Conjugate Gradients (CG)

A n×n
A symmetric positive definite

Since $A$ is spd, it gives a norm

$$\| x \|_A = (x^T A x)^{1/2} \quad \text{"A-norm"}$$

CG has the following property:
In each iteration $k$, it finds $x_k \in \mathbb{R}^n$ that minimizes the A-norm of the error $e_k$. I.e.,

$$\| e_k \|_A = \min_{x_k \in \mathbb{R}^n} \| e_k \|_A$$
Solution: $\mathbf{Ax}^* = \mathbf{b}$

$$\min_{\mathbf{x} \in \mathcal{K}_k} (\mathbf{x}^*-\mathbf{x}_k)^T \mathbf{A} (\mathbf{x}^*-\mathbf{x}_k)$$

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

$$\delta \phi = \frac{1}{2} \delta \mathbf{x}^T \mathbf{A} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \delta \mathbf{x} - \mathbf{b}^T \delta \mathbf{x}$$

$$\Delta \phi = \mathbf{A} \mathbf{x} - \mathbf{b}$$

$$\phi(\mathbf{x}-\mathbf{x}^*) = \frac{1}{2} (\mathbf{x}-\mathbf{x}^*)^T \mathbf{A} (\mathbf{x}-\mathbf{x}^*)$$

$$+ \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x}^* + (\mathbf{x}-\mathbf{x}^*)^T \mathbf{A} \mathbf{x}^*$$

$$- \mathbf{b}^T (\mathbf{x} - \mathbf{x}^*) - \mathbf{b}^T \mathbf{x}^* + c$$

$$\langle \mathbf{Ax}^* = \mathbf{b} \rangle = \frac{1}{2} \mathbf{e}^T \mathbf{A} \mathbf{e} + \frac{1}{2} \mathbf{b}^T \mathbf{x}^* + \mathbf{e}^T \mathbf{b} - \mathbf{b}^T \mathbf{e} - \mathbf{b}^T \mathbf{x}^* + c$$

$$\phi(\mathbf{x}^* + \mathbf{e}) = \frac{1}{2} \mathbf{e}^T \mathbf{A} \mathbf{e} - \frac{1}{2} \mathbf{b}^T \mathbf{x}^* + c$$

$$= \frac{1}{2} \mathbf{e}^T \mathbf{A} \mathbf{e} + \text{constant}$$
\[ x_0, \quad r_0 = b - Ax_0, \quad s_0 = r_0 \]

for \( k = 0, 1, 2, \ldots \)
\[
\alpha_k = \begin{array}{c}
\alpha_k \\
X_{k+1} = x_k + \alpha_k s_k \\
r_{k+1} = r_k - \alpha_k A s_k \\
s_{k+1} = \begin{array}{c}
\end{array}
\end{array}
\]

end
C.G.

**Step size** \( \alpha_k \)

\[ f(x_k + \alpha_k s_k) \] one-dim. minimization

\[ \phi(\alpha_k) = f(x_k + \alpha_k s_k) \]

\[ \frac{d\phi}{d\alpha}(\alpha_k) = \nabla f(x_k + \alpha_k s_k)^T s_k = 0 \]

For e.g., \( \nabla f(x) = b - Ax = r \)

\[ \Rightarrow \left( \frac{d\phi}{d\alpha}(\alpha_k) = 0 \right) \Rightarrow \]

\[ r_{ke}^T s_k = [b - A(x_k + \alpha_k s_k)]^T s_k \]

\[ = b^T s_k - x_k^T A^T s_k - \alpha_k s_k^T A^T s_k = 0 \]

\[ = (b - Ax_k)^T s_k - \alpha_k s_k^T A^T s_k = 0 \]

\[ \Rightarrow \alpha_k = \frac{r_{ke}^T s_k}{s_k^T A^T s_k} \]
Steepest Descent Method

or gradient descent

this will zigzag very slowly toward the solution

\[ x_0, r_0 = b - Ax_0, s_0 = r_0 \]

for \( k = 0, 1, 2, \ldots \)

\[ \alpha_k = \frac{r_k^T s_k}{s_k^T A s_k} \]

\[ x_{k+1} = x_k + \alpha_k s_k \]

\[ s_{k+1} = s_k - \alpha_k A s_k \]

end

Progress

\[ \frac{\phi(x_k) - \phi(x^*)}{\phi(x_{k-1}) - \phi(x^*)} \leq 1 - \frac{1}{\text{Cond } A} \]
What if we try \( S_k \) mutually orthogonal? If we tried to do \( n \) steps with orthogonal directions, as illustrated, the step size we would need would not be the one for exact line search.

