

Numerical Optimization in Robotics

■ Lecture 1: Preliminaries



主讲人 Zhepei
Wang

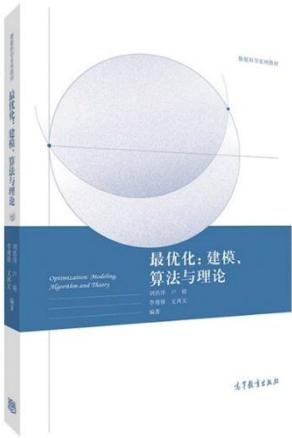
Ph.D. in Robotics
Zhejiang University



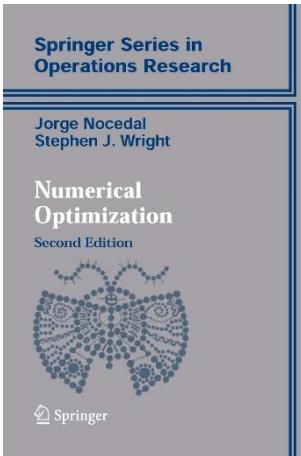
Introduction



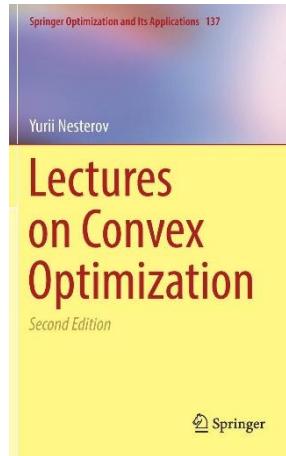
Recommended Reading List



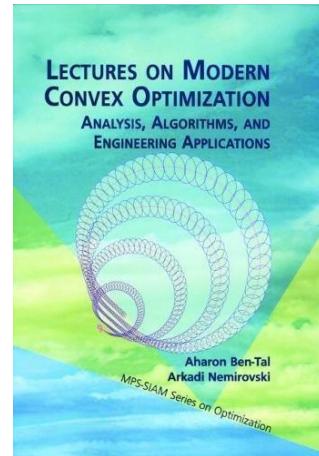
Friendly to Chinese students



Numerical implementation



Convex Optimization Theory



Conic Programming



Mathematical Optimization

optimization problem

$$\begin{aligned} & \min f(x) \\ \text{s.t. } & g(x) \leq 0 \\ & h(x) = 0 \end{aligned}$$

1. $f(x)$ is lower bounded(值域有界)

2. $f(x)$ has bounded level set(定义域有界)

$x = (x_1, \dots, x_n) \in \mathbb{R}^n$: optimization variables

$f: \mathbb{R}^n \mapsto \mathbb{R}$: objective function

$g: \mathbb{R}^n \mapsto \mathbb{R}^{m_g}$: inequality constraint functions

$h: \mathbb{R}^n \mapsto \mathbb{R}^{m_h}$: equality constraint functions

optimal solution x^* has smallest value f among all vectors that satisfy the constraints.



Mathematical Optimization

optimization problem

$$\begin{aligned} & \min f(x) \\ \text{s.t. } & g(x) \leq 0 \\ & h(x) = 0 \end{aligned}$$

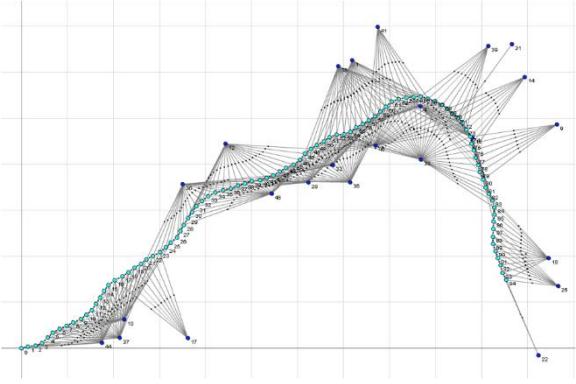
default assumptions:

1. The objective function is lower bounded.
2. The objective function has bounded sub-level sets.

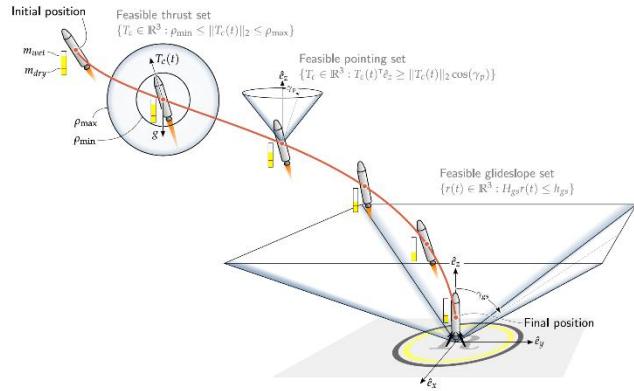


Optimization in Robotics

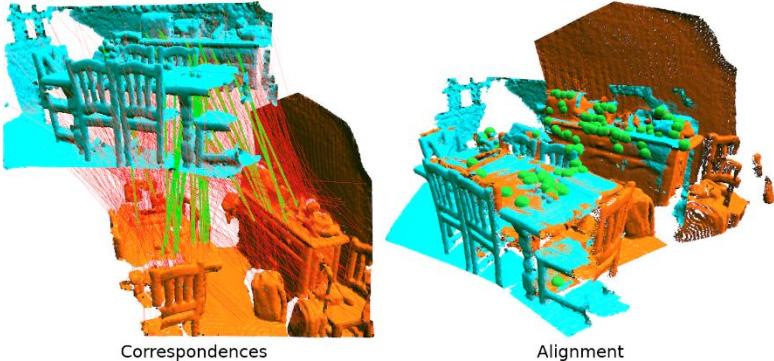
Smooth and Mapping: Nonlinear Least Squares



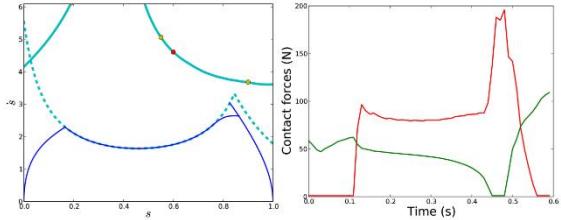
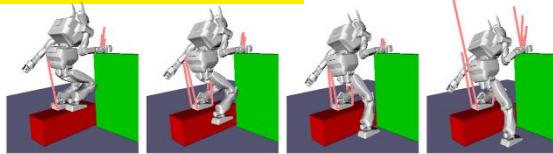
Trajectory Planning: Nonlinear Program



Registration: Semi-Definite Program



Time-Optimal Path Parameterization: Second-Order Conic Program





Course Topics

1st Course: Fundamentals of Numerical Optimization

1. Mathematical Programming and Robotics
2. Convexity in Nonconvex Optimization
3. Convex Sets and Convex Functions
4. Unconstrained Nonconvex Optimization
5. Line-Search Steepest Gradient Descent
6. Modified Damped Newton's Method

Nonconvex Problems

2nd Course: Unconstrained Optimization

1. Quasi-Newton Methods
2. BFGS Update
3. Strong/Weak Curvature Conditions
4. Li-Fukushima Global Convergence Condition
5. Limited Memory BFGS Update
6. Lewis-Overton Line Search for Non-Smooth Functions
7. Linear Conjugate Gradient Descent
8. Newton-CG Method
9. Application: Smooth Navigation Path Generation

Nonconvex Problems



Course Topics

Nonconvex Problems

3rd Course: Constrained Optimization

1. Category and Complexity of Constrained Optimization
2. Low-Dimensional Linear Program: Seidel's LP
3. Low-Dimensional Strictly Convex Quadratic Program
4. Sequential Unconstrained Minimization Technique (SUMT)
5. SUMT: Penalty Method and Barrier Method
6. SUMT: Lagrangian Relaxation and Uzawa's Method
7. Karush-Kuhn-Tucker Conditions
8. PHR Augmented Lagrangian Method
9. Application 1: Control Allocation Problem
10. Application 2: Collision Distance Computation
11. Application 3: Nonlinear Model Predictive Control

4th Course: Optimization for Symmetric Cones

1. Cones and Symmetric Cones
2. Symmetric Cones and Euclidean Jordan Algebra
3. Representability of Cones
4. Augmented Lagrangian for Symmetric Cones
5. Semi-Smooth Newton Method
6. Application: Time-Optimal Path Parameterization

Convex Problems

5th Course: Techniques in Formulating and Solving Problems

1. Smoothing Techniques
2. Degrees of Freedom and Adjoint Method
3. Category and Features of Linear Solvers
4. Project: Enabling Safe Navigation in Dense-Obstacle Environments

Nonconvex Problems

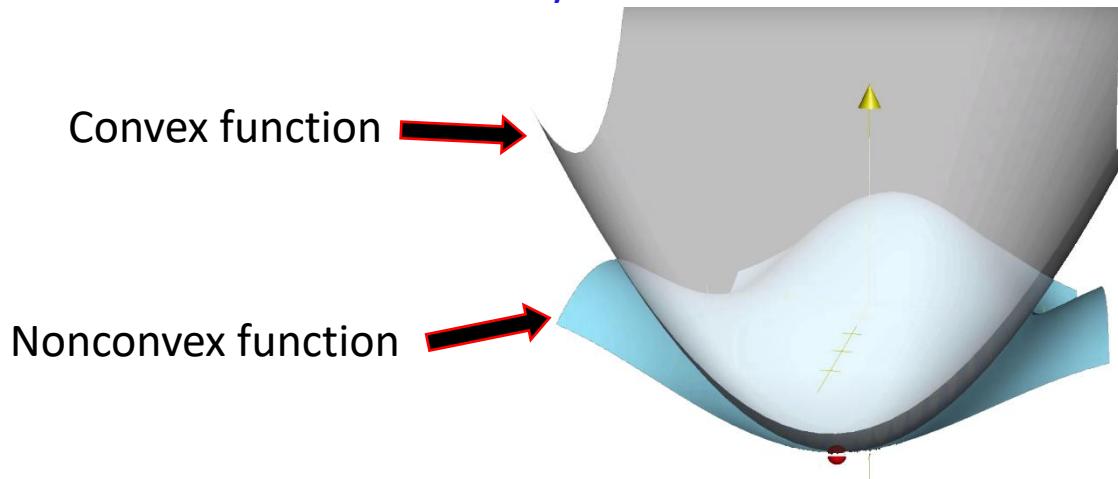
非凸优化中的凸性质

Convexity in Nonconvex Optimization



Convexity in Nonconvex Optimization

Why do we need to know convexity?



- Optimization of convex functions over convex sets is well studied.
- Optimization algs utilize properties of convex fun/set to analyze the convergence.
- Some important problems have convex formulation/relaxiation.
- Many **nonconvex functions** are **locally convex** near the interested local minima.
- In practice, **local minima** of nonconvex functions can be good enough. 连续，确定性的优化

Convex Sets



Convex Sets

凸集的概念

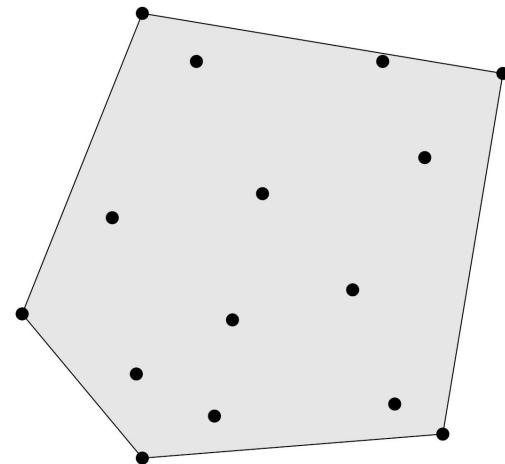
Def: A set is convex if every line between its two points stays in the set?

$$\theta x_1 + (1 - \theta)x_2, \quad 0 \leq \theta \leq 1$$

More General:
All **convex combinations**
lie in set.

$$\begin{aligned}\sum \theta_i x_i &= \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \\ \sum \theta_i &= \theta_1 + \theta_2 + \theta_3 = 1, \quad \theta_i \geq 0 \forall i\end{aligned}$$

凸组合

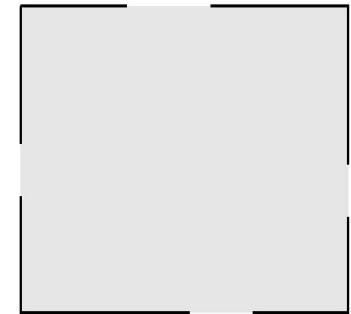
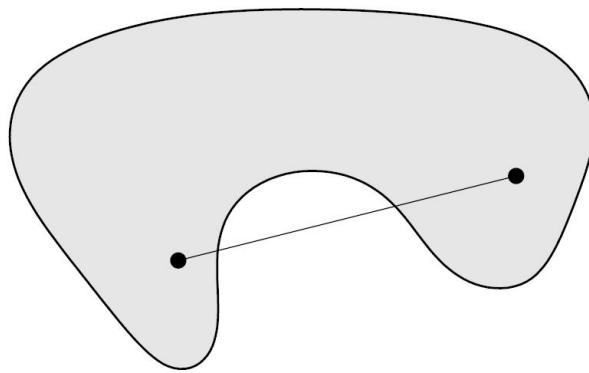
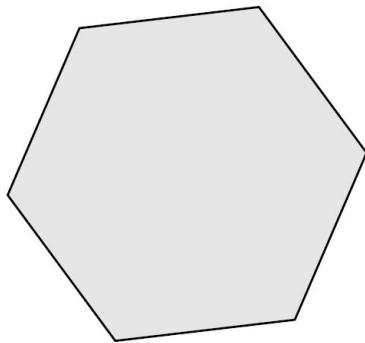


These definitions are the same!



Convex Sets

Is it convex?



