Lecture 14

Overview of unconstrained optimization algorithms (Handout from last time)

Line search methods
Descent directions
What goes wrong (and how it differs from simplex)
How to fix it: Wolfe conditions, i.e. goldilocks conditions.

Proof that we can do this (if time)

Terms:
Descent direction
Wolfe conditions
Sufficient decrease (Armijo condition)
Curvature condition
Strong Wolfe conditions

Midterm: March 13th (in class)

Back to min $f(x)$

$f : \mathbb{R}^n \to \mathbb{R}$

Assume twice cont. diff.

2nd order Necessary cond.

$g(\underline{x}) = 0 \qquad H(\underline{x}) \succeq 0$

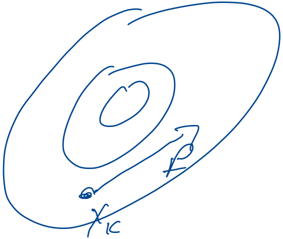
2nd order Suff. cond.

$g(\underline{x}) = 0 \qquad H(\underline{x}) \succ 0$

All methods are iterative

$$\underline{x}^{(k)} \to \underline{x}^{(k+1)}$$

Line Search

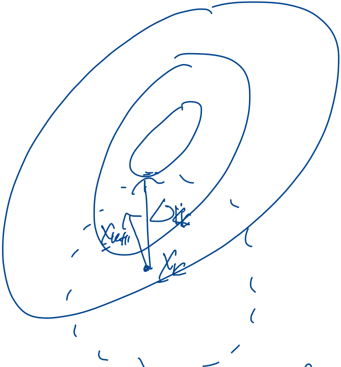


Find $\underline{p}$ then
find α_k s.t. $\underline{x}_{k+1} = \underline{x}_k + \alpha_k$
$f(\underline{x}_{k+1}) < f(\underline{x}_k)$

CS 520 - Purdue University - Computer Science - David F. Gleich - 2025
CS 520 - Purdue Uni

$= p_k$

Trust Region



find reduced $f(x)$

st. $\|\underline{x}_k - \underline{x}_{k+1}\| \le \Delta_k$.

*** Notation Alert ***

We tire of writing indices!

$$\underline{x} = \underline{x}^{(k)}$$

$$\underline{x}^+ = \underline{x}^{(k+1)}$$

$$g = g(\underline{x}^{(k)})$$

$$g^+ = g(\underline{x}^{(k+1)})$$

$$H = H(\underline{x}_k)$$

$$H^+ = H(x_{k+1})$$

$f_k = f(\underline{x}_k)$

$f^+ = f(x_{k+1})$

i.e.

$f(\underline{x} - \alpha g) =$

$f(\underline{x}_k - \alpha_k g(\underline{x}_k))$

Line Search Methods

$$\underline{x}^+ = \underline{x} + \alpha \underline{p}$$

② pick α

① pick $\underline{p}$

How to pick $\underline{p}$?

- Stochastic gradient descent $p \approx -g$
- gradient descent $\underline{p} = -g$
- Newton $p = -H^{-1} g$
- Quasi-Newton Let $B \simeq H$, then $\underline{p} = -B^{-1} g$
- Conj. grad.

Pseudo code for a LS method

```
while not done (i.e. the gradient is large)
    • Pick $p$ as a search dir
    • Pick $\alpha$ as a step length
    • Move $x \leftarrow x + \alpha p$
```

Key questions: does any p work?
How to pick α?

Goal: minimize $f(x)$

First, we will analyze directions that make f small(er)

Definition:

p is a descent direction at x_k if $p^T \underset{g(x_k)}{\underset{\uparrow}{g}} < 0$.

Why only look @ descent directions?

We can always make f^+ smaller!

$$f(x+\alpha p) = f_k + \gamma p^T g + O(\gamma^2).$$

(By Taylor's theorem)

If γ is small enough, then $f^+ < f_k$...

Lemma Let B be a sym. pos. def. Matrix. Then any p st

$$Bp = -g$$

gives a descent dir. (*) unless you are at a stationary pt. (*)

Pf: $p = -B^{-1} g$

If B is spd. then B^{-1} is too!

$$p^T g = -\underbrace{g^T B^{-1} g}_{\geq 0} < 0.$$

Eg. $B = I \Rightarrow$

$$p = -g$$

gradient descent.

Newton:

If H is pos. def. then $p = -H^{-1} g$ is a des. dir.

