\boldsymbol{p}^T\boldsymbol{b} \\
&\overset{A^T=A}{=} \underbrace{\frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b}}_{=\varphi(\boldsymbol{x})} + \alpha\boldsymbol{p}^T A\boldsymbol{x} - \alpha\boldsymbol{p}^T\boldsymbol{b} + \frac{1}{2}\alpha^2\boldsymbol{p}^T A\boldsymbol{p} \\
&= \varphi(\boldsymbol{x}) + \alpha\boldsymbol{p}^T(A\boldsymbol{x} - \boldsymbol{b}) + \frac{1}{2}\alpha^2\underbrace{\boldsymbol{p}^T A\boldsymbol{p}}_{>0}.
\end{aligned}$$

Thus, we see that φ (as a quadratic function in α with positive leading coefficient) will have to have a minimum along the ray $\boldsymbol{x} + \alpha\boldsymbol{p}$.

We now decide what the value of α at this minimum is. A necessary condition (and also sufficient since the coefficient of α^2 is positive) is

$$\frac{d}{d\alpha}\varphi(\boldsymbol{x} + \alpha\boldsymbol{p}) = 0.$$

To this end we compute

$$\frac{d}{d\alpha}\varphi(\boldsymbol{x} + \alpha\boldsymbol{p}) = \boldsymbol{p}^T(A\boldsymbol{x} - \boldsymbol{b}) + \alpha\boldsymbol{p}^T A\boldsymbol{p},$$

which has its root at

$$\hat{\alpha} = \frac{\boldsymbol{p}^T(\boldsymbol{b} - A\boldsymbol{x})}{\boldsymbol{p}^T A\boldsymbol{p}}.$$

The corresponding minimum value is

$$\varphi(\boldsymbol{x} + \hat{\alpha}\boldsymbol{p}) = \varphi(\boldsymbol{x}) - \underbrace{\frac{\left[\boldsymbol{p}^T(\boldsymbol{b} - A\boldsymbol{x})\right]^2}{2\boldsymbol{p}^T A\boldsymbol{p}}}_{\geq 0}.$$

The last equation shows that

$$\varphi(\boldsymbol{x} + \hat{\alpha}\boldsymbol{p}) < \varphi(\boldsymbol{x}) \quad \text{if and only if} \quad \boldsymbol{p}^T(\boldsymbol{b} - A\boldsymbol{x}) \neq 0,$$

i.e., $\boldsymbol{p}$ is not orthogonal to the residual $\boldsymbol{r} = \boldsymbol{b} - A\boldsymbol{x}$.

To see the equivalence with the solution of the linear system $A\boldsymbol{x} = \boldsymbol{b}$ we need to consider two possibilities:

1. $\boldsymbol{x}$ is such that $A\boldsymbol{x} = \boldsymbol{b}$. Then $\varphi(\boldsymbol{x} + \hat{\alpha}\boldsymbol{p}) = \varphi(\boldsymbol{x})$ and $\varphi(\boldsymbol{x})$ is the minimum value.
2. $\boldsymbol{x}$ is such that $A\boldsymbol{x} \neq \boldsymbol{b}$. Then $\varphi(\boldsymbol{x} + \hat{\alpha}\boldsymbol{p}) < \varphi(\boldsymbol{x})$, i.e., there exists a direction $\boldsymbol{p}$ such that $\boldsymbol{p}^T(\boldsymbol{b} - A\boldsymbol{v}) \neq 0$ and $\varphi(\boldsymbol{x})$ is not the minimum.

■

The preceding proof actually suggests a rough iterative algorithm:

Take $\boldsymbol{x}_0 = \boldsymbol{0}$, $\boldsymbol{r}_0 = \boldsymbol{b}$, $\boldsymbol{p}_0 = \boldsymbol{r}_0$

for $n = 1, 2, 3, \ldots$

Compute a step length

$$\alpha_n = \left(\boldsymbol{p}_{n-1}^T \boldsymbol{r}_{n-1}\right) / \left(\boldsymbol{p}_{n-1}^T A \boldsymbol{p}_{n-1}\right)$$

Update the approximate solution

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + \alpha_n \boldsymbol{p}_{n-1}$$

Update the residual

$$\boldsymbol{r}_n = \boldsymbol{r}_{n-1} - \alpha_n A \boldsymbol{p}_{n-1}$$

Find a new search direction $\boldsymbol{p}_n$

end

Note that at this point we have not specified how to pick the search directions $\boldsymbol{p}_n$. This will be the crucial ingredient in the algorithm.