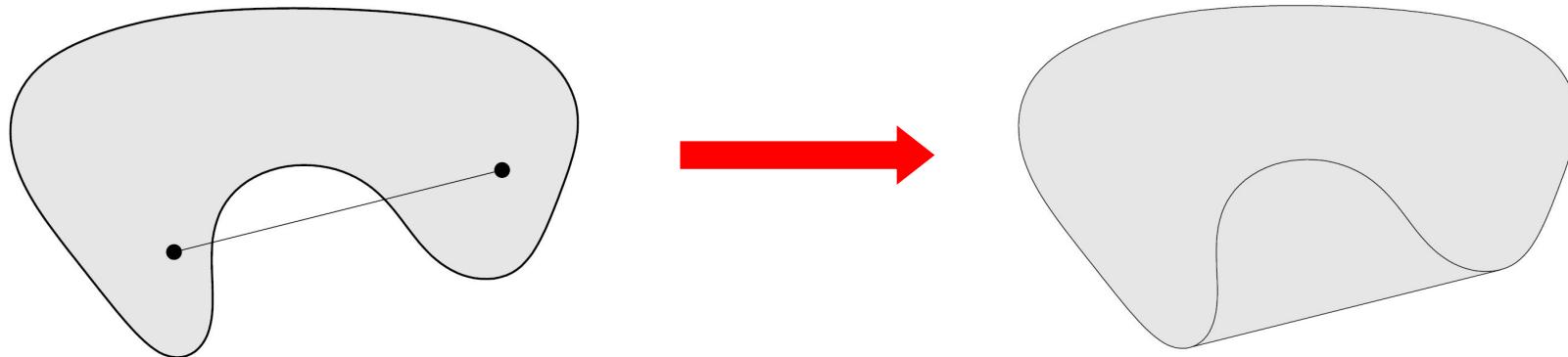
What if the square was round?





Convex Sets

Convex hull: All convex combinations of points in a set



It's always the **smallest** convex superset



Convex Sets

Important examples: Are these convex?

Hyperplane

$$\{x \mid a^T x = b\}$$

Half-space

$$\{x \mid a^T x \geq b\}$$

Sphere

$$\{x \mid \|x - x_0\| = b\}$$

Ball

$$\{x \mid \|x - x_0\| \leq b\}$$

Polynomials

$$\left\{ f \mid f = \sum_i a_i x^i \right\}$$



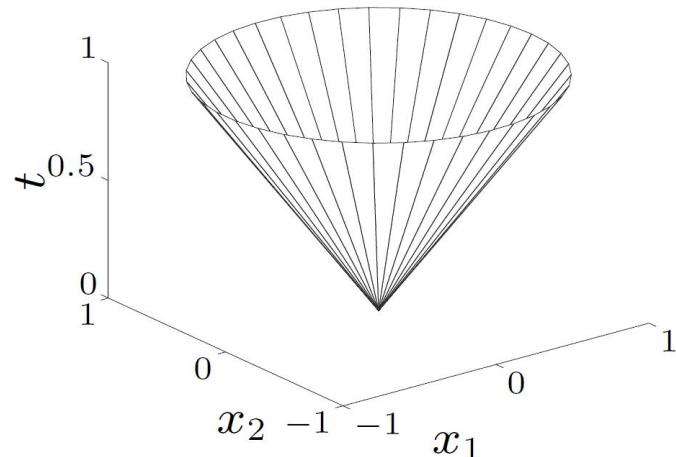
Convex Sets

Cones (not necessarily convex)

$$x \in C \implies ax \in C, \quad \forall a \geq 0$$

Second-order cone:

$$C_2 = \{(x, t) \mid \|x\| \leq t\} \in \mathbb{R}^{n+1}$$





Convex Sets

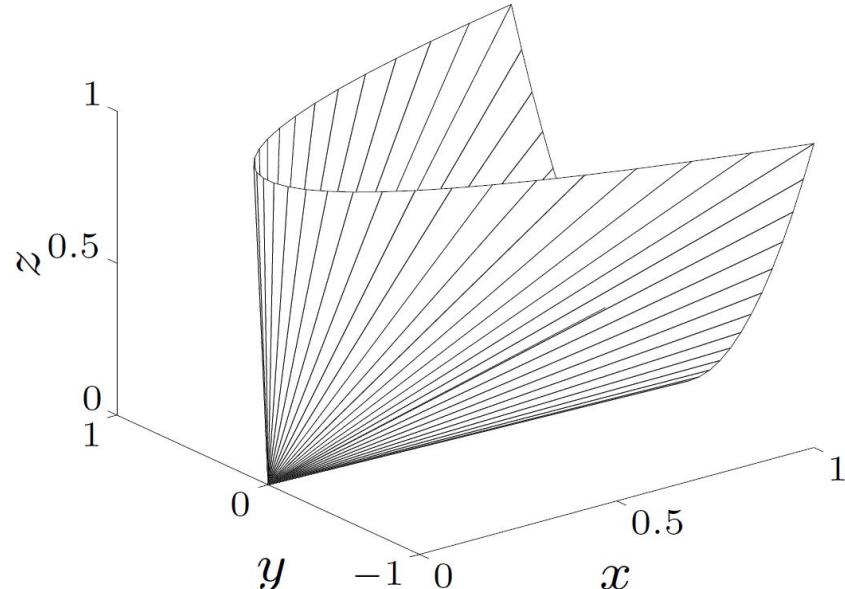
Semi-definite cone

$$\mathcal{S}_+^n = \{A \in \mathbb{R}^{n \times n} \mid A = A^T, A \succeq 0\}$$

Set of PSD matrices?

$$\begin{bmatrix} x & y \\ y & z \end{bmatrix} \in \mathcal{S}_+^2$$

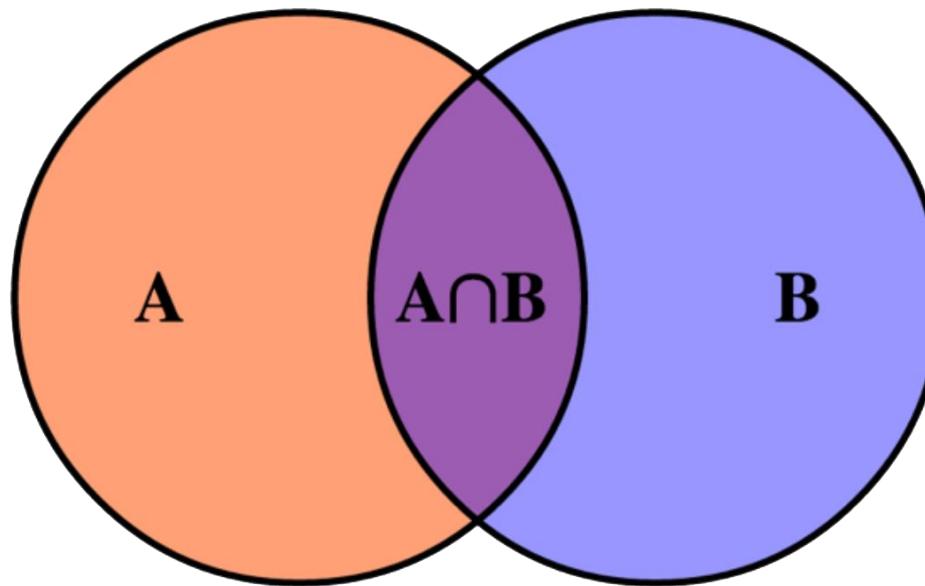
Why is this a cone?





Convex Sets

Allowed operations



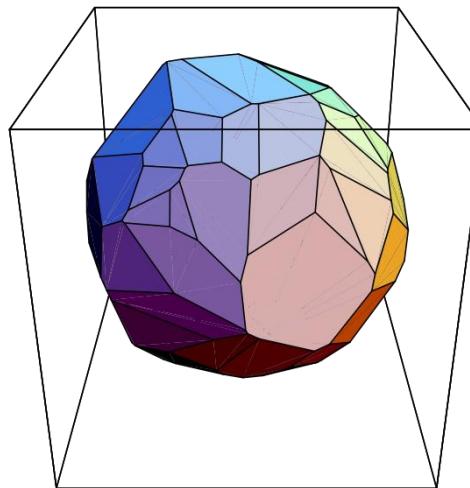
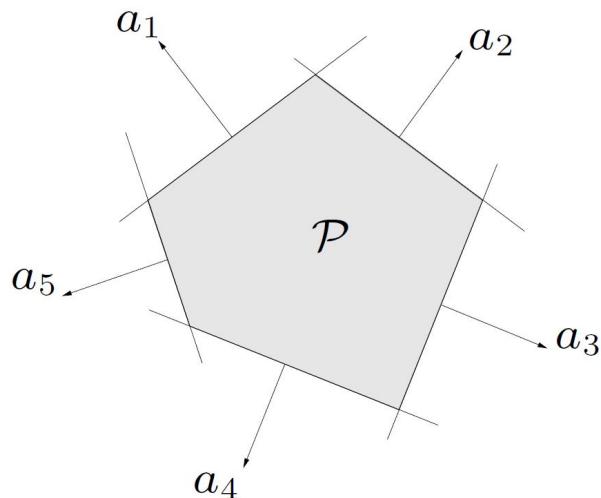
The **intersection** of convex sets is **convex**



Convex Sets

Polytope

$$\{x : Ax \leq b\}$$



Is it convex? Why?



Convex Sets

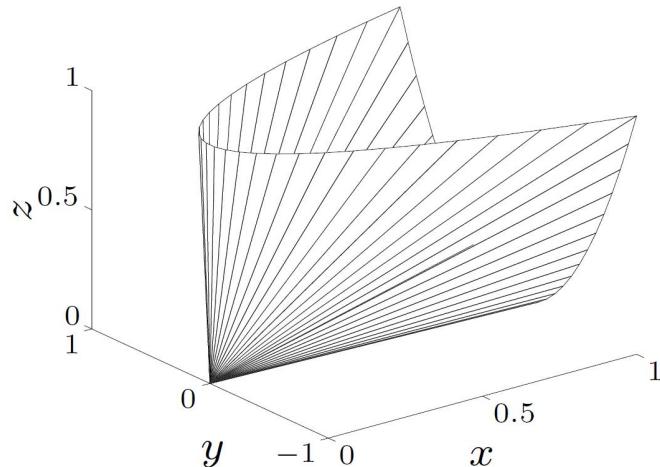
Semi-definite cone

$$x^T A x \geq 0, \forall x$$

$$\mathcal{S}_+^n = \bigcap_{x \in \mathbb{R}^n} \{A \mid x^T A x \geq 0\}$$



Convex?





Convex Sets

Other allowed operations

Set sum

$$A + B = \{x + y \mid x \in A, y \in B\}$$

Set product

$$A \times B = \{(x, y) \mid x \in A, y \in B\}$$

What about union?

$$A \cup B$$

函数的高阶信息

High-Order Info of Functions



High-Order Info of Functions

Function $f(x) = f(x_1, x_2, x_3)$

Gradient $\nabla f(x) = \begin{pmatrix} \partial_1 f(x) \\ \partial_2 f(x) \\ \partial_3 f(x) \end{pmatrix}$

Hessian $\nabla^2 f(x) = \begin{pmatrix} \partial_1^2 f(x) & \partial_1 \partial_2 f(x) & \partial_1 \partial_3 f(x) \\ \partial_2 \partial_1 f(x) & \partial_2^2 f(x) & \partial_2 \partial_3 f(x) \\ \partial_3 \partial_1 f(x) & \partial_3 \partial_2 f(x) & \partial_3^2 f(x) \end{pmatrix}$

Symmetric for
smooth func

泰勒展开

Linear approx

Approx at zero $f(x) = f(0) + x^T \nabla f(0) + \frac{1}{2} x^T \nabla^2 f(0) x + O(\|x - x_0\|^3)$

Quadratic approx



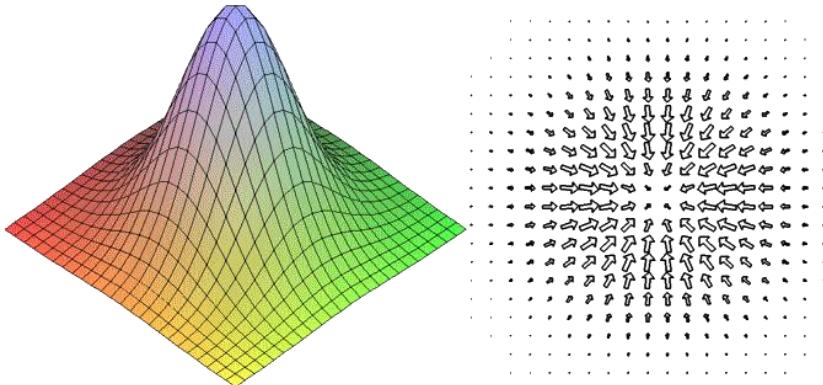
High-Order Info of Functions

Gradient

Let $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, then vector, which contains all first order partial derivatives:

$$\nabla f(x) = \frac{df}{dx} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

named gradient of $f(x)$. This vector indicates the direction of steepest ascent. Thus, vector $-\nabla f(x)$ means the direction of the steepest descent of the function in the point. Moreover, the gradient vector is always orthogonal to the contour line in the point.





High-Order Info of Functions

Hessian

Let $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$, then matrix, containing all the second order partial derivatives:

$$f''(x) = \frac{d(\nabla f)}{dx^T} = \frac{d(\nabla f^T)}{dx} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix}$$

But actually, Hessian could be a tensor in such a way: $(f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m)$ is just 3d tensor, every slice is just hessian of corresponding scalar function $(H(f_1(x)), H(f_2(x)), \dots, H(f_m(x)))$

Jacobian

The extension of the gradient of multidimensional $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}^m$:

$$f'(x) = \frac{df}{dx^T} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{pmatrix}$$



High-Order Info of Functions

Jacobian of $g(x)$:

$$\begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots & \frac{\partial g_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1} & \frac{\partial g_m}{\partial x_2} & \cdots & \frac{\partial g_m}{\partial x_n} \end{pmatrix}$$

Substitution:

$$g(x) := \nabla f(x), \quad g_i(x) = \frac{\partial f}{\partial x_i}$$

The Hessian of $f(x)$:

$$\begin{pmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix}$$



High-Order Info of Functions

Matrix and vector derivatives

Table

$$f(x) : X \rightarrow Y; \quad \frac{\partial f(x)}{\partial x} \in G$$

X	Y	G	Name
\mathbb{R}	\mathbb{R}	\mathbb{R}	$f'(x)$ (derivative)
\mathbb{R}^n	\mathbb{R}	\mathbb{R}^n	$\frac{\partial f}{\partial x_i}$ (gradient)
\mathbb{R}^n	\mathbb{R}^m	$\mathbb{R}^{n \times m}$	$\frac{\partial f_i}{\partial x_j}$ (jacobian)
$\mathbb{R}^{m \times n}$	\mathbb{R}	$\mathbb{R}^{m \times n}$	$\frac{\partial f}{\partial x_{ij}}$

The notation of differentials could be useful here

$$dA = 0$$

$$d(\alpha X) = \alpha(dX)$$

$$d(AXB) = A(dX)B$$

$$d(X + Y) = dX + dY$$

$$d(X^\top) = (dX)^\top$$

$$d(XY) = (dX)Y + X(dY)$$

$$d\langle X, Y \rangle = \langle dX, Y \rangle + \langle X, dY \rangle$$

$$d\left(\frac{X}{\phi}\right) = \frac{\phi dX - (d\phi)X}{\phi^2}$$

$$d \operatorname{tr} X = I$$

$$df(g(x)) = \frac{df}{dg} \cdot dg(x)$$



High-Order Info of Functions

https://en.wikipedia.org/wiki/Matrix_calculus

Matrix calculus - Wikipedia

en.wikipedia.org/wiki/Matrix_calculus

Vector-by-vector identities [edit]

This is presented first because all of the operations that apply to vector-by-vector differentiation apply directly to vector-by-scalar or scalar-by-vector differentiation simply by reducing the appropriate vector in the numerator or denominator to a scalar.

