Why *"Conjugate Gradients"* ?

- It's clear that CG is a KSP, which pretty much defines the method.
- Why do we call it *"Conjugate Gradients"* ?
- Consider the scalar

$$\begin{aligned}
\phi(\underline{x}_k) &= \frac{1}{2}\|\underline{e}_k\|_A^2 && (1)\\
&= \frac{1}{2}\underline{e}_k^T A\underline{e}_k \;=\; \frac{1}{2}(\underline{x}-\underline{x}_k)^T A(\underline{x}-\underline{x}_k) && (2)\\
&= \frac{1}{2}\left[\underline{x}^T A\underline{x} \,+\, \underline{x}_k^T A\underline{x}_k \,-\, 2\underline{x}_k^T A\underline{x}\right] && (3)\\
&= \frac{1}{2}\left[\underline{x}^T A\underline{x} \,+\, \underline{x}_k^T A\underline{x}_k \,-\, 2\underline{x}_k^T \underline{b}\right] && (4)
\end{aligned}$$

- Let $\underline{e}(\underline{v}) \;:=\; \underline{x} - \underline{v}$ and consider $\phi(\underline{v})$, for $\underline{v} = [v_1 \; v_2 \; \dots \; v_n]^T$.
- Consider the optimization problem of finding a $\underline{v}$ that will minimize $\phi(\underline{v})$.
- If we are at a given $\underline{v}$, what direction should we proceed to decrease $\phi(\underline{v})$?
- The standard approach is to evaluate the *gradient* of ϕ,

$$\nabla\phi(\underline{v}) \;=\; \left[\frac{\partial\phi}{\partial v_i}\right] \tag{5}$$

$$\frac{\partial}{\partial v_i}\underline{x}^T A\underline{x} \;=\; 0 \tag{6}$$

$$\frac{\partial}{\partial v_i}\underline{v}^T\underline{b} \;=\; \frac{\partial}{\partial v_i}\sum_{j=1}^{n} b_j\, v_j \;=\; b_i \tag{7}$$

$$\frac{\partial}{\partial v_i}\underline{v}^T A\underline{v} \;=\; \frac{\partial}{\partial v_i}\sum_{j=1}^{n}\sum_{k=1}^{n} v_j a_{jk} v_k \;=\; \sum_{j=1}^{n} v_j a_{ji} \;+\; \sum_{k=1}^{n} a_{ik} v_k \tag{8}$$

$$=\; 2\sum_{k=1}^{n} a_{ik} v_k \;=\; 2\,[A\underline{v}]_i\,. \tag{9}$$

- Thus, the gradient of ϕ is

$$\nabla\phi \;=\; A\underline{v} \;-\; \underline{b} \;=\; -\underline{r}. \tag{10}$$

- Our *descent* direction, is therefore $\underline{r}$.

- The *steepest descent* algorithm is almost identical to CG:

- Starting with $\underline{x} = 0$, $\underline{p} = 0$, $\underline{w} = 0$, $\underline{r} = \underline{b}$, and $\rho_1 = 1$;

$$\text{for} \quad k = 1, \ldots, k_{\max} \tag{11}$$

$$\rho_0 \; = \; \rho_1, \qquad \rho_1 \; = \; \underline{r}^T \underline{r} \tag{12}$$

$$\underline{p} \; = \; \underline{r} \tag{13}$$

$$\underline{w} \; = \; A\underline{p}, \quad \alpha \; = \; \frac{\rho_1}{\underline{p}^T \underline{w}} \tag{14}$$

$$\underline{x} \; = \; \underline{x} \; + \; \alpha \underline{p} \tag{15}$$

$$\underline{r} \; = \; \underline{r} \; - \; \alpha \underline{w}. \tag{16}$$

- All we've done is turned off the correction to $\underline{p}$ by setting $\beta = 0$.

- Good news: CG starts with a good direction, and makes a small correction to obtain a *projector*.

- Bad news: Steepest descent requires $O(\kappa)$ iterations, not $O(\sqrt{\kappa})$.

- Demo: `stp_des.m`