Lecture 3

So far

Problem:

minimize f over Ω (usually $\mathbb{R}^n$)

- necessary condition
- sufficient condition

Convex functions: main point

If f is convex, then every local min is a global min.

Next Topic: algorithm for finding minima.

- function of a single variable
  several variables
- concrete method
- general method

Algorithm for minimizing function of ~~1 variable~~ single $\quad f: \mathbb{R} \to \mathbb{R}$

Basic plan with several variants

a). choose a point x_0

b). approximate f near x_k by a function that is easier to minimize

c). find a minima pt for simplest function

d). call that x_{k+1}

repeat then

Hopefully, $x_k \to$ min pt. of x.

Newton's Method

Ex. Newton's method

- approximate f near x_k by $q(x) = f(x_k) + (x - x_k) \cdot f'(x_k) + \frac{1}{2}(x - x_k)^2 f''(x_k)$

(2nd order Taylor Polynomial)

Find critical pt of q (minima):

set $q' = 0$ and solve:

Note $q'(x) = f'(x_k) + (x - x_k) \cdot f''(x_k)$

x_{k+1} solves $q'(x_{k+1}) = 0 \Rightarrow x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$

Thus we get the Algorithm

- guess x_0
- $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$ for $k = 0, 1, 2, 3, \dots$

The (Calculus) Newton's Method for solving equation $\boxed{g(x) = 0}$, $g = f'$

For solving $g(x) = 0$, idea was approximate g by tangent line of x_k

$l(x) = g(x_k) + (x - x_k) g'(x_k)$

Solve $l(x_{k+1}) = 0$. This gives $x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)}$

Ex. minimize $f(x)=x^4-4x^2=(x^2-2)^2-4$

we have $f'(x)=4x^3-8x=4x(x^2-2)$

$f''(x)=12x^2-8=4(3x^2-2)$

$$x_{k+1}=x_k-\frac{x_k^3-2x_k}{3x_k^2-2}=\frac{2x_k^3}{3x_k^2-2}$$

fact: $x_0=3$, then: $x_1=2.16$, $x_2=1.46$, 1.416, 1.41421, $\ldots$ ......

If $x_0=5$, then $\cdots$

Proposition (convergence of Newton's Method)

To solve $g(x)=0$:

Assume that g is C^2. x^* solves $g(x^*)=0$, $g'(x^*)\neq 0$

Then if x_0 is close enough to x^*, the sequence $x_{k+1}=x_k-\frac{g(x_k)}{g'(x_k)}$ converges to x^*

Proof: $x_{k+1}-x^*=x_k-x^*-\frac{g(x_k)}{g'(x_k)}=x_k-x^*-\frac{g(x_k)-g(x^*)}{g'(x_k)}$

understand this

$$=\frac{g(x^*)-g(x_k)-g'(x_k)(x^*-x_k)}{g'(x_k)}$$

$$\overset{MVT}{=}\frac{(x^*-x_k)[g'(s)-g'(x_k)]}{g'(x_k)}$$ s between x^* and x_k

$$\overset{MVT}{=}\frac{(x^*-x_k)(s-x_k)\cdot g''(t)}{g'(x_k)}$$ t between x_k and s

Also since g is C^2 and $g'(x^*)\neq 0$, $\exists$ a neighbourhood of x^* where $|g'(x)|\geq k_1$, $|g''(x)|\leq k_2$

Then $|x_{k+1}-x_k|\leq\frac{k_2}{k_1}|x_k-x^*||s-x_k|\leq\frac{k_2}{k_1}(x_k-x^*)^2$

In particular, if $\frac{k_2}{k_1}|x_k-x^*|<\frac{1}{2}$

then $|x_{k+1}-x^*|\leq\frac{1}{2}|x_k-x^*|$.

this implies $x_k\to x^*$ as $k\to\infty$

Ex 2. Approximate f near x_k by (General minimization plan)

$$q(x)=f(x_k)+(x-x_k)f'(x_k)+\frac{1}{2}(x-x_k)^2\left(\frac{f'(x_k)-f'(x_{k-1})}{x_k-x_{k-1}}\right)$$

advantage: don't need to know 2nd order derivative. (but need f' at 2 pts)

Why is q reasonable?