Identities: vector-by-vector $\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$

Condition	Expression	Numerator layout, i.e. by \mathbf{y} and \mathbf{x}^T	Denominator layout, i.e. by \mathbf{y}^T and \mathbf{x}
\mathbf{a} is not a function of \mathbf{x}	$\frac{\partial \mathbf{a}}{\partial \mathbf{x}} =$	$\mathbf{0}$	
	$\frac{\partial \mathbf{x}}{\partial \mathbf{x}} =$	\mathbf{I}	
\mathbf{A} is not a function of \mathbf{x}	$\frac{\partial \mathbf{A}\mathbf{x}}{\partial \mathbf{x}} =$	\mathbf{A}	\mathbf{A}^T
\mathbf{A} is not a function of \mathbf{x}	$\frac{\partial \mathbf{x}^T \mathbf{A}}{\partial \mathbf{x}} =$	\mathbf{A}^T	\mathbf{A}
\mathbf{a} is not a function of \mathbf{x} , $\mathbf{u} = \mathbf{u}(\mathbf{x})$	$\frac{\partial \mathbf{a}\mathbf{u}}{\partial \mathbf{x}} =$	$\mathbf{a} \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	
$v = v(\mathbf{x})$, \mathbf{a} is not a function of \mathbf{x}	$\frac{\partial \mathbf{a}v}{\partial \mathbf{x}} =$	$\mathbf{a} \frac{\partial v}{\partial \mathbf{x}}$	$\frac{\partial v}{\partial \mathbf{x}} \mathbf{a}^T$
$v = v(\mathbf{x})$, $\mathbf{u} = \mathbf{u}(\mathbf{x})$	$\frac{\partial v\mathbf{u}}{\partial \mathbf{x}} =$	$v \frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \mathbf{u} \frac{\partial v}{\partial \mathbf{x}}$	$v \frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial v}{\partial \mathbf{x}} \mathbf{u}^T$
\mathbf{A} is not a function of \mathbf{x} , $\mathbf{u} = \mathbf{u}(\mathbf{x})$	$\frac{\partial \mathbf{A}\mathbf{u}}{\partial \mathbf{x}} =$	$\mathbf{A} \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \mathbf{A}^T$
$\mathbf{u} = \mathbf{u}(\mathbf{x})$, $v = v(\mathbf{x})$	$\frac{\partial(\mathbf{u} + \mathbf{v})}{\partial \mathbf{x}} =$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{v}}{\partial \mathbf{x}}$	
$\mathbf{u} = \mathbf{u}(\mathbf{x})$	$\frac{\partial g(\mathbf{u})}{\partial \mathbf{x}} =$	$\frac{\partial g(\mathbf{u})}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \frac{\partial g(\mathbf{u})}{\partial \mathbf{u}}$
$\mathbf{u} = \mathbf{u}(\mathbf{x})$	$\frac{\partial f(g(\mathbf{u}))}{\partial \mathbf{x}} =$	$\frac{\partial f(g)}{\partial g} \frac{\partial g(\mathbf{u})}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{u}}{\partial \mathbf{x}} \frac{\partial g(\mathbf{u})}{\partial \mathbf{u}} \frac{\partial f(g)}{\partial g}$



High-Order Info of Functions

Examples

Calculate $\nabla f(x)$, if $f(x) = \frac{1}{2}x^T Ax + b^T x + c$

$$f(x) = \frac{1}{2} \sum_{i,j=1}^n [x_i a_{ij} x_j] + \sum_{i=1}^n b_i x_i + c_i$$

$$\frac{\partial f(x)}{\partial x_i} = \frac{1}{2} \sum_{i,j=1}^n (a_{ij} + a_{ji}) x_i + b_i \rightarrow \nabla f(x) = \frac{1}{2} (A + A^T)x + b$$

Calculate $\nabla f(X)$, if $f(X) = \text{tr } AX$

$$g(Y) = \text{tr } Y$$

$$h(X) = AX$$

$$\frac{df}{dX} = \frac{dg(h(X))}{dX} = \left(\frac{dh}{dX} \right)^T \frac{dg}{dh} = (A)^T I = A^T$$

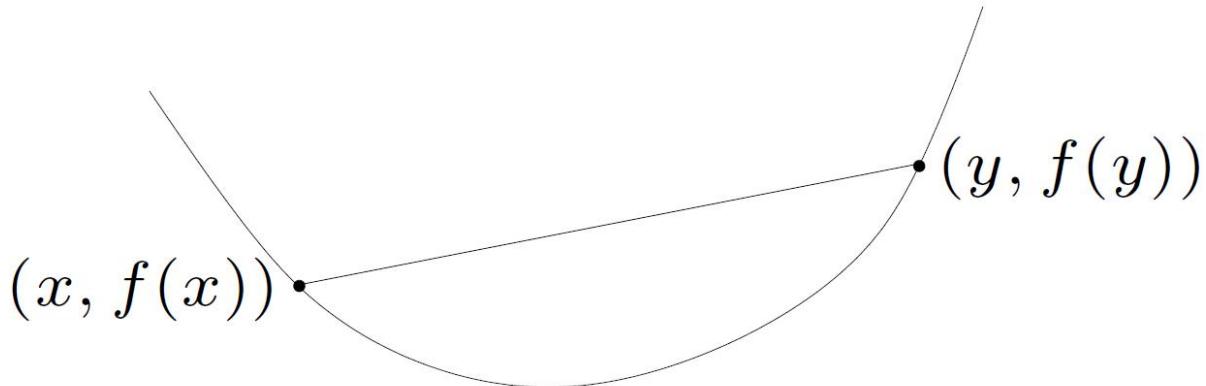
Convex Functions



Convex Functions

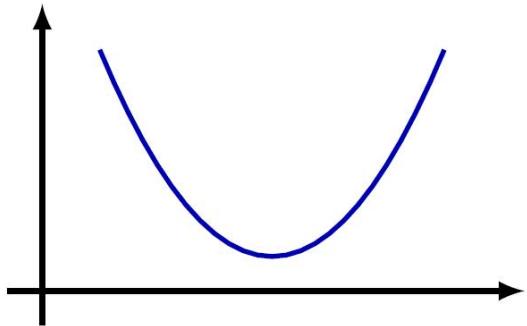
Jensen's inequality

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$$

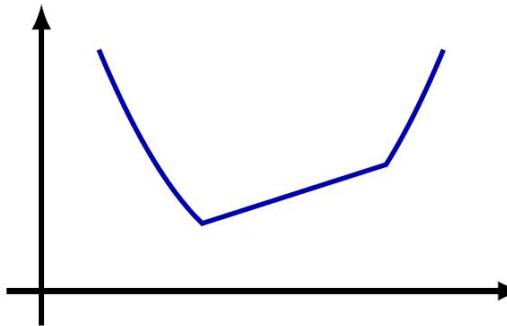




Convex Functions

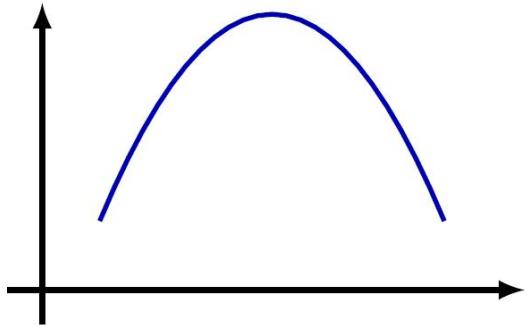


strictly convex function

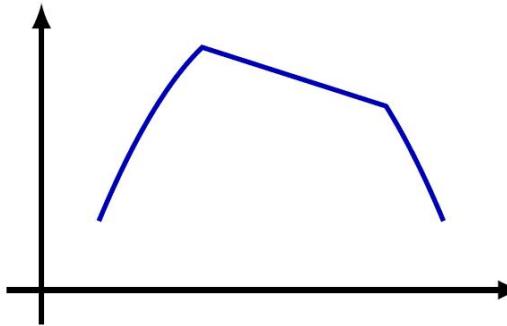


convex function

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$$



strictly concave function



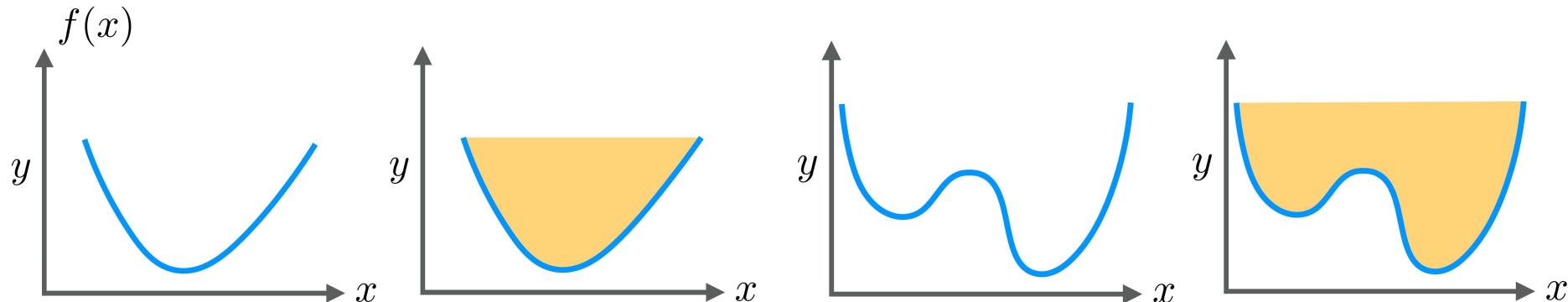
concave function



Convex Functions

Epigraph (上方图)

$$\text{epi}(f) = \{(x, y) \mid f(x) \leq y\}$$



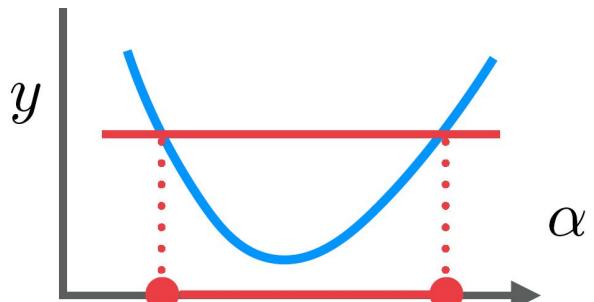
convex function = convex epigraph



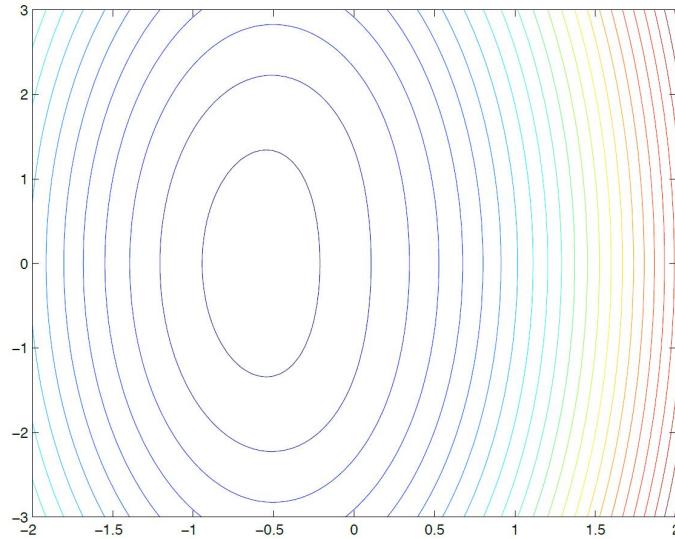
Convex Functions

Why do we care about convex functions?

Convex functions have **convex sub-level sets**



$$f_\alpha = \{x \mid f(x) \leq \alpha\}$$



convex contours

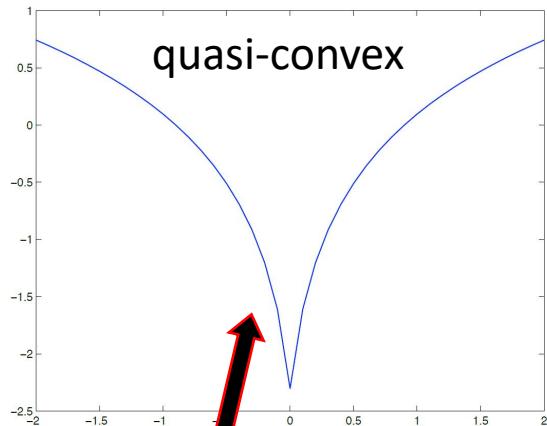


Convex Functions

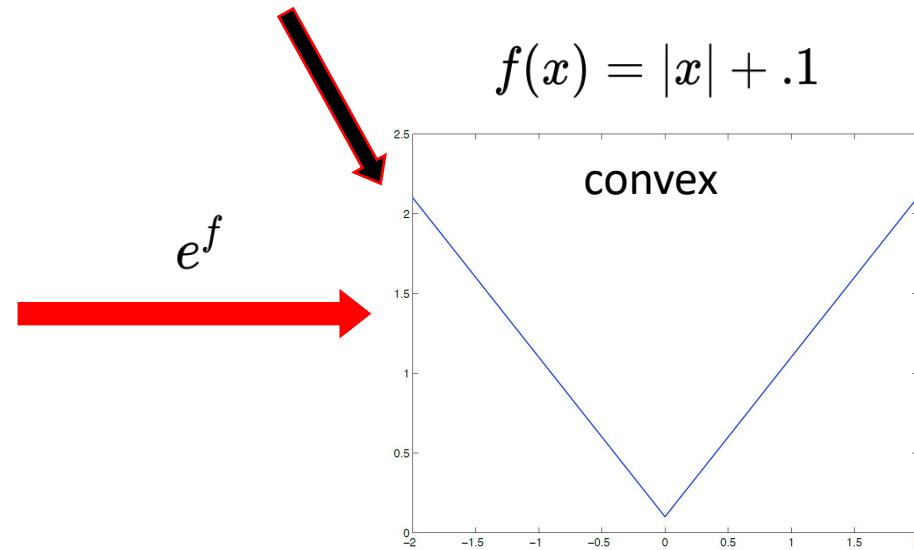
Why do we care about convex functions?

