1 Note on Conjugate Gradient's derivation

We built the CG algorithm as a modification of the steepest descent algorithm to minimize the quadratic $\phi_{\mathbf{A}}(x) = \frac{1}{2}x^T\mathbf{A}x - b^Tx$. We reached the following conclusions:

$$x_{k+1} = x_k + \alpha_k p_k \tag{1.1}$$

$$\alpha_k = \frac{p_k^T r_0}{p_k^T \mathbf{A} p_k} \tag{1.2}$$

$$r_{k+1} = r_k - \alpha_k \mathbf{A} p_k = r_0 - \sum_{j=0}^{k} \alpha_j \mathbf{A} p_j \tag{1.3}$$

In class, we argued that the conjugate directions obtained via the Gram-Schmidt process applied to the new residual r_{k+1}:

$$p_{k+1} = r_{k+1} - \sum_{j=0}^{k} \frac{r_{k+1}^T \mathbf{A} p_j}{p_j^T \mathbf{A} p_j} p_j$$

Can really be obtained as an update of p_k. That is, we must show that the numerator in our coefficients $r_{k+1}^T \mathbf{A} p_j$ is 0 for $j = 0, \ldots, k-1$.

In order to show this, it is useful to prove a few auxiliary results about the CG iteration, and the properties of its residuals and conjugate directions:

Theorem 1.1 *Consider the first k iterations of the CG algorithm, and assume it has not converged to a solution yet, meaning $r_k \neq 0$. Then,*

$$\begin{aligned}\operatorname{span}(\{r_0, \ldots, r_k\}) &= \operatorname{span}(\{p_0, \ldots, p_k\}) \\ \operatorname{span}(\{r_0, \ldots, r_k\}) &= \operatorname{span}(\{r_0, \mathbf{A}r_0, \ldots, \mathbf{A}^k r_0\})\end{aligned}$$

We denote $\mathcal{K}(r0, \mathbf{A}, k) = \operatorname{span}(r_0, \mathbf{A}r_0, \ldots, \mathbf{A}^k r_0)$ the Krylov subspace of degree k for r_0 and $\mathbf{A}$.

The proof of the first statement can be obtained from inspection of the CG algorithm, and our design of the conjugate directions as an A-orthogonal basis of the span of the residuals. To show the second statement, we note that $p_0 = r_0$, and assuming the statement is true for a value of k, equation 1.3 tells us that r_{k+1} is a linear combination of r_k and Ap_k. By assumption:

$$\begin{aligned}r_k &\in \operatorname{span}(\{r_0, \mathbf{A}r_0, \ldots, \mathbf{A}^k r_0\}) \\ Ap_k &\in \operatorname{span}(\{\mathbf{A}r_0, \mathbf{A}^2 r_0, \ldots, \mathbf{A}^{k+1} r_0\})\end{aligned}$$

Which implies $r_{k+1} \in \mathcal{K}(r0, \mathbf{A}, k+1)$. This shows the span of the residuals is contained inside of this Krylov subspace of degree $k+1$. To show that these spans are equal, we can observe that $\mathbf{A}^{k+1} r_0 = \mathbf{A}(\mathbf{A}^k r_0)$, and the vector inside the parenthesis is in the span of the conjugate directions p_j for $j = 0, \ldots, k$.

Next, we can show that the k-th residual is in fact *orthogonal* (in the euclidean sense) to the span of the previous conjugate directions, that is,

Theorem 1.2 *Consider the first k iterations of the CG algorithm, and assume it has not converged to a solution yet, meaning $r_k \neq 0$. Then,*

$$r_k^T p_j = 0 \quad \forall j = 0, \dots, k-1 \tag{1.4}$$

To show this, we again use the definition of the residual in equation 1.3 and the fact that the directions are conjugate, to show:

$$\begin{aligned} p_i^T r_k &= p_i^T \left(r_0 - \sum_{j=0}^{k-1} \alpha_j A p_j \right) \\ &= p_i^T r_0 - \alpha_i p_i^T A p_i = p_i^T r_0 - \frac{p_i^T r_0}{p_i^T \mathbf{A} p_i} p_i^T A p_i = 0 \end{aligned}$$

Now, we can combine both theorems to show the desired result. Consider $r_{k+1}^T \mathbf{A} p_j$, for $j = 0, \dots, k-1$. By Theorem 1.1, Ap_j is in the span of $\{Ar_0, \dots, A^{j+1} r_0\}$, which in turn is contained in the span of $\{p_0, \dots, p_{j+1}\}$. By theorem 1.2, since $j+1 \leq k$, then it follows that $r_{k+1}^T \mathbf{A} p_j = 0$. And so,

$$p_{k+1} = r_{k+1} - \frac{r_{k+1}^T \mathbf{A} p_k}{p_k^T \mathbf{A} p_k} p_k = r_{k+1} + \beta_k p_k$$

