Optimization

Pt 2 Steepest Descent & Newton's Method

Recap Optimize $\varphi(x) = \frac{1}{2} x^* A x - x^* b$

A ← Symmetric Positive Def.

Conjugate Gradient

$x_0 = 0,\ r_0 = b,\ p_0 = r_0$

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

$\alpha_n = (r_{n-1}^T r_{n-1}) / (p_{n-1}^T A p_{n-1})$

$x_n = x_{n-1} + \alpha_n p_{n-1}$

$r_n = r_{n-1} - \alpha_n A p_{n-1}$

$\beta_n = (r_n^T r_n) / (r_{n-1}^T r_{n-1})$

$p_n = r_n + \beta_n p_{n-1}$

Steepest Descent

$x_0 = 0,\ r_0 = b$

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

$\alpha_n = \dfrac{r_{n-1}^T r_{n-1}}{r_{n-1}^T A r_{n-1}}$

$x_n = x_{n-1} + \alpha_n r_{n-1}$

$r_{n+1} = r_n - \alpha_n A r_n$

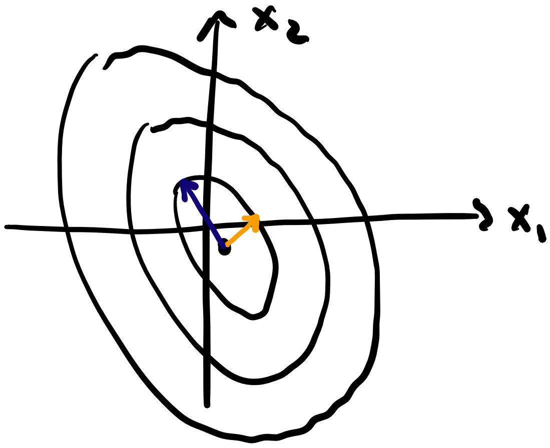
Recall, $-\nabla \varphi(x) = r_n$

⇒ Steepest Descent looks like CG with search direction $-\nabla \varphi = r_n$ rather than p_n.

⇒ How does this affect convergence?

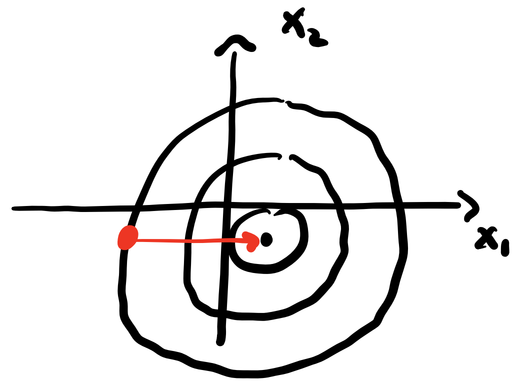
Convergence

level sets of $\varphi(x)$



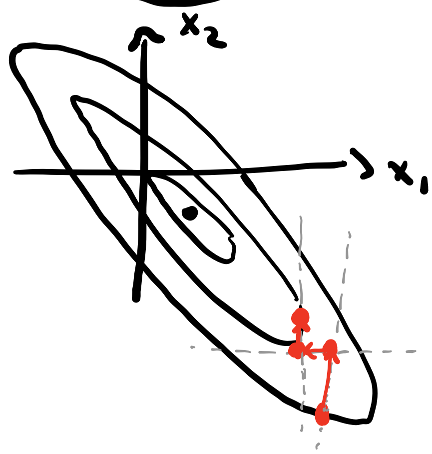
$\kappa(A)=1$

$\lambda_1 = \lambda_2$



$\lambda_1 >> \lambda_2$

$\kappa(A) >> 1$



$$A = \begin{bmatrix} | & | \\ v_1 & v_2 \\ | & | \end{bmatrix} \begin{bmatrix} \lambda_1 & \\ & \lambda_2 \end{bmatrix} \begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \end{bmatrix}$$

Similar to CG, well-conditioned A leads to rapid convergence and ill-conditioned problems can lead to slow convergence.