Convexity are somewhat easy to preserve

$$f(x) = \log(|x| + .1)$$



$$f(x) = |x| + .1$$



convex contours = quasi-convex > convex

Sum of quasi-convex functions is not necessary quasi-convex.



Convex Functions

Why do we care about convex functions?

Any minimizer is **global** (why)

Set of minima is **convex** (why)

Convex functions are **closed under many operations**.



Convex Functions

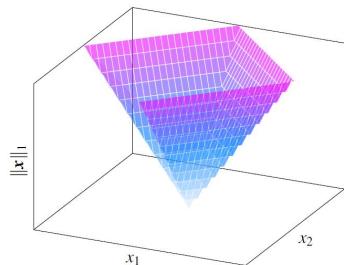
Positive weighted sum

$$g(x) = \sum_i f_i(x)$$

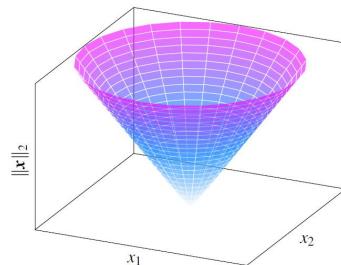
Examples:

$$f(x) = \|x\|^2 = \sum_i x_i^2$$

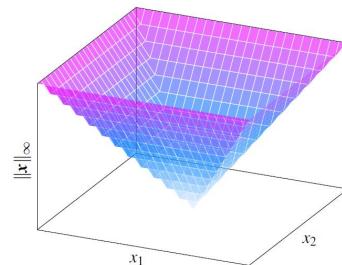
$$f(x) = x + x^2 + x^6 + |x|$$



1-norm



2-norm



∞ -norm

Sums of convex functions are convex



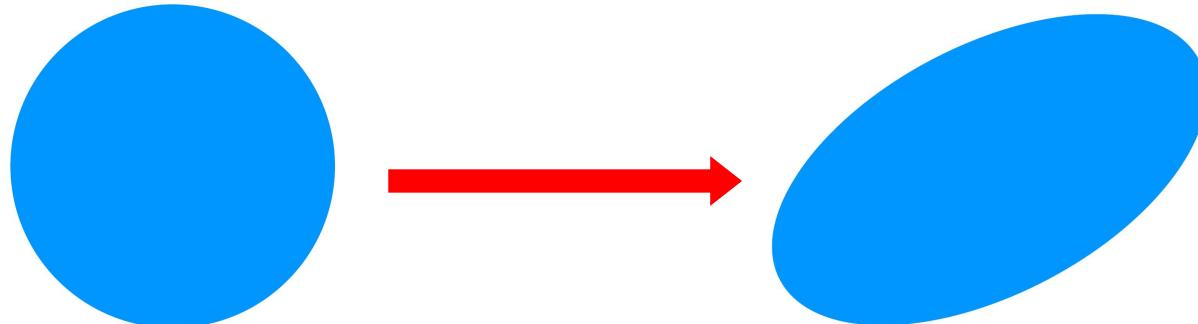
Convex Functions

Affine composition

Affine composition with convex function = convex

$$f(x) \xrightarrow{\hspace{1cm}} f(Ax + b)$$

Is the convexity of epigraphs preserved by these operations?





Convex Functions

Point-wise max preserves convexity

$$g(x) = \max_i f_i(x)$$

Absolute value $|x| = \max\{x, -x\}$

Infinity norm $\|x\|_\infty = \max_i |x_i|$

Max eigenvalue $\|A\|_2 = \max_v v^T A v$

Is the convexity of epigraphs preserved by these operations?



Convex Functions

Why are these convex?

Trace

$$f(X) = \text{trace}(A^T X)$$

Linear operator

Distance over set

$$f(x) = \max_{y \in C} \|x - y\|$$

Max over convex

Distance to convex set

$$f(x) = \min_{y \in C} \|x - y\|$$

Special case

Affine Norm

$$f(x) = \|b + \sum_i A_i x_i\|_2$$

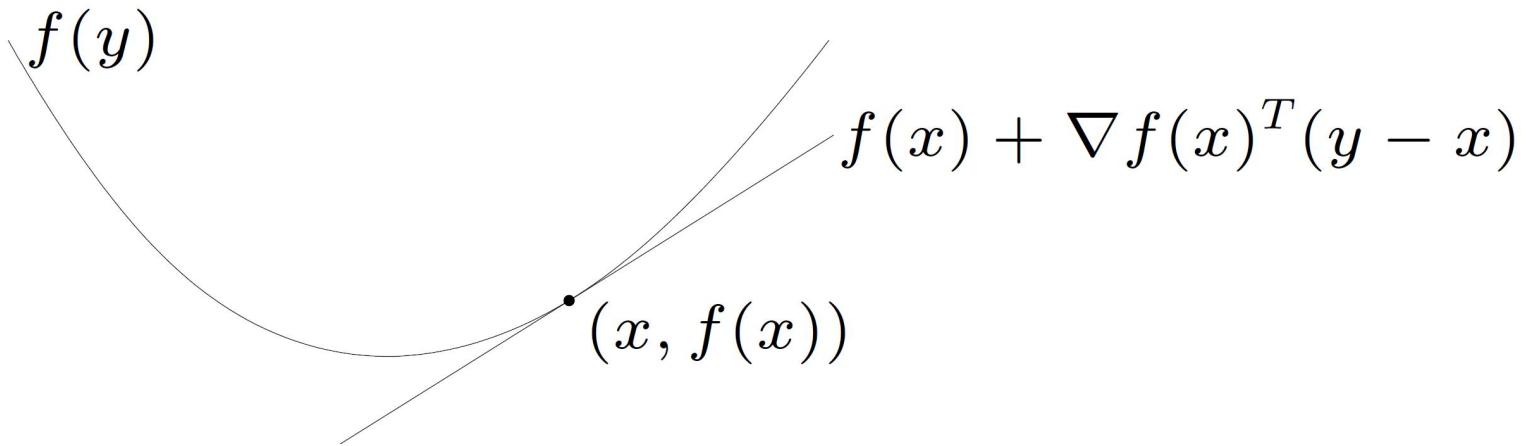
Affine comp

If $g(x,y)$ is convex, then minimizing for y preserves convexity



Convex Function Property

Convex functions lie **ABOVE** their linear approximation



$$f(y) \geq f(x) + \nabla f(x)^T(y - x) \quad \text{for all } x, y \in \text{dom } f$$



Convex Function Property

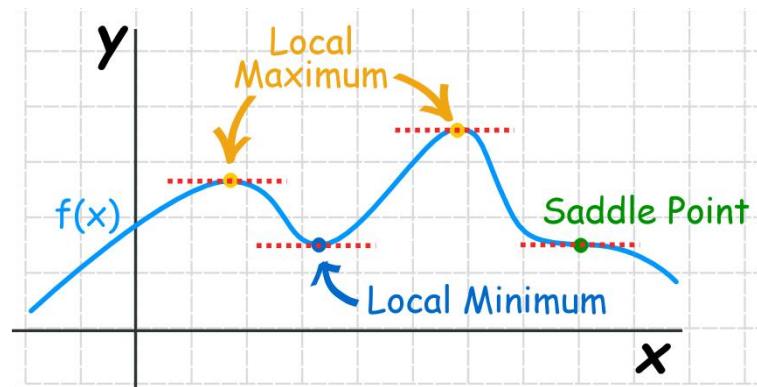
First-order conditions = optimality for convex funcs only

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

$$\nabla f(x)^T (y - x) = 0$$

$$f(y) \geq f(x)$$

This doesn't happen!



Any minima is a **global** minima

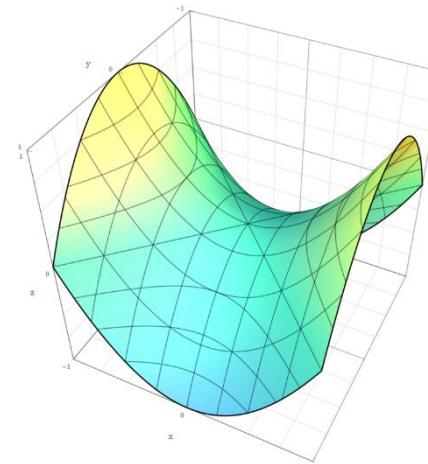
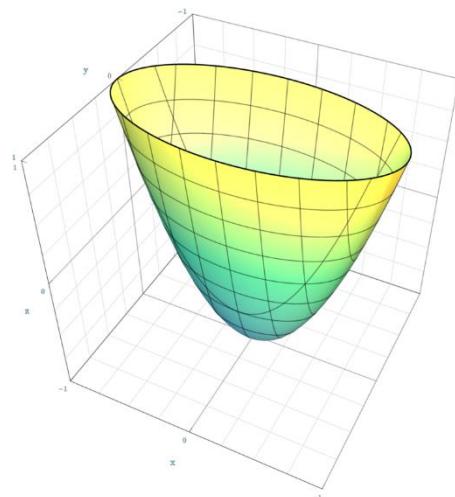


Convex Function Property

Second-order conditions

A smooth function is convex iff $\nabla^2 f(x) \succeq 0, \quad \forall x$ (semi-definite)

For non-convex functions, minima satisfy: $\nabla^2 f(x^*) \succeq 0$

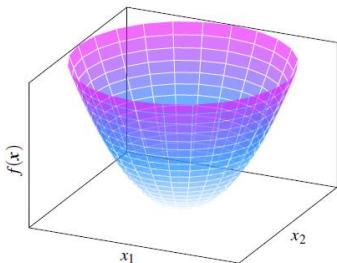


The Hessian is a good local model of a smooth function

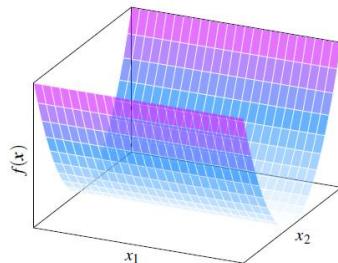


Convex Function Property

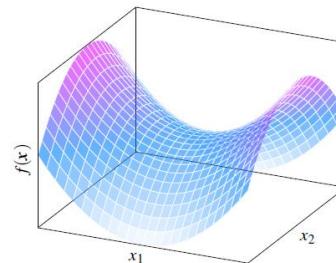
Quadratic functions $f(x) = x^T Qx \in \mathbb{R}^2$



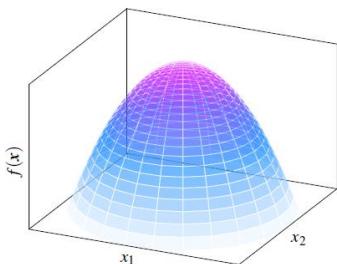
$Q = \text{diag}\{1, 1\}$
strictly convex



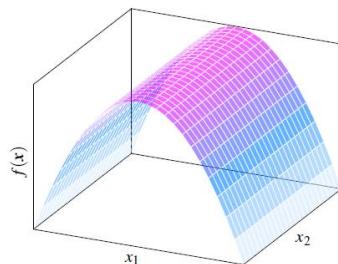
$Q = \text{diag}\{0, 1\}$
convex



$Q = \text{diag}\{1, -1\}$
neither convex nor concave



$Q = \text{diag}\{-1, -1\}$
strictly concave



$Q = \text{diag}\{-1, 0\}$
concave



Convex Function Property

Strong convexity

$$f(y) \geq \underbrace{f(x) + (y - x)^T \nabla f(x)}_{\text{holds for any convex func}} + \underbrace{\frac{m}{2} \|y - x\|^2}_{\text{min curvature}}$$

When Hessian exists

$$\begin{aligned} f(y) &\approx f(x) + (y - x)^T \nabla f(x) + \frac{1}{2} (y - x)^T \nabla^2 f(x) (y - x) \\ &\geq f(x) + (y - x)^T \nabla f(x) + \frac{\lambda_{\min}}{2} \|y - x\|^2 \end{aligned}$$

and so

$$\nabla^2 f(x) \succeq mI$$



Convex Function Property

The Lipschitz constant M of $\nabla f(x)$ satisfies

$$\|\nabla f(x) - \nabla f(y)\| \leq M\|y - x\|$$



$$f(y) \leq f(x) + (y - x)^T \nabla f(x) + \frac{M}{2} \|y - x\|^2$$

We can bound the objective error in terms of distance from minimizer

$$f(y) - f(x^*) \geq \frac{m}{2} \|y - x^*\|^2 \quad f(y) - f(x^*) \leq \frac{M}{2} \|y - x^*\|^2$$



Convex Function Property

Condition number

For any function

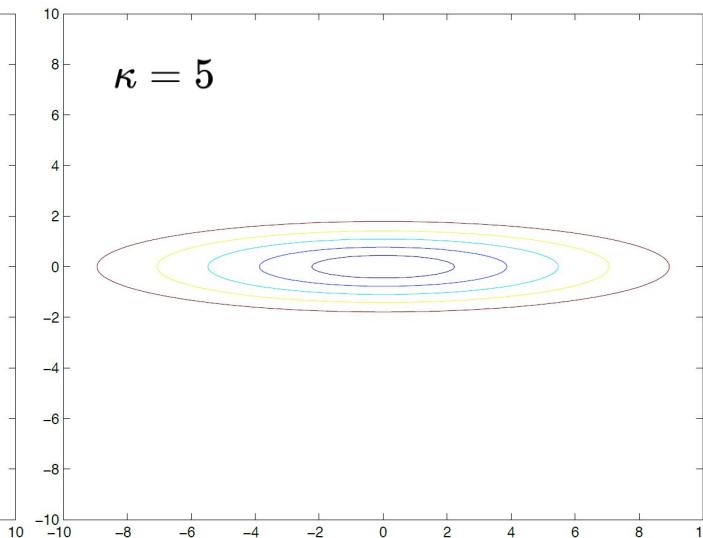
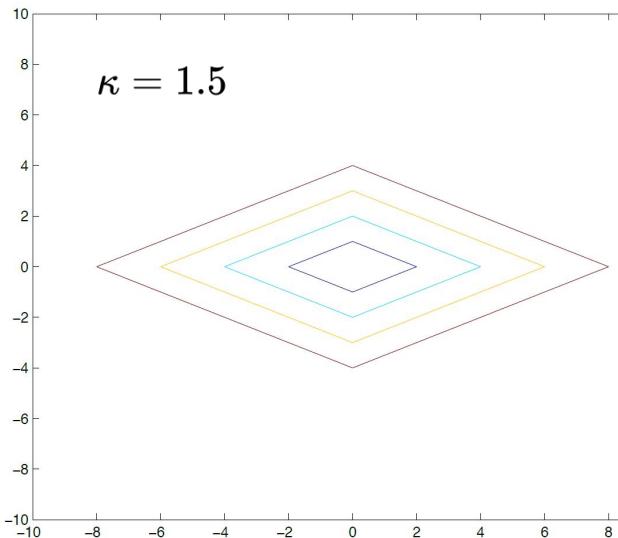
$$\kappa = \frac{\text{major axis}}{\text{minor axis}}$$