The formula above for the residual update follows from

$$\begin{aligned} \boldsymbol{r}_n &= \boldsymbol{b} - A\boldsymbol{x}_n = \boldsymbol{b} - A(\boldsymbol{x}_{n-1} + \alpha_n \boldsymbol{p}_{n-1}) \\ &= \boldsymbol{b} - A\boldsymbol{x}_{n-1} - \alpha_n A\boldsymbol{p}_{n-1} \\ &= \boldsymbol{r}_{n-1} - \alpha_n A\boldsymbol{p}_{n-1}. \end{aligned}$$

15.1 The Steepest Descent Algorithm

An obvious choice for the selection of the search direction is

$$\boldsymbol{p}_n = -\nabla\varphi(\boldsymbol{x}_n)$$

since we know from calculus that the direction of largest decrease of φ is in the direction opposite its gradient. Moreover, since $\varphi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b}$ we have

$$\nabla\varphi(\boldsymbol{x}) = A\boldsymbol{x}_n - \boldsymbol{b}.$$

This leads to

Algorithm (Steepest Descent)

Take $\boldsymbol{x}_0 = \boldsymbol{0}$, $\boldsymbol{r}_0 = \boldsymbol{b}$, $\boldsymbol{p}_0 = \boldsymbol{r}_0$

for $n = 1, 2, 3, \ldots$

Compute a step length

$$\alpha_n = \left(\boldsymbol{p}_{n-1}^T \boldsymbol{r}_{n-1}\right) / \left(\boldsymbol{p}_{n-1}^T A \boldsymbol{p}_{n-1}\right)$$

Update the approximate solution

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + \alpha_n \boldsymbol{p}_{n-1}$$

Update the residual

$$\boldsymbol{r}_n = \boldsymbol{r}_{n-1} - \alpha_n A \boldsymbol{p}_{n-1}$$

Set the new search direction

$$\boldsymbol{p}_n = \boldsymbol{r}_n$$

end

Note that for this choice of search direction the step length α can also be written as

$$\alpha_n = \left(\boldsymbol{r}_{n-1}^T \boldsymbol{r}_{n-1}\right) / \left(\boldsymbol{p}_{n-1}^T A \boldsymbol{p}_{n-1}\right).$$

15.2 The Conjugate Gradient Algorithm

It turns out that the "obvious" search directions are not ideal (since they are applied in an iterative fashion). Convergence of the steepest descent algorithm is usually rather slow. It is better to employ so-called *conjugate search directions*. The idea is to somehow remove from the gradient at each step those components parallel to previously used search directions. The resulting algorithm is

Algorithm (Conjugate Gradient)

Take $\boldsymbol{x}_0 = \boldsymbol{0}$, $\boldsymbol{r}_0 = \boldsymbol{b}$, $\boldsymbol{p}_0 = \boldsymbol{r}_0$

for $n = 1, 2, 3, \ldots$

Compute a step length

$$\alpha_n = \left(\boldsymbol{r}_{n-1}^T \boldsymbol{r}_{n-1}\right) / \left(\boldsymbol{p}_{n-1}^T A \boldsymbol{p}_{n-1}\right)$$

Update the approximate solution

$$\boldsymbol{x}_n = \boldsymbol{x}_{n-1} + \alpha_n \boldsymbol{p}_{n-1}$$

Update the residual

$$\boldsymbol{r}_n = \boldsymbol{r}_{n-1} - \alpha_n A \boldsymbol{p}_{n-1}$$

Compute a gradient correction factor

$$\beta_n = \left(\boldsymbol{r}_n^T \boldsymbol{r}_n\right) / \left(\boldsymbol{r}_{n-1}^T \boldsymbol{r}_{n-1}\right)$$

Set the new search direction

$$\boldsymbol{p}_n = \boldsymbol{r}_n + \beta_n \boldsymbol{p}_{n-1}$$

end

For both the steepest descent and the conjugate gradient algorithm the main computational cost is hidden in the one matrix-vector multiplication that is required per iteration. As with the Arnoldi and GMRES methods, this operation is treated as a "black box" and can be accomplished in $\mathcal{O}(m)$ to $\mathcal{O}(m^2)$ operations depending on the structure of A. In many practical cases the entire (preconditioned) CG algorithm will require only $\mathcal{O}(m)$ operations. This is very fast.

As mentioned at the beginning of this section, one can also establish a connection to Krylov subspace methods.

