# Introduction to (Mathematical) Optimization

AJ Friend, Nick Henderson
(w/ material from Stephen Boyd and Steven Diamond)

June 28, 2015

# Optimization

**Optimization** finding a best (or good enough) choice among the set of options for a certain objective

# Optimization

Optimization finding a best (or good enough) choice among the set of options for a certain objective

- system: mathematical model
- change: change to input variables (parameters)
- outcome: a measure of performance of the model, objective function

# Mathematical optimization

Mathematical optimization problem has form

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le 0, \quad i = 1, \ldots, m \end{array}$$

- $x \in \mathbf{R}^n$ is **decision variable** (to be found)
- $f_0$ is objective function; $f_i$ are constraint functions
- problem data are hidden inside $f_0, \ldots, f_m$

# The good news

**Everything** is an optimization problem

- *choose parameters* in model to fit data (minimize misfit or error on observed data)
- *optimize actions* (minimize cost or maximize profit)
- *allocate resources* over time (minimize cost, power; maximize utility)
- *engineering design* (trade off weight, power, speed, performance, lifetime)

# The bad news

In full generality, optimization problems can be quite difficult

- generally NP-hard
- heuristics required, hand-tuning, luck, babysitting

# The bad news

In full generality, optimization problems can be quite difficult

- generally NP-hard
- heuristics required, hand-tuning, luck, babysitting

But...

- we can do a lot by restricting to convex models (AJ's talk)
- we have good computational tools
  - modeling languages (CVX, CVXPY, JuMP, AMPL, GAMS) to write problems down
  - solvers (IPOPT, SNOPT, Gurobi, CPLEX, Sedumi, SDPT3, ...) to obtain solutions

# Example: The Raptor Problem

See other slides

# Optimization in one variable

$$\text{minimize} \quad f(x) \in C^2 : \mathbf{R} \to \mathbf{R}$$

# Optimization in one variable

$$\text{minimize} \quad f(x) \in C^2 : \mathbf{R} \to \mathbf{R}$$

- $x$ is a real variable

# Optimization in one variable

$$\text{minimize} \quad f(x) \in C^2 : \mathbf{R} \to \mathbf{R}$$

- $x$ is a real variable
- $f(x)$ is the objective function, which returns a single real number

# Optimization in one variable

$$\text{minimize} \quad f(x) \in C^2 : \mathbf{R} \to \mathbf{R}$$

- $x$ is a real variable
- $f(x)$ is the objective function, which returns a single real number
- Local optimization: look for a point $x^*$ such that $f(x^*) \leq f(x)$ for all points $x$ near $x^*$

# Optimization in one variable

$$\text{minimize} \quad f(x) \in C^2 : \mathbf{R} \to \mathbf{R}$$

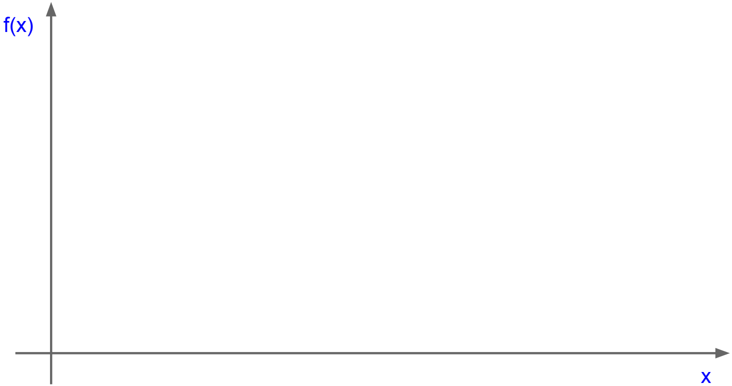
- $x$ is a real variable
- $f(x)$ is the objective function, which returns a single real number
- Local optimization: look for a point $x^*$ such that $f(x^*) \leq f(x)$ for all points $x$ near $x^*$
- Global optimization: look for a point $x^*$ such that $f(x^*) \leq f(x)$ for all points $x$ in domain of interest