For smooth functions

$$\kappa \approx \text{cond}(\nabla^2 f(x))$$

For differentiable functions

$$\kappa = M/m$$

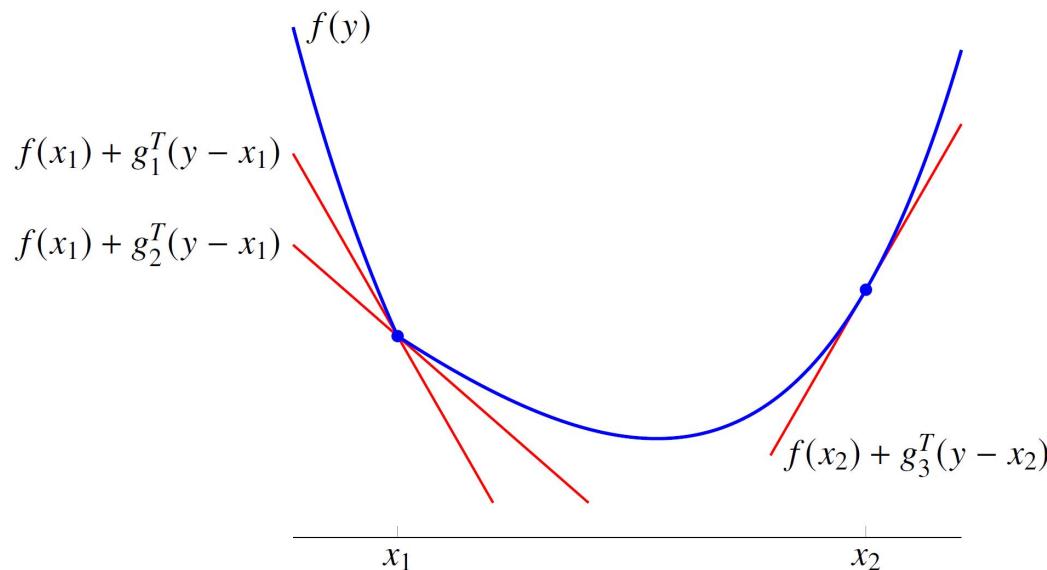




Convex Function Property

Sub-differential

$$\partial f(x) = \{g : f(y) > f(x) + (y - x)^T g, \forall y\}$$



Optimality: $0 \in \partial f(x^*)$

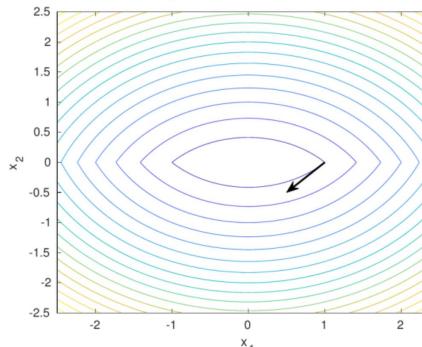
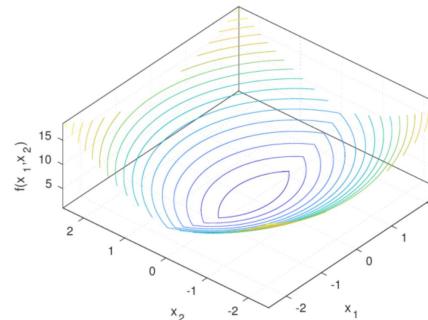
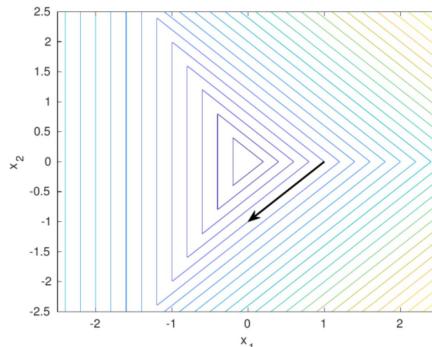
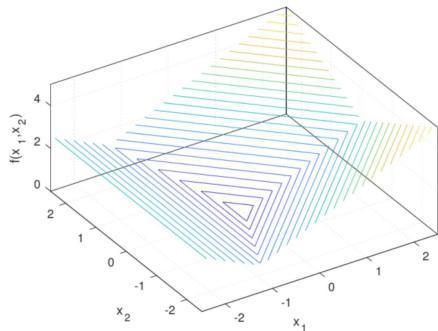


Convex Function Property

A direction in negative sub-differential is not necessarily a descent one

$$f(\mathbf{x}) = \max[-x_1, x_1 - x_2, x_1 + x_2]$$

$$f(\mathbf{x}) = \max[x_1^2 + (x_2 + 1)^2, x_1^2 + (x_2 - 1)^2]$$



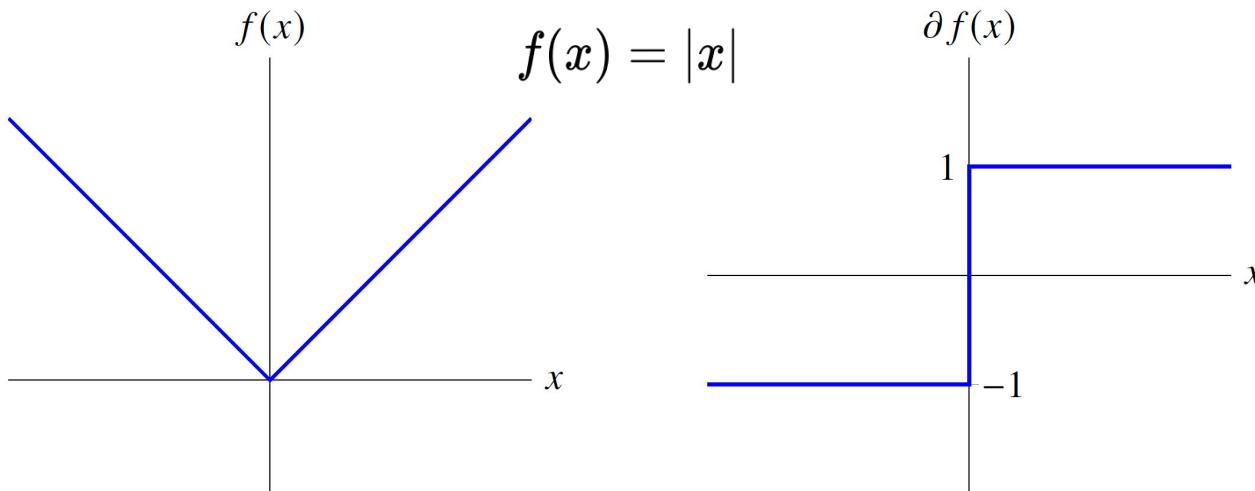
Steepest descent direction is the least-norm one in neg sub-diff.



Convex Function Property

Sub-differential example

$$\partial f(x) = \{g : f(y) > f(x) + (y - x)^T g, \forall y\}$$



$$\partial f(0) = [-1, 1]$$



Convex Function Property

Monotonicity: The (sub) gradient of any convex func is monotone

$$\langle y - x, \nabla f(y) - \nabla f(x) \rangle \geq 0$$

or

$$\langle y - x, g_y - g_x \rangle \geq 0, \quad g_x \in \partial f(x), \quad g_y \in \partial f(y)$$

This can be obtained by adding two equations below.

$$f(y) \geq f(x) + \nabla f(x)^T (y - x)$$

$$f(x) \geq f(y) + \nabla f(y)^T (x - y)$$

Unconstrained Optimization for Nonconvex Functions



Unconstrained Optimization

unconstrained optimization

$$\min f(x)$$

$x = (x_1, \dots, x_n) \in \mathbb{R}^n$: optimization variables

$f: \mathbb{R}^n \mapsto \mathbb{R}$: objective function

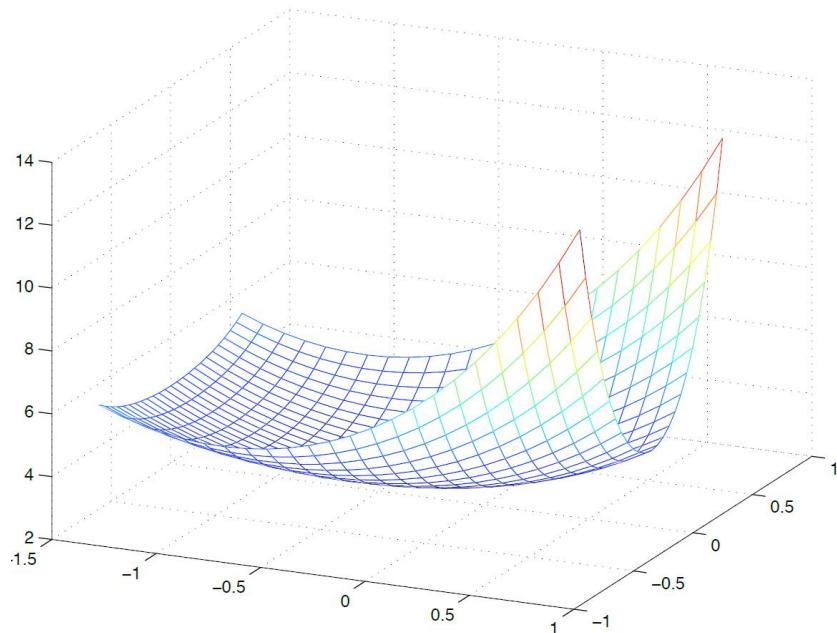
optimal solution x^* has smallest value f among all vectors.

Line-Search Steepest Gradient Descent

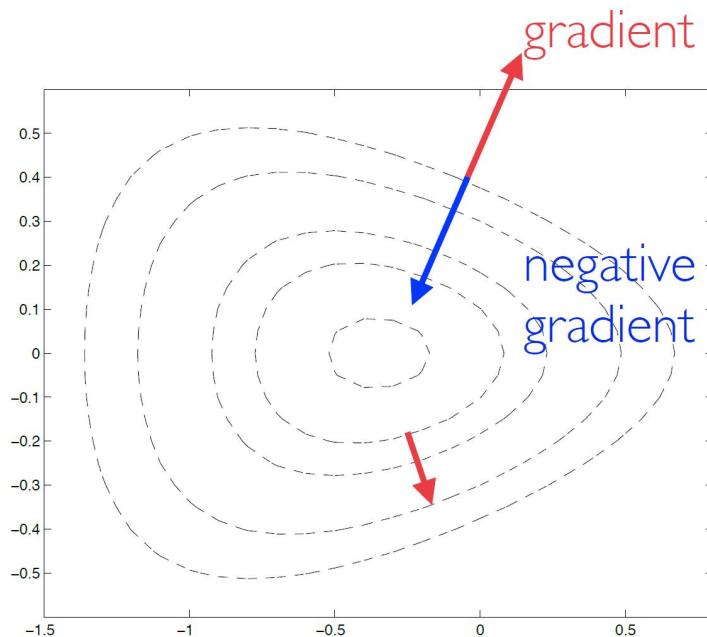


Steepest Gradient Descent

Convex function



Iso-contours





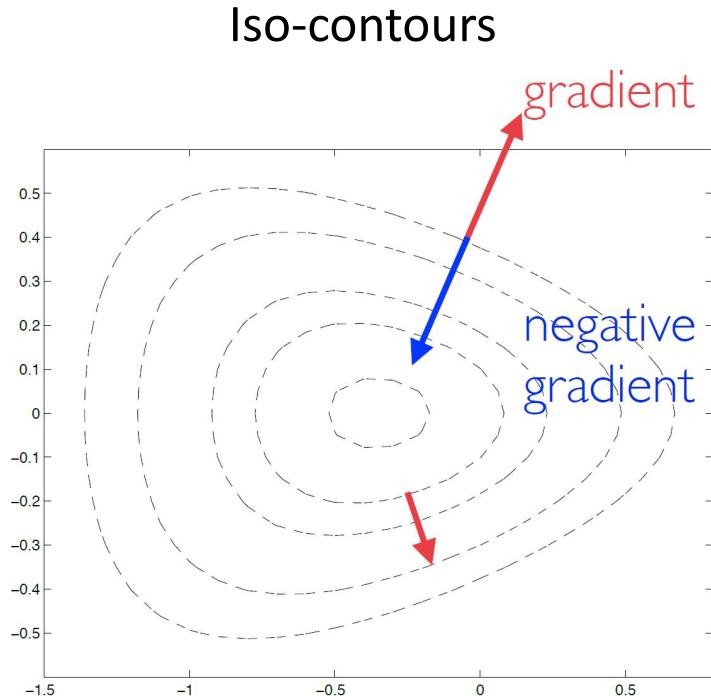
Steepest Gradient Descent

Steepest gradient descent

$$x^{k+1} = x^k - \tau \nabla f(x^k)$$



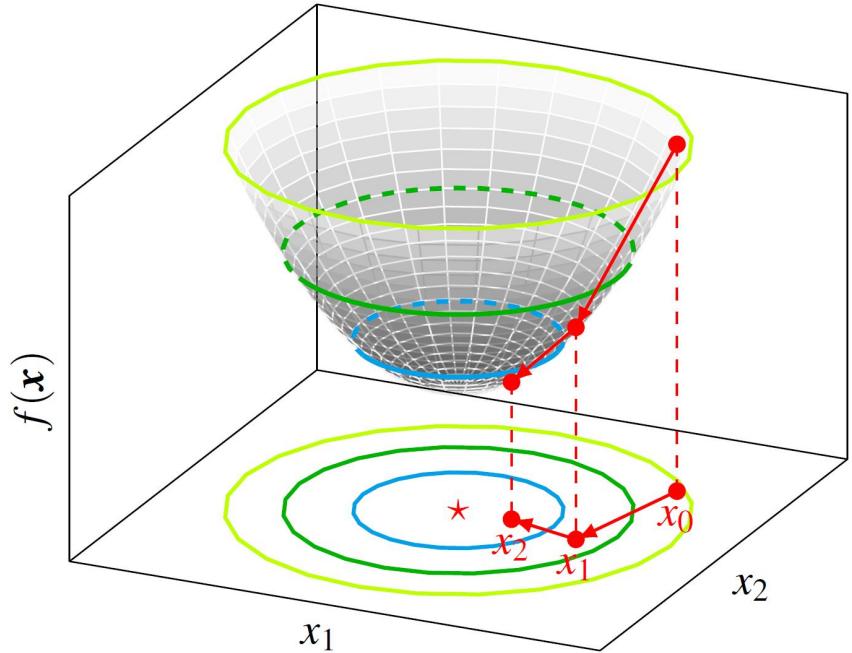
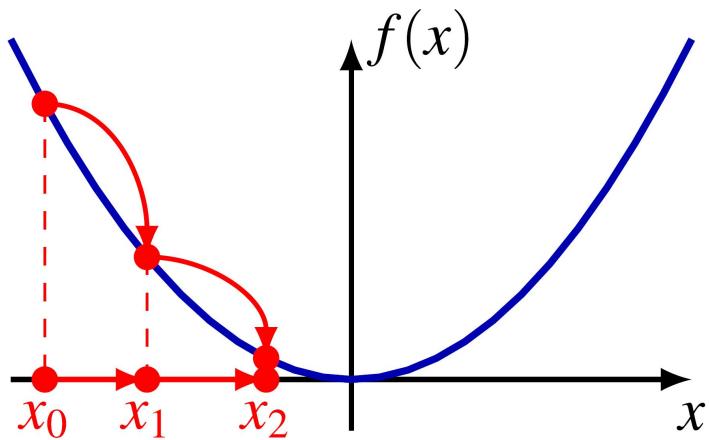
Grad or Least-norm sub-grad