Mathematically, we want to start at \( x_0 \) and reach \( x^* \) in \( n \) steps along \( S_k \):

\[
x^* = x_0 + \sum_{k=0}^{n-1} \alpha_k S_k
\]

Let \( e_0 = x_0 - x^* \) and \( e_k = x_k - x^* \)

Find \( \alpha_k \):

\[
e_0 = x_0 - x^* = -\sum_{k=0}^{n-1} \alpha_k S_k
\]

\[
S_j^T e_0 = -\sum_{k=0}^{n-1} \alpha_k S_j^T S_k = \langle \text{assume } S_j^T S_k = 0, j \neq k \rangle
\]

\[
\Rightarrow \quad \alpha_j = \frac{S_j^T e_0}{S_j^T S_j} = \frac{S_j^T e_k}{S_j^T S_j}
\]

Problem: we don't know \( e_0 \) or \( e_k \).
What about $A$-orthogonal directions $s_k$?

\[
e_0 = -\sum_{k=0}^{n-1} \alpha_k s_k
\]

\[
s_j^T A e_0 = -\sum_{k=0}^{n-1} \alpha_k s_j^T A s_k
\]

\[
= -\alpha_j s_j^T A s_j
\]

\[\Rightarrow \alpha_j = -\frac{s_j^T A e_0}{s_j^T A s_j} = -\frac{s_j^T A e_j}{s_j^T A s_j} = \frac{s_j^T r_j}{s_j^T A s_j}
\]

This is a quantity we know, and it is the one given by exact line search along $s_j$. 
But what are the search directions \( S_k \)?

A-orthogonal search directions will allow us to terminate in \( n \) steps:

\[
e_0 = \sum_{i=0}^{n-1} \alpha_i s_i
\]

\[
S_k^T A e_0 = -\sum_{i=0}^{n-1} \alpha_i s_i^T A s_i = -\alpha_k s_k^T A s_k
\]

\[
\Rightarrow \alpha_k = -\frac{S_k^T A e_0}{S_k^T A s_k} = -\frac{S_k^T A e_k}{S_k^T A s_k} = \frac{S_k^T r_k}{S_k^T A s_k}
\]

\[
x_0 + e_0 = x, \quad e_0 = x - x_0
\]

\[
x_0 + \sum_{k=0}^{n-1} \alpha_k s_k = x
\]

Create the search directions through Gram-Schmidt A-orthogonalization:

\[
U_0, \ldots, U_{n-1}
\]

\[
s_i = u_i + \sum_{j=0}^{i-1} \beta_{ij} s_j
\]

\[
0 \leq e_k, S_k^T A s_i = S_k^T A u_i + \sum_{j=0}^{i-1} \beta_{ij} S_k^T A s_j = 0
\]

\[
\Rightarrow \beta_{ik} = -\frac{S_k^T A u_i}{S_k^T A s_k}
\]

\[\text{need to keep all old } s_i \text{ in memory, then orthogonalize against all.}\]
\[ S_k = u_k + \sum_{j=0}^{k-1} \beta_{kj} s_j \]

Gram-Schmidt A-orthogonalization:

\[ \begin{align*}
0 = s_i^T \hat{A} s_k &= s_i^T A u_k + \sum_{j=0}^{k-1} \beta_{kj} s_i^T A s_j \\
&= s_i^T A u_k + \beta_{ki} s_i^T A s_i \\
\Rightarrow \beta_{ki} &= -\frac{s_i^T A u_k}{s_i^T A s_i}
\end{align*} \]