Theorem 15.2 *Let A be symmetric positive definite. As long as the conjugate gradient method has not yet converged (i.e., as long as $\boldsymbol{r}_{n-1} \neq \boldsymbol{0}$) we have*

$$\begin{aligned} \operatorname{span}\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\} &= \operatorname{span}\{\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{n-1}\} \\ &= \operatorname{span}\{\boldsymbol{r}_0, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_{n-1}\} \\ &= \operatorname{span}\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{n-1}\boldsymbol{b}\} = \mathcal{K}_n. \end{aligned}$$

Moreover, the residuals are orthogonal in the usual sense, i.e,

$$\boldsymbol{r}_n^T \boldsymbol{r}_j = 0, \qquad j < n,$$

and the search directions ar A-orthogonal (or A-conjugate), i.e.,

$$\boldsymbol{p}_n^T A \boldsymbol{p}_j = 0, \qquad j < n.$$

Proof An inductive proof of this theorem can be found in the book [Trefethen/Bau]. ■

15.3 Convergence of the CG Algorithm

Recall that the GMRES algorithm minimizes the 2-norm of the residual, $\|\boldsymbol{r}_n\|_2 \to \min$. We will now show that the CG algorithm satisfies a different optimality criterion. It minimizes the A-norm of the error, i.e., if $\boldsymbol{e}_n = \boldsymbol{x}^* - \boldsymbol{x}_n$, is the error between the exact solution $\boldsymbol{x}^* = A^{-1}\boldsymbol{b}$ and the n-th approximation $\boldsymbol{x}_n$, then CG minimizes

$$\|\boldsymbol{e}_n\|_A = \sqrt{\boldsymbol{e}_n^T A \boldsymbol{e}_n}.$$

Theorem 15.3 *Let A be symmetric positive definite. If the conjugate gradient algorithm has not yet converged (i.e., $\boldsymbol{r}_{n-1} \neq \boldsymbol{0}$) then $\boldsymbol{x}_n$ is the unique vector in $\mathcal{K}_n$ such that $\|\boldsymbol{e}_n\|_A$ is minimized.*

Moreover, $\|\boldsymbol{e}_n\|_A \leq \|\boldsymbol{e}_{n-1}\|_A$ and (if we are using exact arithmetic) $\boldsymbol{e}_n = \boldsymbol{0}$ for some $n \leq m$.

Proof We will prove the first part only. From the previous theorem we know that the approximate solution $\boldsymbol{x}_n$ lies in the Krylov subspace $\mathcal{K}_n$. In order to show that $\boldsymbol{x}_n$ is the unique minimizer of $\|\boldsymbol{e}\|_A$ we consider an arbitrary vector

$$\boldsymbol{x} = \boldsymbol{x}_n - \Delta\boldsymbol{x} \quad \in \mathcal{K}_n$$

and show that in order to minimize $\|\boldsymbol{e}\|_A$ we necessarily have $\Delta\boldsymbol{x} = \boldsymbol{0}$.

If $\boldsymbol{x}^*$ is the exact solution of $A\boldsymbol{x} = \boldsymbol{b}$, then

$$\boldsymbol{e} = \boldsymbol{x}^* - \boldsymbol{x} = \boldsymbol{x}^* - \boldsymbol{x}_n + \Delta\boldsymbol{x} = \boldsymbol{e}_n + \Delta\boldsymbol{x}.$$

Therefore,

$$\begin{aligned} \|\boldsymbol{e}\|_A^2 &= \|\boldsymbol{e}_n + \Delta\boldsymbol{x}\|_A^2 \\ &= (\boldsymbol{e}_n + \Delta\boldsymbol{x})^T A\,(\boldsymbol{e}_n + \Delta\boldsymbol{x}) \\ &= \boldsymbol{e}_n^T A\boldsymbol{e}_n + 2\boldsymbol{e}_n^T A\,(\Delta\boldsymbol{x}) + (\Delta\boldsymbol{x})^T A\,(\Delta\boldsymbol{x}) \end{aligned}$$

since A is symmetric.

Next we realize that

$$\boldsymbol{e}_n^T A = (\boldsymbol{x}^* - \boldsymbol{x}_n)^T A = \left(A^{-1}\boldsymbol{b} - \boldsymbol{x}_n\right) A = \boldsymbol{b}^T - \boldsymbol{x}_n^T A = \boldsymbol{r}_n^T,$$

and observe that

$$\boldsymbol{r}_n^T \Delta\boldsymbol{x} = 0$$

since $\Delta\boldsymbol{x} \in \mathcal{K}_n = \operatorname{span}\{\boldsymbol{r}_0, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_{n-1}\}$ and $\boldsymbol{r}_n^T \boldsymbol{r}_j = 0$ for $j < n$ by the previous theorem.