Steepest Gradient Descent

$$\text{Iteration: } x^{k+1} = x^k - \tau \nabla f(x^k)$$



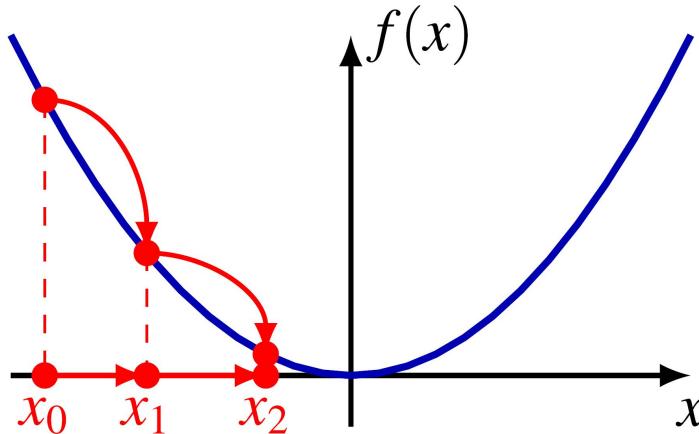
How to choose step size?



Steepest Gradient Descent

$$\begin{aligned}x^{k+1} &= x^k + \tau d, \\d &= -\nabla f(x^k)\end{aligned}$$

- Constant step size $\tau = c$
- Diminishing step size $\tau = c/k$
- Exact line search $\tau = \arg \min_{\alpha} f(x^k + \alpha d)$
- Inexact line search $\tau \in \{\alpha \mid f(x^k) - f(x^k + \alpha d) \geq -c \cdot \alpha d^T \nabla f(x^k)\}$

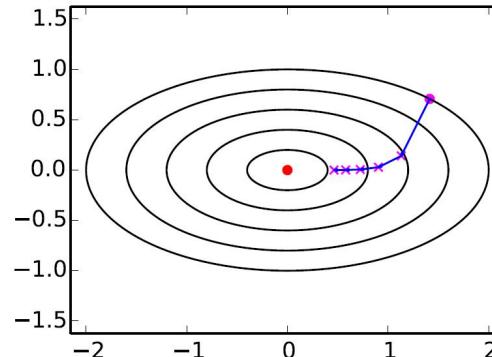
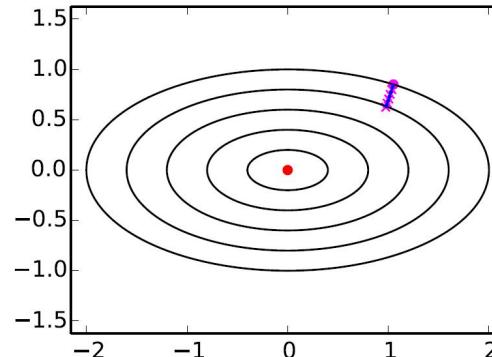
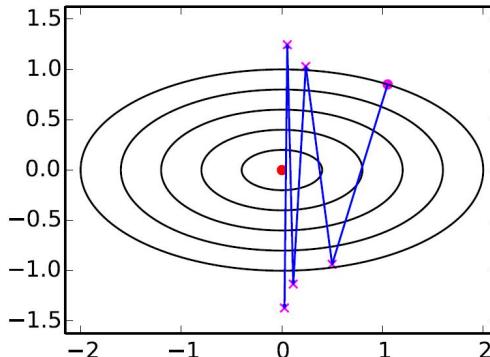
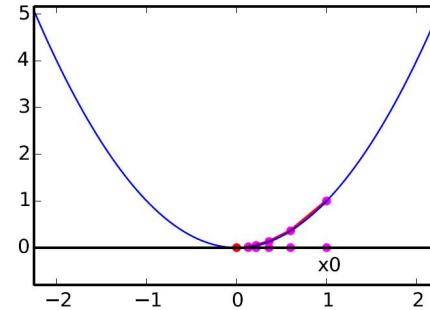
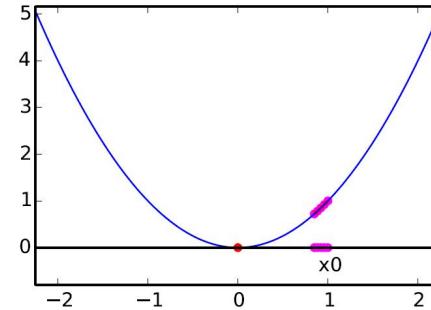
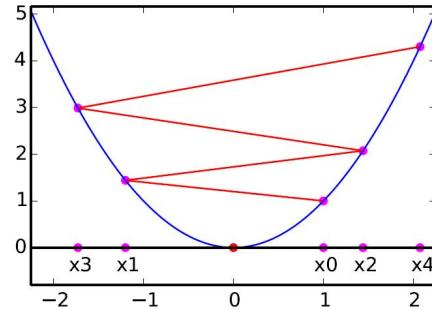




Steepest Gradient Descent

Constant step size

- Too large: may oscillate and diverge
- Too small: may be too slow
- “Just right”: fast convergence





Steepest Gradient Descent

- Constant step size $\tau = c$ Not intelligent
- Diminishing step size $\tau = c/k$ Robbins-Monro rule for expensive func
- Exact line search $\tau = \arg \min_{\alpha} f(x^k + \alpha d)$ Generally nontrivial
- Inexact line search $\tau \in \{\alpha \mid f(x^k) - f(x^k + \alpha d) \geq -c \cdot \alpha d^T \nabla f(x^k)\}$

Armijo condition, easy to satisfy

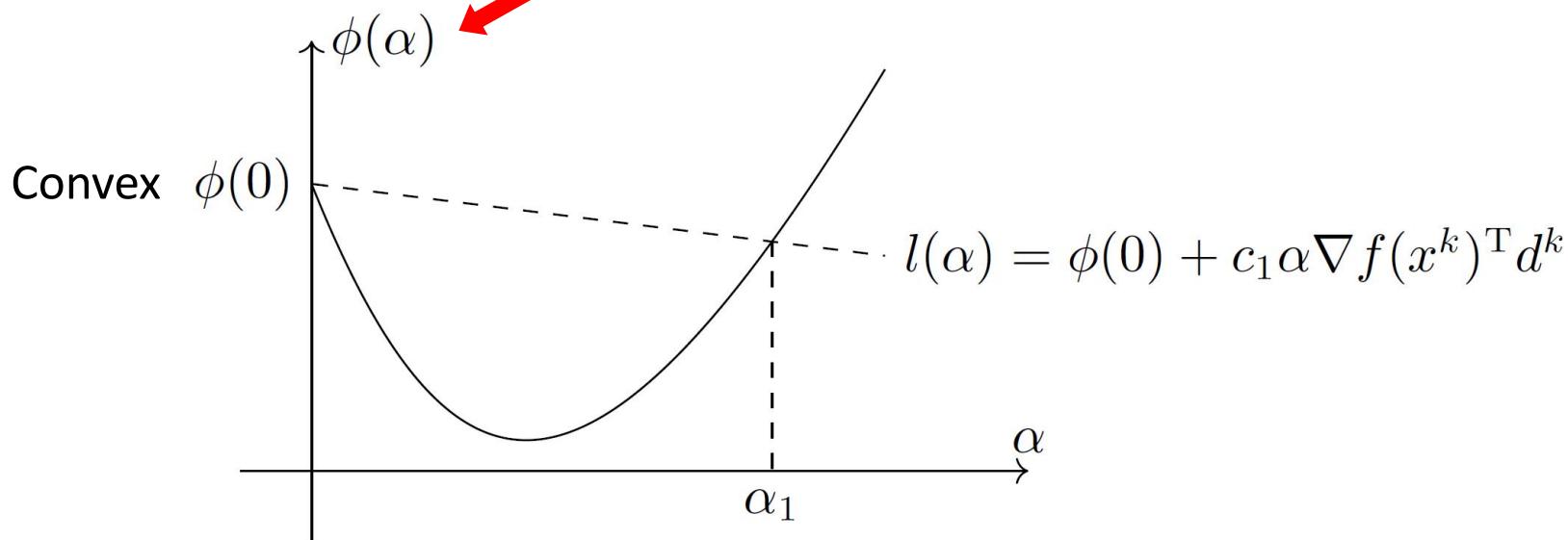


Steepest Gradient Descent

Armijo condition (sufficient decrease condition)

$$\tau \in \{\alpha \mid f(x^k) - f(x^k + \alpha d) \geq -c \cdot \alpha d^T \nabla f(x^k)\}$$

$c \in (0, 1)$





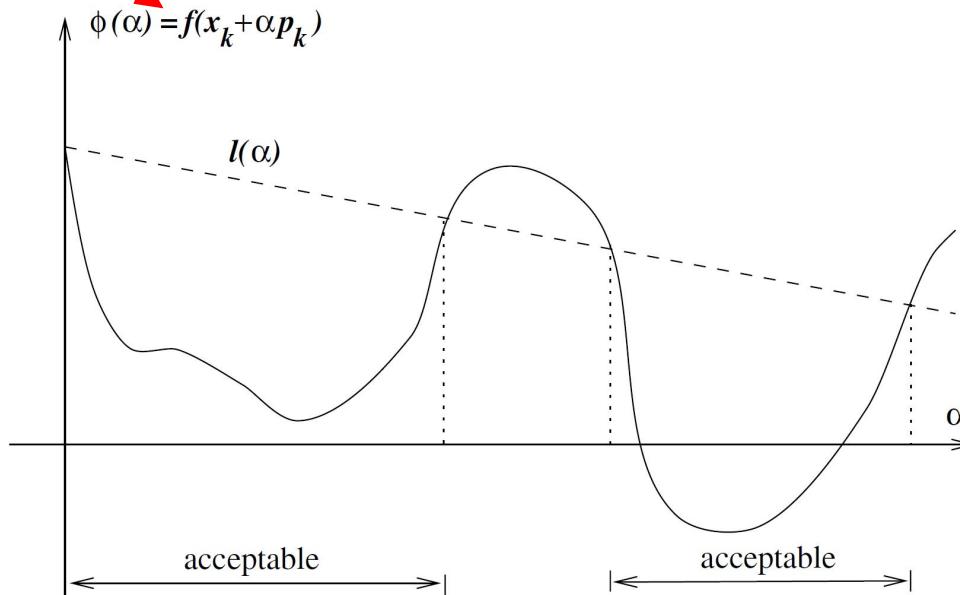
Steepest Gradient Descent

Armijo condition (sufficient decrease condition)

$$\tau \in \{\alpha \mid f(x^k) - f(x^k + \alpha d) \geq -c \cdot \alpha d^T \nabla f(x^k)\}$$

$c \in (0, 1)$

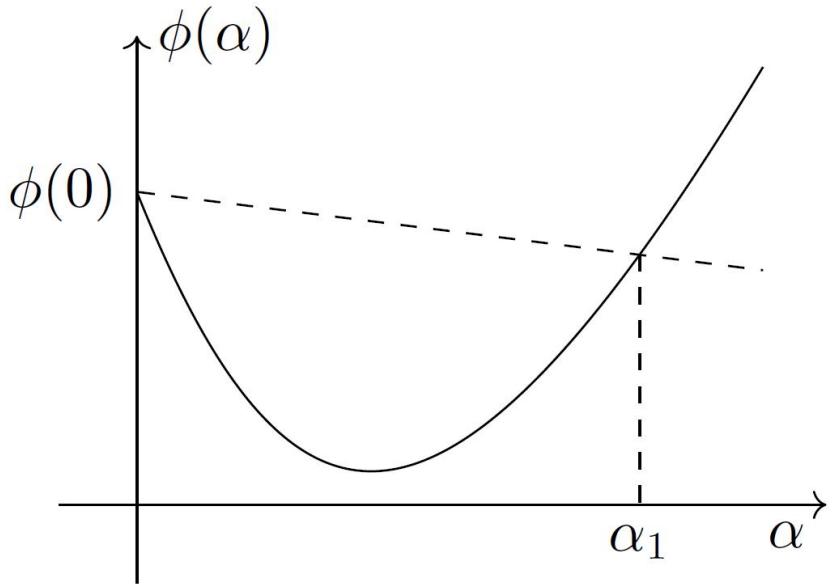
Nonconvex





Steepest Gradient Descent

Backtracking/Armijo line search



Choose search direction: $d = -\nabla f(x^k)$
While $f(x^k + \tau d) > f(x^k) + c \cdot \tau d^T \nabla f(x^k)$
 $\tau \leftarrow \tau/2$
Update iterate $x^{k+1} = x^k + \tau d$

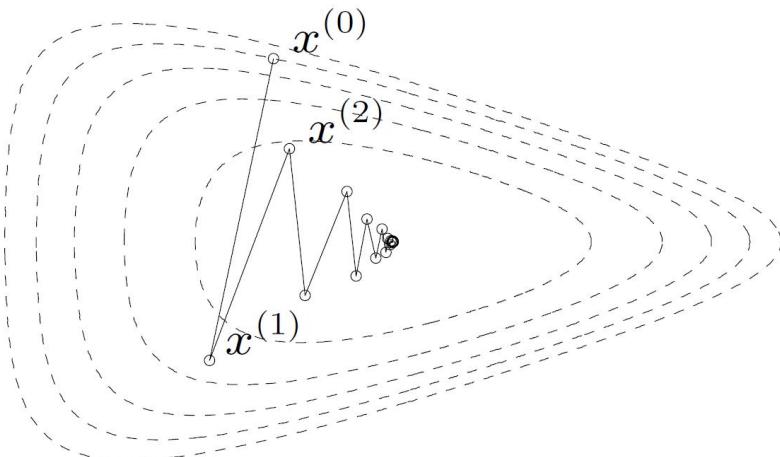
Repeat this until gradient is small
or subdifferential contains zero.