Quasi-Newton

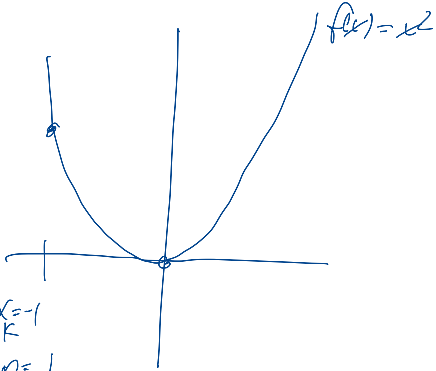
$B \simeq H$, but always sym. pos. def.

We always assume p is a descent direction!

Is a descent direction enough to minimize f?

No!

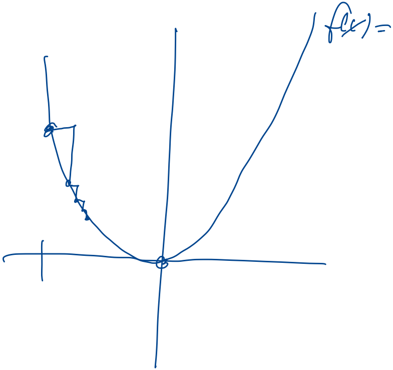
1. Example 1.



$x_k = -1$

$p = 1.$

$x_k = x_k + \alpha$: Line Search.



Steps $\alpha_k =$

$\frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32} \ldots$

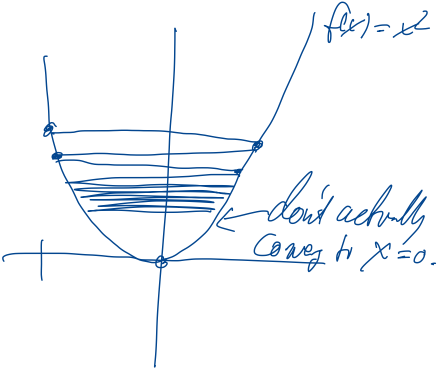
$x_{k+1} = x_k + \frac{1}{2^{k+1}}$

$x_1 = -1$, then $f(x_{k+1}) < f(x_k)$

x^2

But I never get to a minimizer

The problem is tiny steps.



Map previous non-converg sequen to long steps.

$\alpha > 1$ but $x_k \to \pm 1/4$

and $f(x_{k+1}) < f(x_k)$

We take big Steps but... Only reduce f by a tiny amount.

So we need Some Conditions to make progress toward a minimizer.

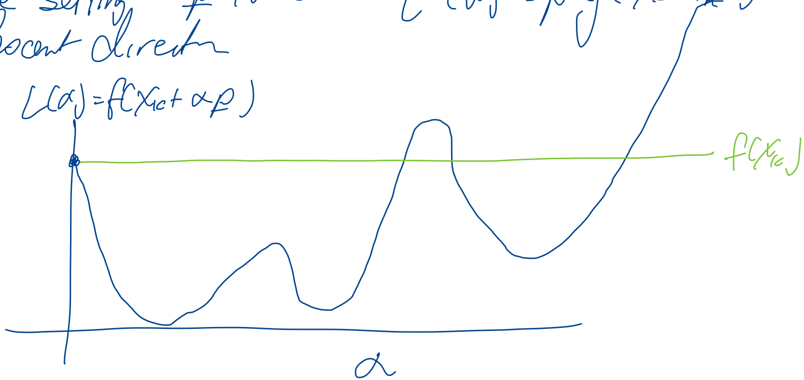
Goal: look at a condition on α s.t. we make progress toward making f small and making g flat.

The setting: p is a descent direction

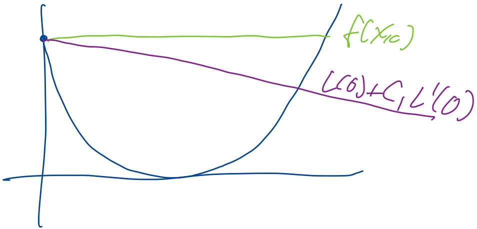
$L(\alpha) = f(x_k + \alpha p)$

⊕ Don't want to minimize $L(\alpha)$ Exactly! Just want to be quick and "good enough".

Also: We can compute $L'(\alpha) = p^T g(x_k + \alpha p)$



To avoid useless large steps, ⇒ if we take a big step, it should really help make f small.



$1 > c_1 \geq 0$

1. A tiny step

w