$q=f$ at x_k

$q'=f'$ at x_k, x_{k-1}

Indeed $q'(x)=f'(x_k)-(x-x_k)\frac{f'(x_k)-f'(x_{k-1})}{x_k-x_{k-1}}$

Note also: $\frac{f'(x_k)-f'(x_{k-1})}{x_k-x_{k-1}}\approx f''(x_k)$ if x_k-x_{k-1} is small

The equation $g'(x_{k+1})=0$ gives

$$\boxed{x_{k+1} = x_k - \frac{f'(x_k)(x_k - x_{k-1})}{f'(x_k) - f'(x_{k-1})}}$$

This gives new algorithm (need to guess both x_0, x_1)

Similar to Newton's method, can prove it converges if x_0, x_1 are close enough to min x^*, where $f(x^*) \neq 0$.

Ex3: want more pts & fewer derivatives
choose x_0, x_1, x_2 let $g(x)$ = quadratic function that equal to f at x_{k-2}, x_{k-1}, x_k and x_{k+1} = pt where $g'(x)=0$

$$g(x) = f(x_{k-2})\frac{(x-x_k)(x-x_{k-1})}{(x_{k-2}-x_k)(x_{k-2}-x_{k-1})} + f(x_{k-1})\frac{(x-x_k)(x-x_{k-2})}{(x_{k-1}-x_k)(x_{k-1}-x_{k-2})} + \text{similar stuff}$$

We could in principle deduce a formula for x_{k+1} in terms of $x_k, x_{k-1}, x_{k-2}, f(x_k), f(x_{k-1}), f(x_{k-2})$
(see page 225, equation (21))
Formula is complicated.

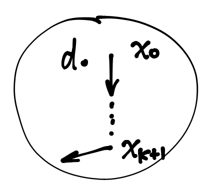
However, if f hard to differentiate.

Ex3. may be easy to compute, can also be used to define method that is guaran-teed to converge (even for bad initial guess).

Function of several variables
Assume we have a "subroutine" that can minimize functions of a single variable.

General strategy (line search algorithm)
1. start with pt x_k
2. choose a direction d_k.
3. Choose x_{k+1} to be a min of f in the half-line

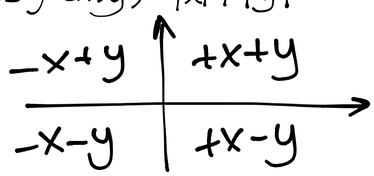
$$\{ y = x_k + s \cdot d_k : s > 0 \}$$



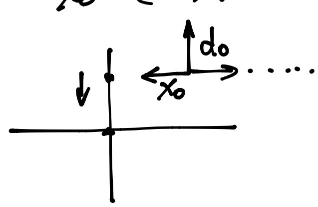
To design a good algorithm, need a good way of choosing direction.

Ex. for $f: E^2 \to \mathbb{R}$, try above procedure with $d_0 = (1,0)$, $d_1 = (0,1)$, $d_2 = (-1,0)$, $d_3 = (0,-1)$

Sps $f(x,y) = |x| + |y|$

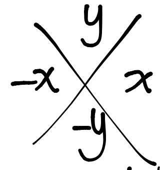


$x_0 = (a,b)$



$x_1 = x_0$
$x_2 = x_1 = x_0$
$x_3 = (0,b)$.
$x_4 = (0,0)$

Now $f(x,y)=\max\{|x|,|y|\}$



x_2 = some point (a,b)
$b<b_1\leq a$ as $x_1=x_0$

$x_0=(a,b)$ $(a>0, b<0)$
$x_1=x_0$, $x_2=x_1$

The point is if don't make a good choice among many minima, algorithm will fail.

Def'n: Sps X and Y are sets. A point-to-set mapping from X to Y is a function A, whose domain is X and range is {subsets of Y}, for $\forall x$, $A(x)$ is a subset of Y.

