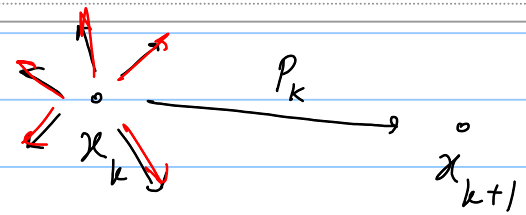
Line Search Methods.

p? $\longrightarrow$ $$\boxed{\nabla_p f(x_k + \varepsilon p) = 0}$$



$$p_k = \underset{p}{\arg\min} \left\| \nabla_p f(x_k + \varepsilon p) \right\|$$

$$\left. \frac{\partial^2 f}{\partial x_1 \partial x_2} = \frac{\partial^2 f}{\partial x_2 \partial x_1} \right\}$$

$$p_N = -(\nabla^2 f)^{-1} \nabla f \quad \checkmark$$

Line Search.

$$x_k \xrightarrow{\alpha p_k} x_{k+1}$$

$$x_{k+1} = x_k + \alpha p_k$$

α → Step length → TBD

"Learning rate"

p_k → fixed

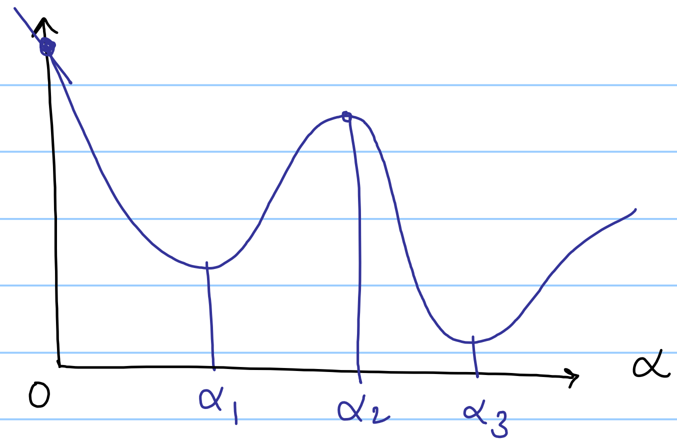
If we take small values of α

$$\nabla f_k^T p_k < 0$$

Define: $\phi(\alpha) = f(x_k + \alpha p_k)$, $\alpha > 0$, $\because p_k$ is a descent dir

What is the desired quality of α?

$$\phi'(\alpha) = 0$$



Starting at $\alpha = 0$

$\phi(0)$ & $\phi'(0)$

Computing $\underbrace{\phi(\alpha) \text{ and } \phi'(\alpha)}$ is EXPENSIVE.

Exact α $\rightarrow$ Not practical

$\Rightarrow$ inexact α $\rightarrow$ inexact line search.

$\rightarrow$ Wolfe Conditions.

~~Aside.~~

$\frac{d}{d\alpha}\phi(\alpha) \longleftrightarrow f, p$?

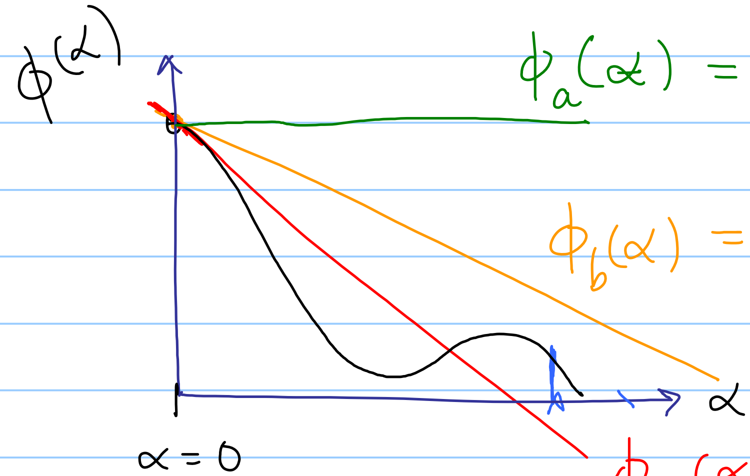
$\phi(\alpha) = f(\underbrace{x_k + \alpha p_k})$

$x_k = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad p_k = \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \quad \begin{bmatrix} x_1 + \alpha p_1 \\ x_2 + \alpha p_2 \end{bmatrix}$

$$\frac{d\phi(\alpha)}{d\alpha} = \frac{\partial \phi}{\partial x_1}\underbrace{\frac{d x_1}{d\alpha}}_{p_1} + \frac{\partial \phi}{\partial x_2}\underbrace{\frac{d x_2}{d\alpha}}_{p_2}$$

$$\frac{d\phi(\alpha)}{d\alpha} = \begin{bmatrix} \frac{\partial \phi}{\partial x_1} & \frac{\partial \phi}{\partial x_2} \end{bmatrix}\begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \boxed{\nabla f(x_k + \alpha p_k)^T p_k}$$