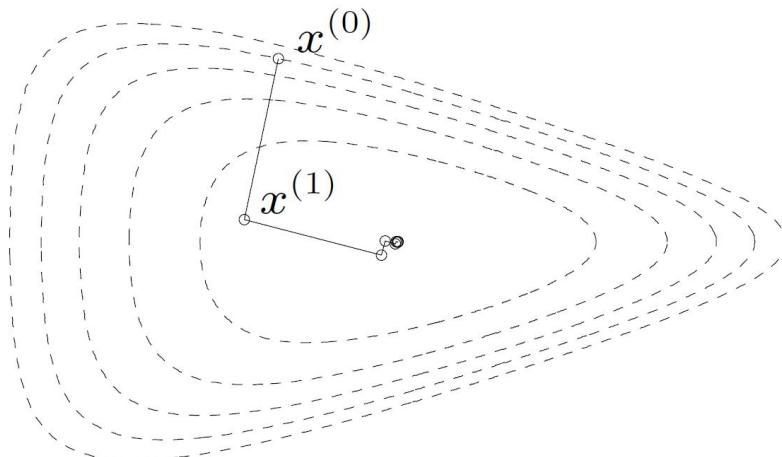
Steepest Gradient Descent

Exact/Inexact line search

$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$$



Inexact



Exact

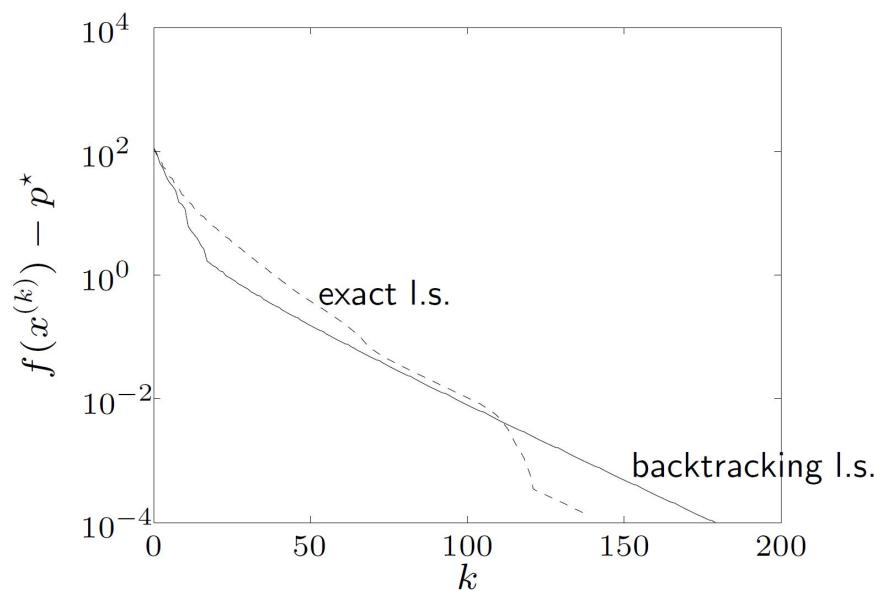
Fewer iterations do not guarantee higher efficiency



Steepest Gradient Descent

Exact/Inexact line search

$$f(x) = c^T x - \sum_{i=1}^{500} \log(b_i - a_i^T x), \quad x \in \mathbb{R}^{100}$$



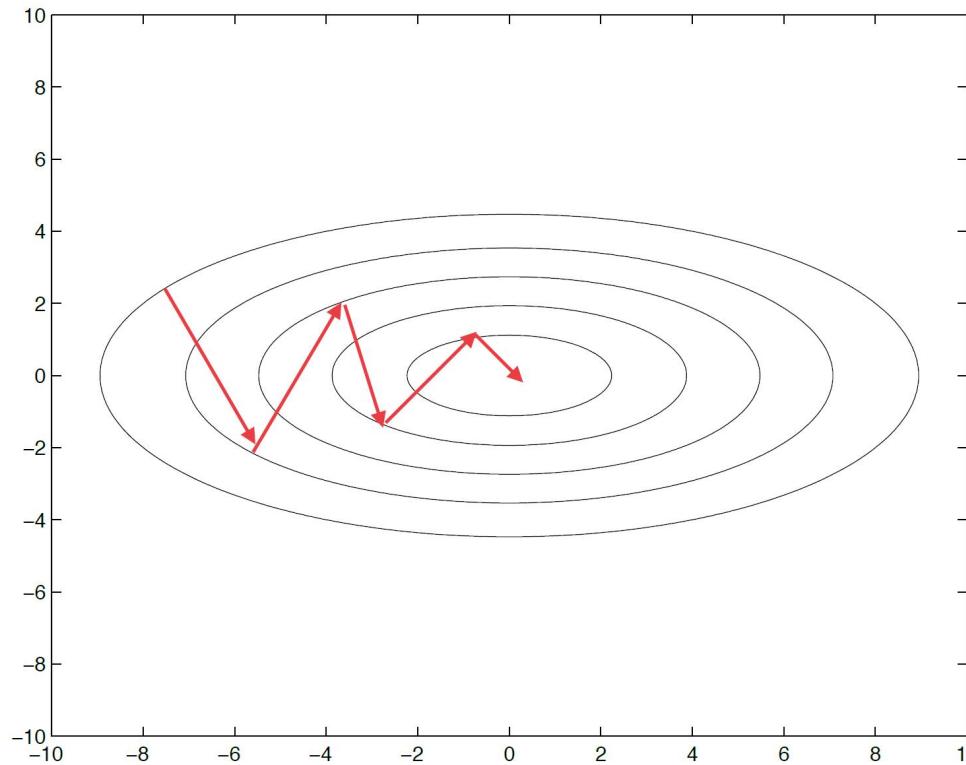
Fewer iterations do not guarantee higher efficiency



Steepest Gradient Descent

Drawbacks: Gradients are perpendicular to contours

$$\kappa = 2$$



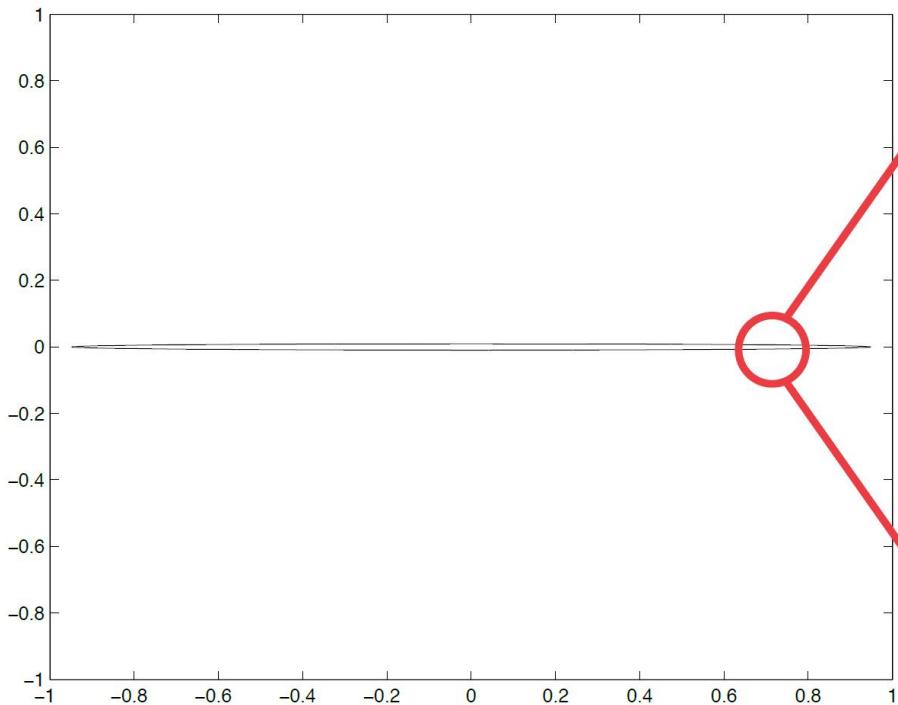


Steepest Gradient Descent

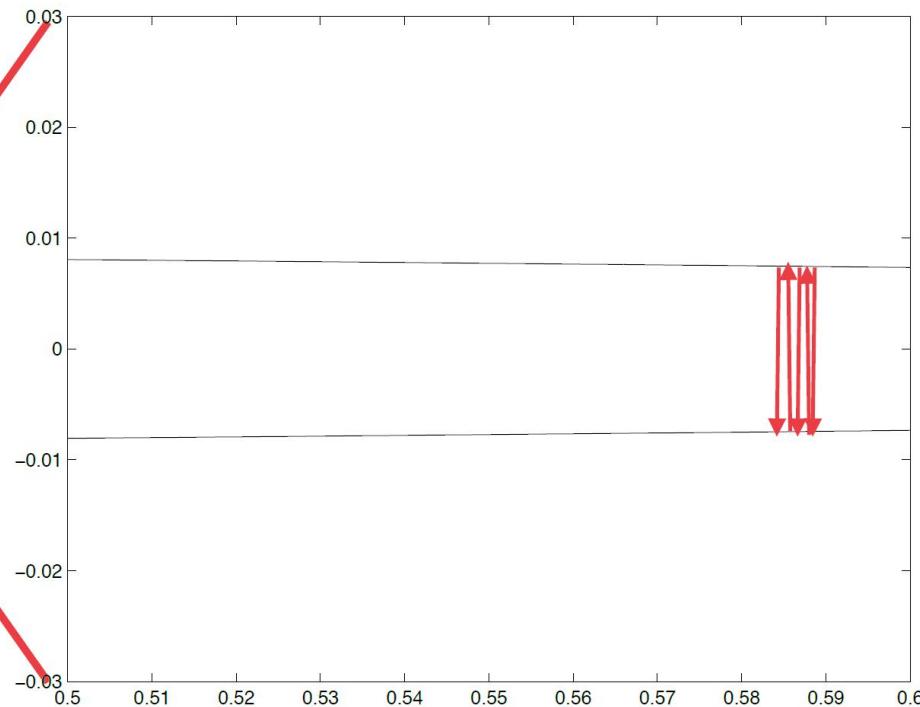
Curvature info is needed!

Drawbacks: Poor conditioning causes performance degeneration

$$\kappa = 100$$



Contours are (almost) parallel!

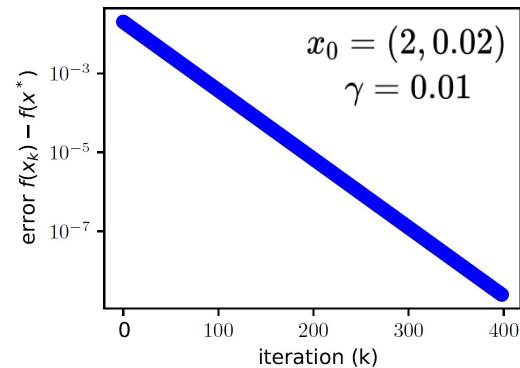
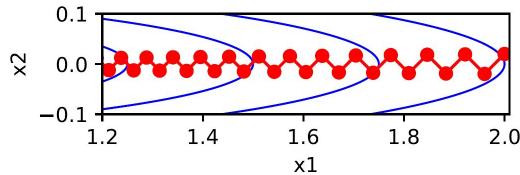
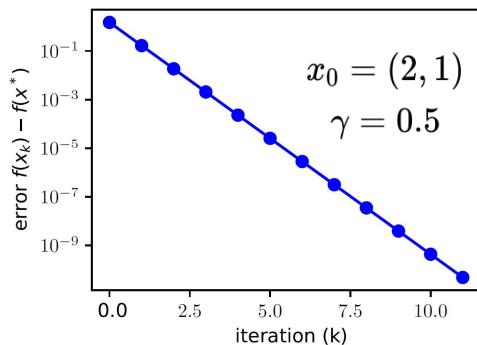
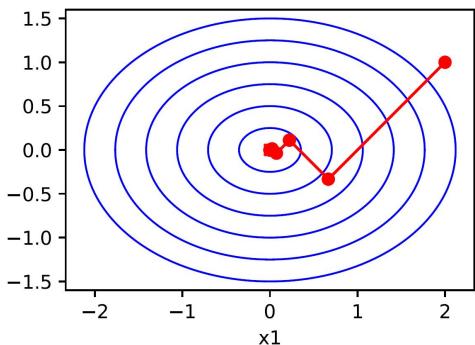




Steepest Gradient Descent

Drawbacks: Poor conditioning causes performance degeneration

$$f(x_1, x_2) = \frac{1}{2} \mathbf{x}^T \mathbf{Q} \mathbf{x} = \frac{\gamma}{2} x_1^2 + \frac{1}{2} x_2^2, \quad \mathbf{Q} = \text{diag}\{\gamma, 1\}$$



Curvature info is needed!