Ex. $X=(x,d)\in E^n\times E^n$
$Y=E^n$
$A(x)=\{\text{min of } f \text{ along half-line } \vec{x}+s\vec{d}, s\geq 0\}$

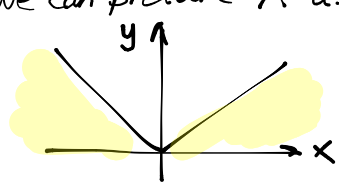
An algorithm is a pt-to-set mapping $A: X\to X$.
Then I can define X_{i+1} a point in $A(X_i)$. I can iterate, hope to solve problems

Def'n: A pt-to-set mapping is closed at X if whenever $X_k\to x$ in X
$y_k\in A(X_k)$, $y_k\to y$, we have $y\in A(x)$

Remark: If A is an ordinary function and it is continuous, then it is closed.

Ex. $X=Y=(-\infty,\infty)$
① Sps, $A(x)=[0,|x|]=\{y: 0\leq y\leq |x|\}$
Claim: A is closed

We can picture A as follow:

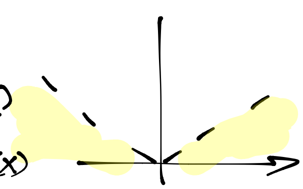


To see that A is closed, sps (X_k) is convergent to X

$y_k\in A(X_k)$, $y_k\to y$

need to check: $y\in A[X]$, $0<y\leq|x|$

$B(x)=[0,|x|)=\{y: 0\leq y<|x|\}$. not closed.
let $X_k=1+\frac{1}{k}\to x=1$
$y_k=1+\frac{1}{2k}\in[0,X_k)$
Then $y_k\to y=1$
but $y\notin B(x)$



A is closed if it is closed at all X.

Convergence Thm:

Goal: minimize f (continuous function). Let $\Gamma = \{x : f(x) = \min \text{ of } f\}$

Assume A is a pt-to-set mapping $X \to X$ s.t.

① : A is closed at every x outside Γ

② if $x \notin \Gamma$, $y \in A(x)$, then $f(y) < f(x)$

③ if $x \in \Gamma$, then $A(x) = \Gamma$

Choose $x_0 \in X$, and $x_{k+1} \in A(x_k)$, $k \geq 0$

If the sequence (x_k) is contained in a compact set $S \subset X$, then $\forall$ limit of $\forall$ convergent subsequence minimize f.

Corollary: For A as above, if $x_k \to x$ in X, then x is a minimizer.

Remark: We always assume $X \subseteq$ some Euclidean space.
Then S is compact $\iff$ $\forall$ sequence in S has a convergent subsequence.
$\iff$ closed & bounded.

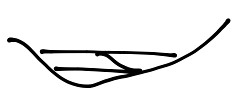
In practice, the corollary is often good enough.

Main points:
① Algorithms should be closed
② algorithm should decrease the value of f

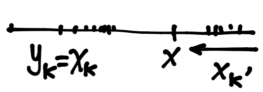
Proof of the theorem:
① $(x_k) \in S$ compact, $\exists$ a convergent subsequence $(x_{k'})$ s.t. $x_{k'} \to x$
Proof by contradiction:
Sps $\exists x \notin \Gamma$ (then hypothesis $\to$ A closed at x_0)
f: cont. and $x_{k'} \to x \Rightarrow f(x_{k'}) \to f(x)$

Claim: the whole sequence $f(x_k) \to f(x)$ ✪
True b/c $f(x_k)$ is a decreasing sequence
and for a decreasing sequence, if a subsequence converges, then the whole sequence converges. (intuitively clear)

Now consider the subsequence $x_{k'+1} \in A(x_{k'})$
Again using the compactness, there is a farther subsequence $(x_{k''})$ s.t.
$x_{k''+1} \in A(x_{k''}) \to y$
and $x_{k''} \to x$ (still)
Then since A is closed at x, y belongs to $A(x)$
Then $f(y) < f(x)$ but impossible by ✪



y sequence can't end up larger than x_k sequence



Preview: To minimize a C^1 function, we'll argue as follow:

Given x_k

if $\nabla f(x_k) = 0$, then x_k = mimima, done.

if $\nabla f(x_k) \neq 0$, let $d_k = -\nabla f(x_k)$ and let x_{k+1} minimizes f in $\{\vec{x} + s\vec{d} : s > 0\}$

Method of steepest

Recall: $\nabla f(x)$ point in direction of fast test b/c $f(\vec{x} + h\vec{d}) = f(\vec{x}) + h \nabla f(\vec{x}) d + O(h)$

max: if d is parallel to gradient

so $-\nabla f(x)$ = direction of steepest descent