The Conjugate Gradient method

Iterative solver for symm pos def systems

\[ Ax = b \quad A \in \mathbb{R}^{n \times n} \text{ symm pos def} \]

\[ \min_{x \in \mathbb{R}^n} \frac{1}{2} x^T A x - x^T b = f(x) \]

\[ \begin{align*}
0 f(x) &= A x - b \\
0 \nabla f(x) &= A
\end{align*} \]

Gradient method:

\[ x_{k+1} = x_k - \alpha_k \nabla f_k \]

where \( \alpha_k \) solves \( \min_{\alpha} f(\alpha) = f(x_k - \alpha \nabla f_k) \), i.e.:

\[ f'(\alpha) = - \nabla f (x_k - \alpha \nabla f_k)^T \nabla f_k \]
\[ = - \nabla f_k^T A (x_k - \alpha \nabla f_k) + \nabla f_k^T b \]
\[ = - \nabla f_k^T \alpha + \alpha \nabla f_k^T A \nabla f_k \]

\[ f''(\alpha) > 0 \quad \Rightarrow \quad \alpha_k = \frac{\nabla f_k^T b}{\nabla f_k^T A \nabla f_k} \]

Possible to show \( \nabla f_k^T \nabla f_{k+1} = 0 \). Method can take long to converge.

"Zig-zagging" 2D:

The linear conjugate gradient method

\[ x_{k+1} = x_k + \alpha_k p_k \; \text{, where } \alpha_k p_k \text{ is determined st.} \]

\[ x_k \text{ solves} \]

\[ \min_{x \in \mathbb{R}^n + \text{span}\{p_0, \ldots, p_{k-1}, r_k\}} \frac{1}{2} x^T A x - b^T x \]  \hspace{1cm} (1)

where \( r_k = - \nabla f_k = b - A x_k \)
\[ \min \frac{1}{2} x^T Ax - x^T r_0 \]
\[ \hat{x} \in \text{span} \{ p_0, \ldots, p_{k-1}, r_k \} \]
where \( \hat{x} = x - x_0 \)

It can be shown that \( \hat{x}_{k+1} \) solves (2) iff
\[ (A \hat{x}_{k+1} - r_0)^T v = 0 \quad \forall v \in \text{span} \{ p_0, \ldots, p_{k-1}, r_k \} \]

\[ (A \hat{x}_{k+1} - b)^T v = 0 \]
\[-\nabla f_{k+1} \]

\[ (\sim \text{Galerkin}) \]

Intuitively:
- Optimal \( x_{k+1} \):
  - No descent direction
  - \( x \in \text{span} \{ p_0, \ldots, p_{k-1}, r_k \} \)

- Non-optimal \( x \):
  - One can find a descent direction (and better points)

In previous step: \( \hat{x}_k \) solves:
\[ \min \frac{1}{2} \hat{x}_k^T A \hat{x}_k - \hat{x}_k^T r_0 \]
\[ \hat{x}_k \in \text{span} \{ p_0, \ldots, p_{k-1} \} \]

Since \( x_{k+1} = x_k + r_k p_k \):
\[ 0 = (A x_{k+1} - b)^T p_j = (A x_k - b)^T p_j + r_k p_k^T A p_j \]
\[ \Rightarrow p_k \perp A \text{-orthogonal} \quad \text{inner prod} \quad (u, v)_A = u^T A v \]

to previous \( k \) directions

\( \Rightarrow \) get \( p_k \) using Gram - Schmidt orthogonalization, which will greatly simplify.

\[ \begin{cases} 0 & p_0 = b \\ p_k^2 = 1 - \sum_{j=0}^{k-1} \frac{r_k^T A p_j}{p_j^T A p_j} \end{cases} \]
(65)
Now that we know in which direction to go, we can use gradient method to find
\[
\alpha_k = \arg\min_{\alpha} \phi(x) = f(x_k + \alpha p_k)
\]
\[
= \begin{bmatrix} \alpha_k \\ p_k \end{bmatrix} = \begin{bmatrix} p_k^T p_k \\ p_k^T A p_k \end{bmatrix}.
\]

A closer look to the subspace used gives:
\[
\text{span}\{p_0, \ldots, p_{k-1}, r_k\} = \text{span}\{p_0, \ldots, p_k\} = \text{span}\{r_0, \ldots, r_k\} = \mathbb{K}_{k+1}(A, r_0) = \text{span}\{r_0, A r_0, A^2 r_0, \ldots, A^k r_0\} = \text{Krylov subspace}
\]
\[
eq \mathbb{K}_2(A, r_0)
\]
\[
\text{why?} \quad r_0 = b - Ax_0 \\
\quad r_1 = b - Ax_1 = b - A(x_0 + \alpha(b - Ax_0)) \in \mathbb{K}_2(A, r_0) \\
\text{etc...}
\]

\text{Why is this useful?}

\text{optimality conditions:} \quad r_{k+1}^T v = 0 \quad \forall v \in \mathbb{K}_{k+1}(A, r_0) \\
\quad r_k^T v = 0 \quad \forall v \in \mathbb{K}_k(A, r_0) \\
\quad \text{take} \quad p_i \in \mathbb{K}_{k-1}(A, r_0) \Rightarrow A p_i \in \mathbb{K}_k(A, r_0) \\
\Rightarrow r_k^T A p_i = 0 \quad i = 0, \ldots, k-2
\]

Thus (65) reduces to:
\[
p_k = r_k - \begin{bmatrix} p_k^T A p_k & 0 \\ 0 & 1 \end{bmatrix} p_k - 1
\]
Can show: 
\[
\alpha_k = \frac{n_k \| h_k \|^2}{p_k^T A p_k} \\
\beta_k = \frac{n_k \| h_k + p_k \|^2}{n_k \| h_k \|^2}
\]

\( A \) is indefinite \( \Rightarrow \) \( \text{min} \{ \lambda \} \neq 0 \)
\( \Rightarrow \) \( A \) has no solution

**What happens if \( A \) is indefinite?**

\( \Rightarrow \) \( \exists k \) for which \( p_k^T A p_k < 0 \)

\( \Rightarrow \) "negative curvature" direction \( \Rightarrow \) stop iterations

**Convergence results:**

Convergence properties determined by eigenvalues of \( A \).

- The fewer "eigen" the better.
- Many iterations
- Few (\( \sim 3 \)) iterations.

**Preconditioning:**

\( A x = b \) \( \Rightarrow \) \( M A x = M b \)

\( M = \) preconditioner

- Easy to compute
- And somehow transforms spectrum of \( A \) from

\( (S1) \) to \( (S2) \)