使用高阶信息

Modified Damped Newton's Method



Newton's Method

By second-order Taylor expansion,

$$f(\mathbf{x}) \approx \hat{f}(\mathbf{x}) \triangleq f(\mathbf{x}_k) + \nabla f(\mathbf{x}_k)^T (\mathbf{x} - \mathbf{x}_k) + \frac{1}{2} (\mathbf{x} - \mathbf{x}_k)^T \nabla^2 f(\mathbf{x}_k) (\mathbf{x} - \mathbf{x}_k)$$

Minimizing quadratic approximation

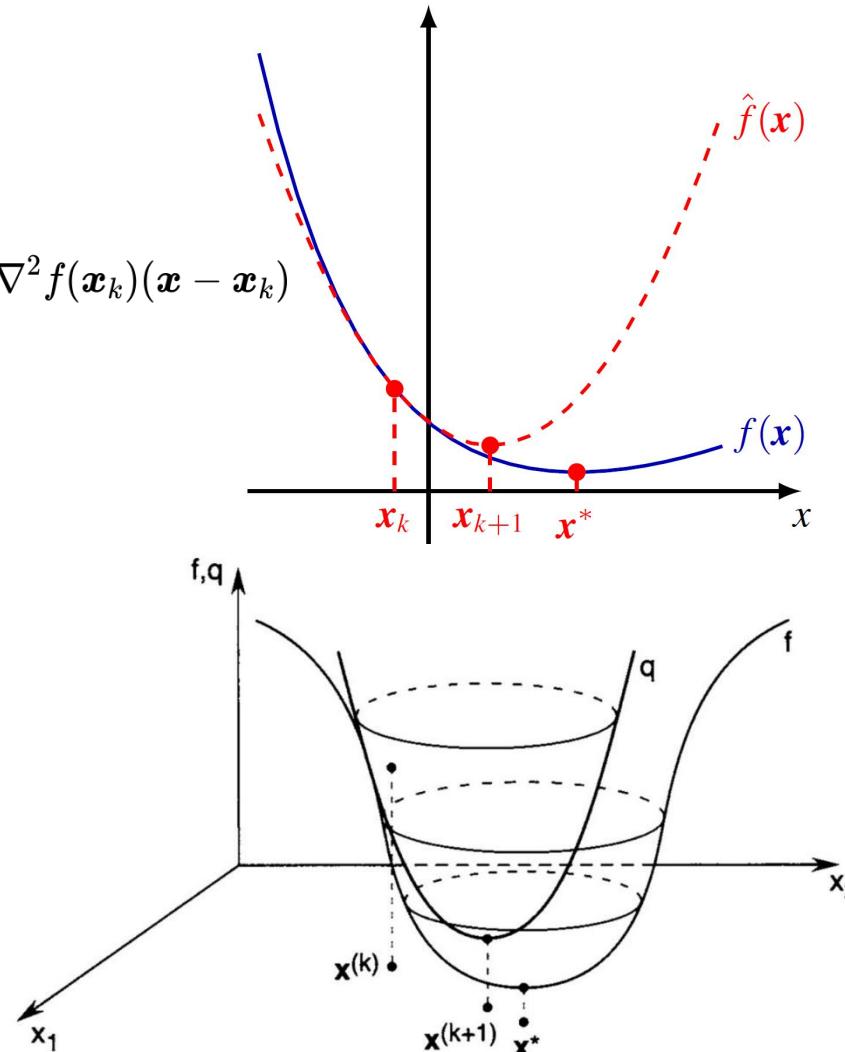
$$\begin{aligned}\nabla \hat{f}(\mathbf{x}) &= \nabla^2 f(\mathbf{x}_k) (\mathbf{x} - \mathbf{x}_k) + \nabla f(\mathbf{x}_k) = \mathbf{0} \\ \implies \mathbf{x} &= \mathbf{x}_k - [\nabla^2 f(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_k)\end{aligned}$$

provided $\nabla^2 f(\mathbf{x}_k) \succ O$

Newton step

$$\boxed{\mathbf{x}_{k+1} = \mathbf{x}_k - [\nabla^2 f(\mathbf{x}_k)]^{-1} \nabla f(\mathbf{x}_k)}$$

Note. If the function is quadratic, then Newton's method gets to the optimum in a single step.

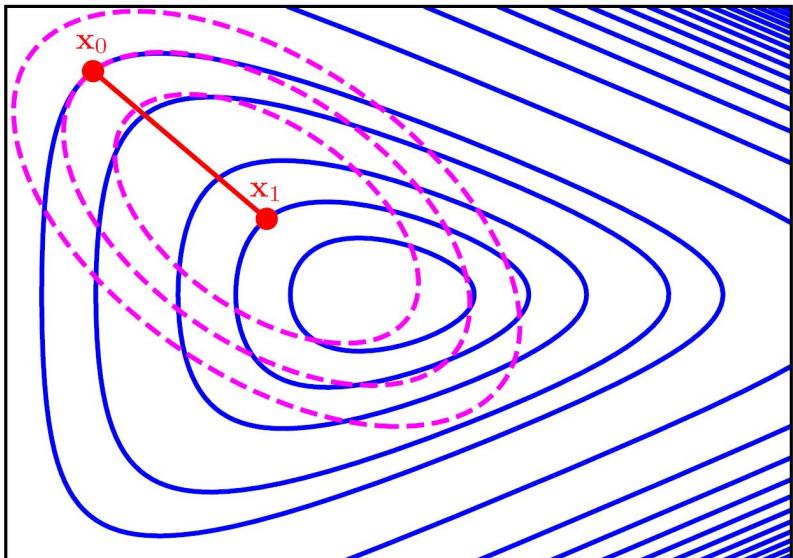




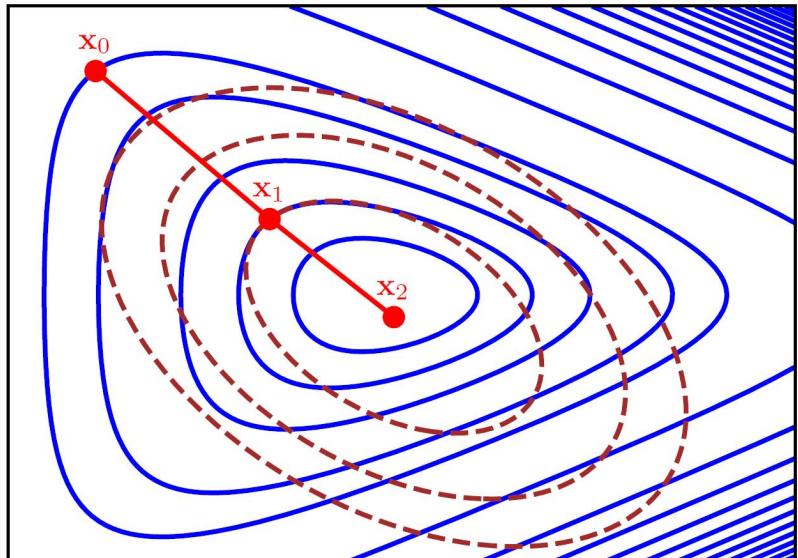
Newton's Method

Example

The magenta curves are the level curves of the quadratic approximation at \mathbf{x}_0



The brown curves are the level curves of the quadratic approximation at \mathbf{x}_1





Newton's Method

Example

$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$$

Newton step at $\mathbf{x}_0 = (-2, 1)^T$

- gradient $\nabla f(\mathbf{x}_0) = e^{-0.1} \begin{pmatrix} e^{x_1+3x_2} + e^{x_1-3x_2} - e^{-x_1} \\ 3e^{x_1+3x_2} - 3e^{x_1-3x_2} \end{pmatrix} \Big|_{\mathbf{x}=\mathbf{x}_0} = \begin{pmatrix} -4.22019458 \\ 7.36051909 \end{pmatrix}$

- Hessian $\nabla^2 f(\mathbf{x}_0) = e^{-0.1} \begin{pmatrix} e^{x_1+3x_2} + e^{x_1-3x_2} + e^{-x_1} & 3e^{x_1+3x_2} - 3e^{x_1-3x_2} \\ 3e^{x_1+3x_2} - 3e^{x_1-3x_2} & 9e^{x_1+3x_2} + 9e^{x_1-3x_2} \end{pmatrix} \Big|_{\mathbf{x}=\mathbf{x}_0}$ $= \begin{pmatrix} 9.1515943 & 7.36051909 \\ 7.36051909 & 22.19129872 \end{pmatrix}$

- Newton step

$$\mathbf{x}_1 = \mathbf{x}_0 - [\nabla^2 f(\mathbf{x}_0)]^{-1} \nabla f(\mathbf{x}_0) = \begin{pmatrix} -1.00725064 \\ 0.33903509 \end{pmatrix}$$

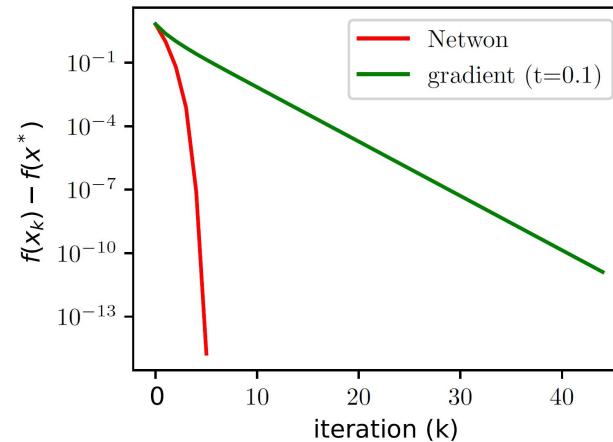
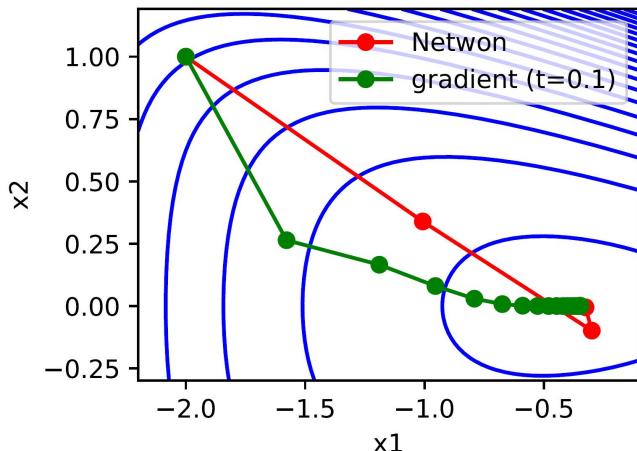


Newton's Method

Example

$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$$

Solution using Newton's method and gradient descent with constant step size 0.1



- Newton's method takes a more “direct” path
- Newton's method requires much fewer iterations, but each iteration is more expensive



Newton's Method

Three aspect to evaluate a numerical optimization method:

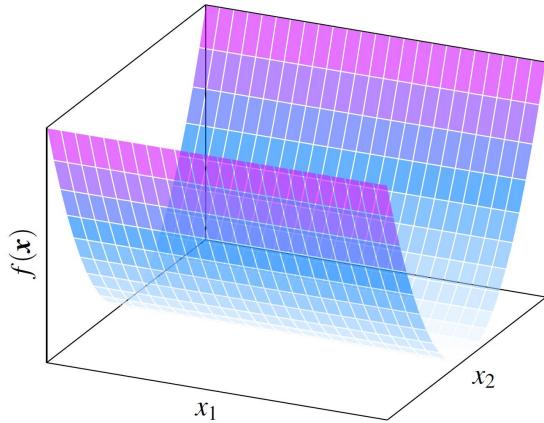
1. Convergence speed (How to measure the rate? In Lec2.).
2. Stability when applied to different functions.
3. Computation work per iteration.



Newton's Method

Drawbacks: In practice Hessian can be singular and indefinite

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - [\nabla^2 f(\boldsymbol{x}_k)]^{-1} \nabla f(\boldsymbol{x}_k)$$



Singular Hessian!

If start at a point of negative curvature, the Newton goes up!

(negative) search direction: must form an acute angle with the gradient



Practical Newton's Method

```
initialization  $\mathbf{x} \leftarrow \mathbf{x}_0 \in \mathbb{R}^n$ 
while  $\|\nabla f(\mathbf{x})\| > \delta$  do
     $\mathbf{d} \leftarrow -\mathbf{M}^{-1}\nabla f(\mathbf{x})$ 
     $t \leftarrow$  backtracking line search
     $\mathbf{x} \leftarrow \mathbf{x} + t\mathbf{d}$ 
end while
return
```

- Choose a positive-definite \mathbf{M} that is close to the Hessian
- Solve the linear system via factorization instead of inversion
- line search does not need grad and Hessian



Practical Newton's Method

$$[\nabla^2 f(\mathbf{x})] \mathbf{d} = -\nabla f(\mathbf{x})$$

is bad conditioned when Hessian is positive semi-definite (PSD) or indefinite.

- If function is convex, its Hessian must be PSD. We choose a \mathbf{M} as

$$\mathbf{M} = \nabla^2 f(\mathbf{x}) + \epsilon \mathbf{I}, \quad \epsilon = \min(1, \|\nabla f(\mathbf{x})\|_\infty)/10$$

Since \mathbf{M} is PD, the search direction is solved by Cholesky factorization

$$\mathbf{M}\mathbf{d} = -\nabla f(\mathbf{x}), \quad \mathbf{M} = \mathbf{L}\mathbf{L}^T$$

where \mathbf{L} is a lower triangular matrix.

- If function is nonconvex, its Hessian can be indefinite. We compute \mathbf{M} on the fly through

Bunch-Kaufman Factorization

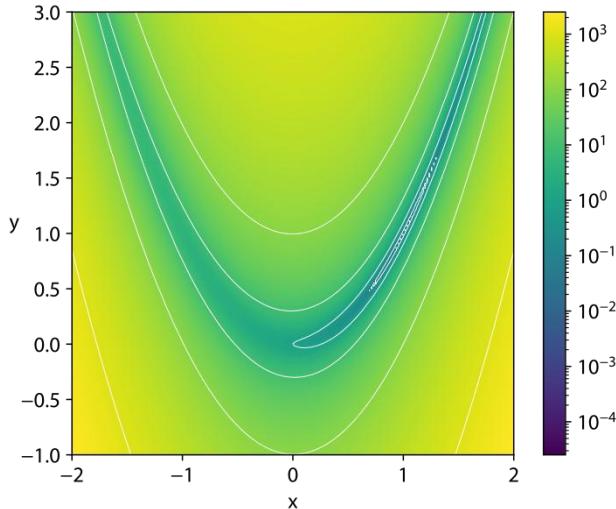
$$\mathbf{M}\mathbf{d} = -\nabla f(\mathbf{x}), \quad \mathbf{M} = \mathbf{L}\mathbf{B}\mathbf{L}^T$$

where \mathbf{B} is block diagonal matrix with block size 1x1 or 2x2. All 2x2 blocks contain nonpositive eigenvalues, which are easy to modify.



Homework

Implement the line-search steepest gradient descent with Armijo condition.



$$f(\mathbf{x}) = f(x_1, x_2, \dots, x_N) = \sum_{i=1}^{N/2} \left[100(x_{2i-1}^2 - x_{2i})^2 + (x_{2i-1} - 1)^2 \right]$$

Your program should properly minimize this Rosenbrock function.

Thanks for Listening!

*Ack.: The slide is prepared by Zhepei Wang and Fei Gao,
improved by Lingfeng Wang, Song Zhao, and Shuo Yang.*