CG: use Krylov subspaces, use residuals instead of arbitrary \( u_k \), and A-orthogonalize residuals:

\[ \beta_{ki} = -\frac{s_i^T A r_k}{s_i^T A s_i} = -\frac{r_k^T (A s_i)}{s_i^T A s_i} \]

\[ = \langle A s_i = \frac{1}{\alpha_i} (r_i - r_{i+1}) \rangle \]

\[ = \frac{r_k^T (r_{i+1} - r_i)}{\alpha_i s_i^T A s_i} = \frac{r_k^T r_{i+1} - r_k^T r_i}{\alpha_i s_i^T A s_i} = \frac{\sum_{i=k-1}^i r_k^T r_i}{\alpha_i s_i^T A s_i} \]

\[ \begin{cases} r_k^T r_k & \text{if } i = k-1 \\ 0 & \text{otherwise} \end{cases} \]

For (1) and (2), see next page.

There is only one non-zero \( \beta_{ki} \in \beta_k \)
1. residuals orthogonal
   \[ r_k^T S_j = 0, \quad j < k \]

   \[ r_k^T S_j = -(Ae_i)^T S_j \]

   \[ = -S_j^T A \left[ - \sum_{i=k}^{n} \alpha_i s_i \right] = 0 \]

   \[ S_j = r_j + \sum_{i=0}^{j-1} \beta_i s_i \]

   So

   \[ r_k^T r_j = r_k^T \left[ S_j - \sum_{i=k}^{j-1} \beta_i s_i \right] = 0 \]

2. \[ \alpha_i s_i^T A s_i = r_i^T r_i \]

   \[ r_{i+1} = r_i - \alpha_i A s_i \Rightarrow \alpha_i A s_i = r_i - r_{i+1} \]

   \[ \alpha_i s_i^T A s_i = s_i^T \left[ r_i - r_{i+1} \right] \]

   \[ = (r_i + \sum_{j=0}^{i-1} \beta_j s_j)^T (r_i - r_{i+1}) = r_i^T r_i \]
Conjugate Gradients for $Sx = b$

- Applies to s.p.d. matrices $S$
- In theory gives the exact result in $n$ steps
- In practice, gives good result in much fewer than $n$ steps.
- Residuals are orthogonal $r_k^T r_j = 0$
- Search directions are $A$-orthogonal $S_k^T A s_j = 0$

C.G.

$x_0 = $ initial guess

$r_0 = b - Ax_0$

$s_0 = r_0$

for $k = 0, 1, 2, \ldots$

\[
\alpha_k = \frac{r_k^T r_k}{s_k^T A s_k}
\]

$x_{k+1} = x_k + \alpha_k s_k$

$r_{k+1} = r_k - \alpha_k A s_k$

$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$

$s_{k+1} = r_{k+1} + \beta_{k+1} s_k$

end
Convergence of CG

\[ \| x - x_k \|_A \leq 2 \| x - x_0 \|_A \left( \frac{\sqrt{\lambda_{\text{max}}^2 - \sqrt{\lambda_{\text{min}}^2}}}{\sqrt{\lambda_{\text{max}}^2 + \sqrt{\lambda_{\text{min}}^2}}} \right)^k \]

\[ K_2(A) = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} \]

\[ \| x - x_k \|_A \leq 2 \| x - x_0 \|_A \left( \frac{\sqrt{k} - 1}{\sqrt{k} + 1} \right)^k \]
Preconditioning $Ax = b$

Find $P$ close to $A$ and solve

$$P^{-1}Ax = P^{-1}b$$

with the idea that solver converges more quickly on $P^{-1}A$ than on $A$.

$P^{-1}$ should be reasonable to computed. (Not actually computed, but represents solve $P^{-1}c \Rightarrow y | Py = c$)

Frequent choices of $P$:

1. main diagonal of $A$ (Jacobi)
2. triangular part of $A$ (Gauss-Seidel)
3. $P = L_0 U_0$ incomplete LU, avoid fill-in
4. $P =$ same diff matrix but on coarser grid (multigrid)