① Condition of Sufficient Decrease.



$$\boxed{\phi(0) \ \& \ \phi'(0) \text{ are given.}}$$

$$\phi_a(\alpha) = f(x_k)$$

$$\phi_b(\alpha) = f(x_k) + \alpha c_1 \nabla f_k^T p_k \qquad c_1 \in [0, 1]$$

$$c_1 \approx 10^{-4}.$$

$$\phi_l(\alpha) = f(x_k) + \alpha \underbrace{\nabla f_k^T p_k}_{\phi'(0)} \leftarrow \text{linear approx}$$

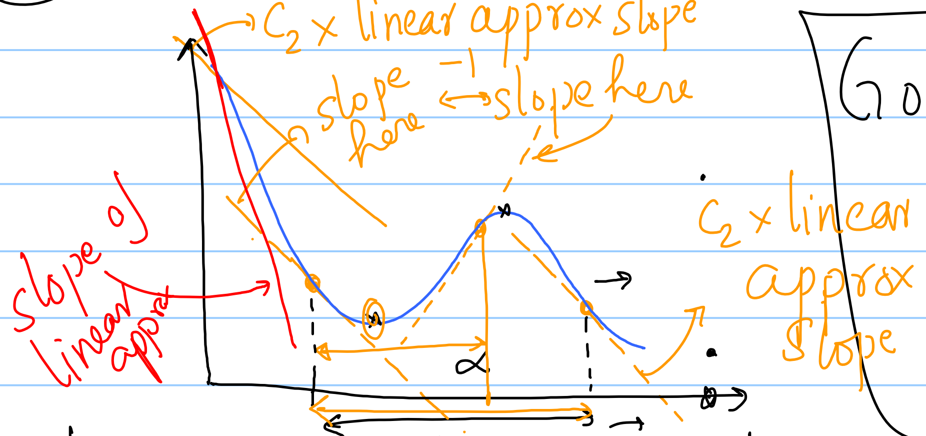
We want that f_{k+1} lie below the linear approximation of f at x_k.

"Armijo" rule

$$\rightarrow f(x_k + \alpha p_k) < f(x_k) + \alpha c_1 \nabla f_k^T p_k. \leftarrow$$

↳ It allows very small values of α

(2) Called the "Curvature Condition"



Goal: $\nabla f(x_k + \alpha p_k) = 0$

$\longrightarrow \phi'(\alpha) = \nabla f(x_k + \alpha p_k)^T p_k$

$\Rightarrow \phi'(\alpha) = 0$

The |derivative| at α be atleast less than the slope of the linear approximation at x_k.

(intution) $|\phi'(\alpha)| < C_2 \times$ Slope of linear approx.

$\because$ p_k is a descent dir, $\phi'(\alpha) < 0$ for small α.

Curvature cond:

$$\underbrace{\underbrace{-\nabla f(x_k + \alpha p_k)^T p_k}_{|\text{slope at } \alpha|}}_{\substack{\text{small } \alpha,\ \geq 0 \\ \text{past } \alpha^*\ \leq 0}} \leq \underbrace{\underbrace{-\nabla f(x_k)^T p_k}_{|\text{slope at } 0|}}_{\geq 0 \text{ always!}} \times c_2$$

$\to$ Allows overshooting!

Strong wolffe condn:

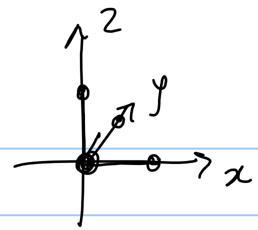
$$\boxed{|\phi'(\alpha)| \leq c_2 |\phi'(0)|} \quad 0 < c_1 < c_2 < 1$$

$O(N)$

∇f?

$$\frac{\partial f}{\partial x_1} = \lim_{\varepsilon \to 0} \frac{f(x_1+\varepsilon) - f(x_1)}{\varepsilon}$$

finite difference.



→ Simplest Algo for finding α.

Backtracking Line Search.

→ ① Start with large α

② Check if $\left[f(x_k + \alpha p_k) \leq f(x_1) + c_1 \nabla f_k^T p_k \right]$

if not true → $\alpha_{new} = \alpha_{old}\, \rho$, $0 < \rho < 1$

if true → accept α.

$\Rightarrow x_{k+1} = x_k + \alpha_k p_k$ is set.

Check if ∇f_{k+1} is 0

if true ✓

if false