# Optimization in one variable

$$\text{minimize} \quad f(x) \in C^2 : \mathbf{R} \to \mathbf{R}$$

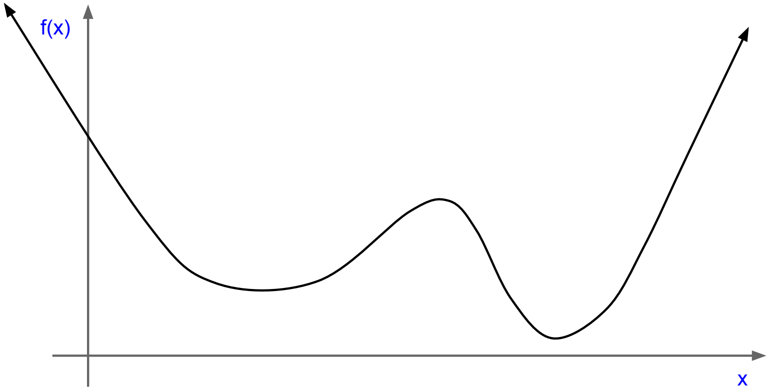
- $x$ is a real variable
- $f(x)$ is the objective function, which returns a single real number
- Local optimization: look for a point $x^*$ such that $f(x^*) \le f(x)$ for all points $x$ near $x^*$
- Global optimization: look for a point $x^*$ such that $f(x^*) \le f(x)$ for all points $x$ in domain of interest
- When $f(x)$ is twice continuously differentiable, then local optimization involves finding a point $x^*$ such that $f'(x^*) = 0$ and $f''(x^*) > 0$

# Optimization in one variable: axis

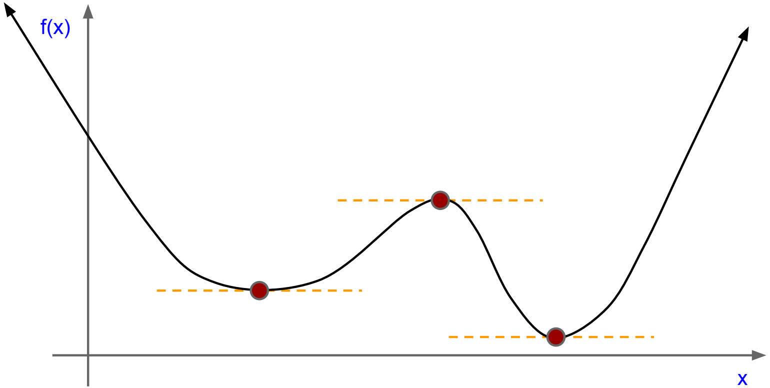


# Optimization in one variable: definitions

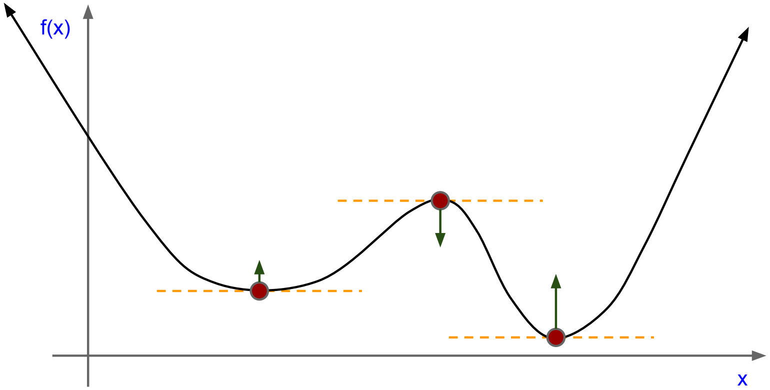
# Optimization in one variable: example objective function