Pseudocode of the CG algorithm

I mentioned that the pseudocode we'd obtain from our derivation looks a little different than the one you can find in textbooks or online, but that this is simply due to small optimizations of how we compute α_k and β_k. The standard pseudocode reads:

```
Given $x_0$
$r_0 = b - Ax_0; \quad p_0 = r_0; \quad k = 0;$
while $||r_k|| > \varepsilon$
    $q_k = \mathbf{A} p_k$
    $\alpha_k = \frac{r_k^T r_k}{p_k^T q_k}$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k q_k$
    $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
    $p_{k+1} = r_{k+1} + \beta_k p_k$
    $k = k + 1;$
end
```

This way of computing the CG iterates is strongly preferred because it reduces the work to *one matrix-vector product* $\mathbf{A} p_k$, and then some $O(n)$ work. The cost of the CG algorithm can then be estimated to be $O($ (# iterations) * (cost of **A** matrix vector product)). If we can make the number of iterations small and the matrix vector product of $\mathbf{A}$ fast, we have a *very* fast linear system solver.

Let's examine the formula for the numerator of α_k:

$$\begin{aligned} r_k^T r_k &= r_k^T (p_k - \beta_{k-1} p_{k-1}) \\ &= r_k^T p_k - \beta_{k-1} r_k^T p_{k-1} = r_k^T p_k \end{aligned}$$

Because r_k is orthogonal to the span of p_j from $j = 0, \ldots, k-1$ (theorem 1.2). Using conjugacy of the p directions, one can also show $r_k^T p_k = r_0^T p_k$.

Now, let's examine the formula for β_k. We can find this by taking the difference between the residuals:

$$\begin{aligned} r_{k+1} - r_k &= \alpha_k A p_k \\ r_{k+1}^T r_{k+1} - r_{k+1}^T r_k &= \frac{r_k^T r_k}{p_k^T A p_k} r_{k+1}^T A p_k \\ \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} &= \frac{r_{k+1}^T A p_k}{p_k^T A p_k} = \beta_k \end{aligned}$$

Using the fact that $r_{k+1}^T r_k = 0$.

Convergence results

The three main theorems concerning convergence of the CG algorithm (Nocedal & Wright, Chapter 5) tell us in how many steps we can expect CG to converge, and a bound on the rate of *linear* convergence dependent on the condition number of $\mathbf{A}$. The first, which is a worst case scenario, can be proven in a few different ways using our construction of the conjugate directions. The second and third require theory of Krylov subspace methods and estimates of the eigenvalues of polynomials of matrix $\mathbf{A}$.

Theorem 1.3 *The iteration defined by the CG algorithm for $\mathbf{A}$ $n \times n$ SPD matrix converges in at most n steps.*

Theorem 1.4 *If we denote the error $e_k = x_k - x^*$ $(Ax^* = b)$ for the CG iteration, then*

$$||e_{k+1}||_{\mathbf{A}} \leq \left(\frac{\sqrt{\kappa(\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{A})} + 1} \right) ||e_k||_{\mathbf{A}} \leq \left(\frac{\sqrt{\kappa(\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{A})} + 1} \right)^{k+1} ||e_0||_{\mathbf{A}}$$

This tells us that *the better conditioned our matrix is, the better rate of linear convergence we can expect.*

Theorem 1.5 *Assume $\mathbf{A}$ $n \times n$ SPD matrix has r distinct eigenvalues. Then, the iteration defined by the CG algorithm converges in at most r steps.*

This theorem tells us that *The more clustered our eigenvalues are, the less amount of steps CG will take.* In practice, these last two theorems inform what we look for in a **preconditioner** of matrix $\mathbf{A}$: the new preconditioned system should be well-conditioned, and eigenvalues should be clustered around one or a few values.

Finally, this is what a preconditioned CG algorithm looks like, for an SPD preconditioner matrix $\mathbf{M} = \mathbf{C}\mathbf{C}^{\mathbf{T}}$.

```
Given $x_0$,    preconditioner $\mathbf{M}$
$r_0 = b - Ax_0$;    $p_0 = r_0$;    $k = 0$;
solve : $My_0 = r_0$
while  $||r_k|| > \varepsilon$
    $q_k = \mathbf{A}p_k$
    $\alpha_k = \frac{r_k^T y_k}{p_k^T q_k}$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k - \alpha_k q_k$
    solve : $My_{k+1} = r_{k+1}$
    $\beta_k = \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k}$
    $p_{k+1} = y_{k+1} + \beta_k p_k$
    $k = k + 1$;
end
```

This CG algorithm can be shown to solve the equivalent system:

$$\mathbf{LAL}^{\mathbf{T}} y = \mathbf{L}b$$

for $\mathbf{L} = \mathbf{C}^{-1}$ and $x = \mathbf{L}^{\mathbf{T}} y$. You can think of $\mathbf{C}$ as the Cholesky factor of $\mathbf{M}$, and in fact, a good general choice of preconditioner for sparse $\mathbf{A}$ is known as *incomplete Cholesky.*