CG

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq \left[\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right]^n$$

SD

$$\frac{\|e_n\|_A}{\|e_0\|_A} \leq \left[\frac{\kappa-1}{\kappa+1}\right]^n$$

where $\kappa = \kappa(A)$.

Why does CG do better than SD?

$\Rightarrow$ SD tends to zig-zag back & forth, taking repeated steps in the same direction. It is only locally optimal.

$\Rightarrow$ CG "remembers" previous search directions due to implict orthogonalization (short recurrence). It optimizes over the whole expanding Krylov space & never repeats a direction.

$\Rightarrow$ CG pick search directions that are A-orthogonal. The A-inner product weights directions v_1, v_2 by λ_1, λ_2, which has the effect of "broadening" the level-sets of φ.

SD and CG both have generalizations to more general nonlinear functions φ.

Newton's Method

Newton's method is a rootfinding method.

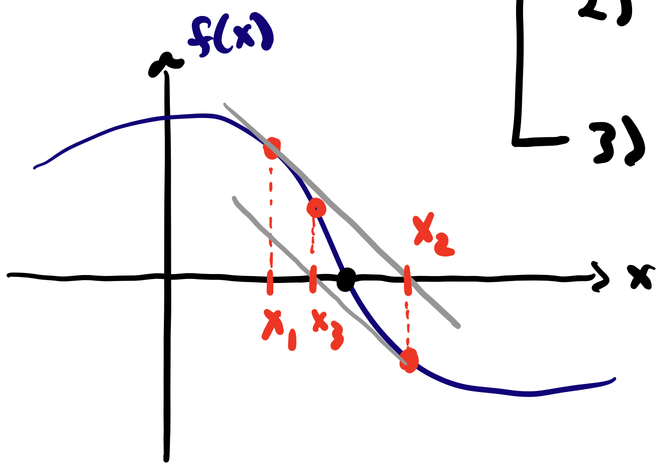
$\Rightarrow$ solve $f(x) = 0$

Of course, it can also be used to find critical points (min, max, etc.) of f by

$$\Rightarrow \text{solve } f'(x) = 0$$

<u>Idea:</u>

1) Guess x_k
2) Find root of tangent line at x_k (i.e. linear approx)
3) Set $x_{k+1} = $ root



Eqn of tangent line: $\ell(x) = f(x_k) + f'(x_k)(x - x_k)$

root $\ell(x_{k+1}) = 0 \iff x_{k+1} = x_k - f(x_k)/f'(x_k)$

Newton's method typically converges quadratically,

$$|e_{n+1}| \leq M |e_n|^2$$

for smooth f and x_0 sufficiently close to root.

Convergence analysis

f'' continuous, root $= x_*$

Taylor expand: $f(x_*) = f(x_k) + f'(x_k)(x_* - x_k) + R_1$

Reminder $R_1 = \frac{1}{2!} f''(\xi_k)(x_* - x_k)^2$ $\left[\begin{array}{l}\xi_k \text{ between} \\ x_k \text{ and } x_*\end{array}\right]$

$$f(x_*) = 0 \iff f(x_k) + f'(x_k)(x_* - x_k) + R_1 = 0$$

$$\overset{f'(x_k)\neq 0}{\iff} \frac{f(x_k)}{f'(x_k)} + x_* - x_k = \frac{-f''(\xi_k)}{2f'(x_k)}(x_* - x_k)^2$$

$$\overset{x_{k+1} = \text{update}}{\implies} \underbrace{x_* - x_{k+1}}_{e_{k+1}} = -\frac{f''(\xi_k)}{2f'(x_k)} \underbrace{(x_* - x_k)^2}_{e_k}$$

$\left|\frac{f''(\xi_k)}{2f'(x_k)}\right| \leq M$ when
1. f'' locally bdd
2. $f' \neq 0$ locally near x_*
3. x_k suff. close to x_*

If $f'(x_*) = 0$, then convergence is <u>usually</u> linear.