This leaves us with

$$\begin{aligned} \|\boldsymbol{e}\|_A^2 &= \boldsymbol{e}_n^T A\boldsymbol{e}_n + 2\underbrace{\underbrace{\boldsymbol{e}_n^T A}_{=\boldsymbol{r}_n^T}(\Delta\boldsymbol{x})}_{=0} + (\Delta\boldsymbol{x})^T A\,(\Delta\boldsymbol{x}) \\ &= \|\boldsymbol{e}_n\|_A^2 + (\Delta\boldsymbol{x})^T A\,(\Delta\boldsymbol{x})\,. \end{aligned}$$

Note that the quadratic form $(\Delta\boldsymbol{x})^T A\,(\Delta\boldsymbol{x})$ is certainly non-negative since A is positive definite. Moreover, it is zero only if $\Delta\boldsymbol{x} = \boldsymbol{0}$.

Thus, the A-norm of the error is minimized if $\Delta\boldsymbol{x} = \boldsymbol{0}$, i.e., for the CG approximate solution $\boldsymbol{x}_n$. ■

We can come to the same conclusion with the following argument:

$$\begin{aligned} \|\boldsymbol{e}_n\|_A^2 &= \boldsymbol{e}_n^T A\boldsymbol{e}_n = (\boldsymbol{x}^* - \boldsymbol{x}_n)^T A\,(\boldsymbol{x}^* - \boldsymbol{x}_n) \\ &= (\boldsymbol{x}^*)^T \underbrace{A\boldsymbol{x}^*}_{=\boldsymbol{b}} - 2\boldsymbol{x}_n^T \underbrace{A\boldsymbol{x}^*}_{=\boldsymbol{b}} + \boldsymbol{x}_n^T A\boldsymbol{x}_n \\ &= (\boldsymbol{x}^*)^T\boldsymbol{b} + \boldsymbol{x}_n^T A\boldsymbol{x}_n - 2\boldsymbol{x}_n^T\boldsymbol{b} \\ &= (\boldsymbol{x}^*)^T\boldsymbol{b} + 2\varphi(\boldsymbol{x}_n). \end{aligned}$$

Here $\varphi(\boldsymbol{x}_n)$ is the same quadratic form used earlier. Since $(\boldsymbol{x}^*)^T\boldsymbol{b}$ is a constant we see that minimizing the A-norm of the error is equivalent to minimizing the quadratic form $\varphi(\boldsymbol{x}_n)$.

For the *convergence rate* of the CG algorithm one can show that

$$\|\boldsymbol{e}_n\|_A \leq \|\boldsymbol{e}_0\|_A \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^n,$$

where $\kappa = \kappa_2(A)$ the 2-norm condition number of A. Since

$$\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1} = 1 - \frac{2}{\sqrt{\kappa}+1}$$

we see that convergence will be very slow if κ is large. This shows that preconditioning efforts for the CG algorithm are aimed at reducing the condition number of A.

For a moderate size κ it turns out that one can expect convergence of the CG algorithm in $\mathcal{O}(\sqrt{\kappa})$ iterations. In fact, in practice the CG algorithm often converges faster than predicted by this upper bound.

Remark It is possible to interpret the conjugate gradient method as an analogue of Lanczos iteration for linear systems. Since we claimed earlier that Lanczos iteration is a special case of Arnoldi iteration for symmetric matrices, it turns out that the $(n+1)$-term recursion we derived earlier for Arnoldi iteration turn into a 3-term recursion. One can indeed show that this 3-term recursion is hidden inside the CG algorithm.

Convergence of the CG algorithm is illustrated in the MATLAB code `CGDemo.m`. The symmetric test matrix is constructed as follows. Initially it contains ones on the main diagonal and random numbers uniformly distributed in $[-1, 1]$ in the off-diagonal positions. Then any off-diagonal entry with $|a_{ij}| > \tau$ is set to zero, where τ is a parameter. For small values of τ the matrix is positive definite and very sparse, and the CG algorithm converges rapidly. For larger values, such as $\tau = 0.2$ the matrix is no longer positive definite, and the CG algorithm does not converge. We also note that for these test matrices, preconditioning does not improve convergence of the CG algorithm.