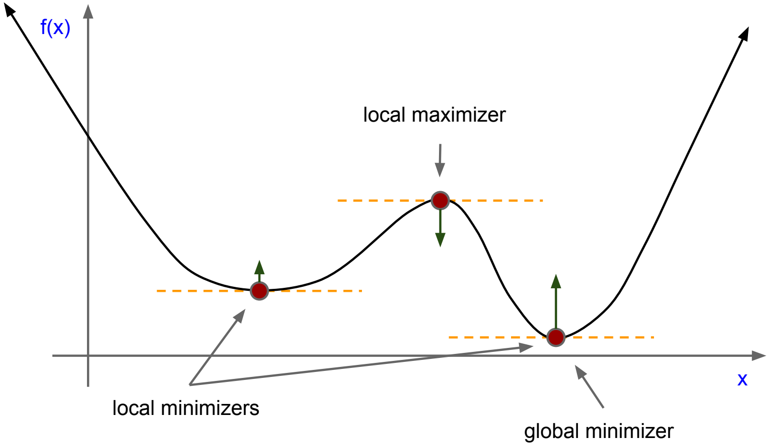
# Optimization in one variable: critical points, $f'(x) = 0$



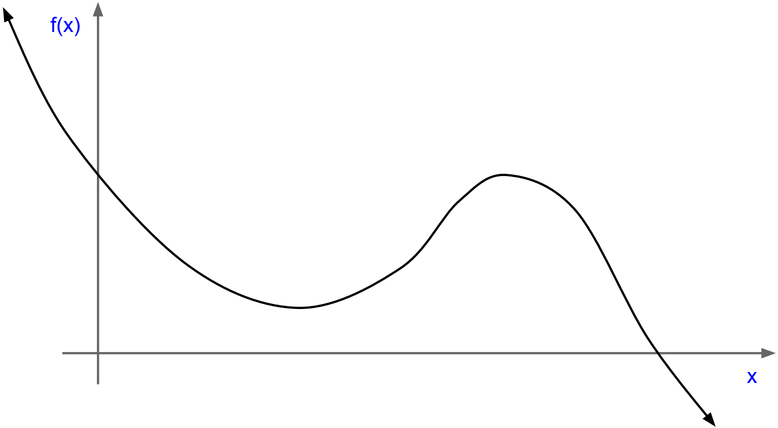
# Optimization in one variable: local optima



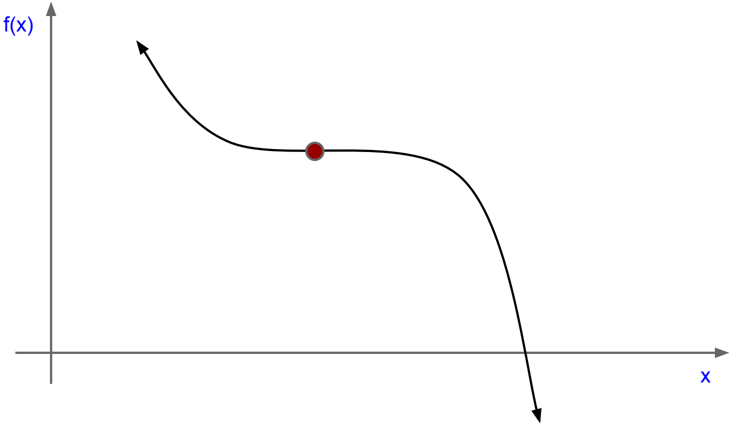
# Optimization in one variable: local optima, $f''(x) = ?$



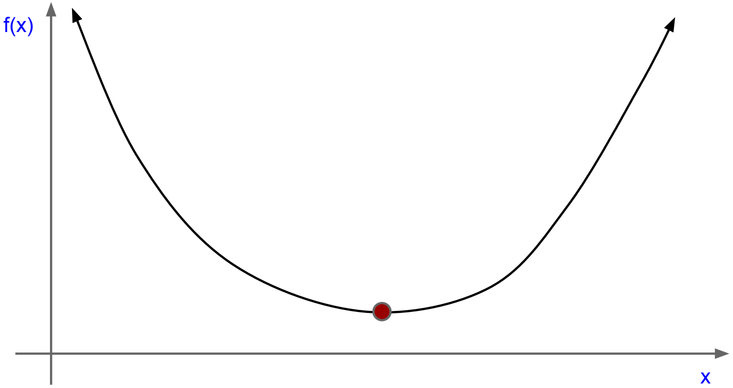
# Optimization in one variable: unbounded below



# Optimization in one variable: saddle point, $f'(x) = 0$ and $f''(x) = 0$



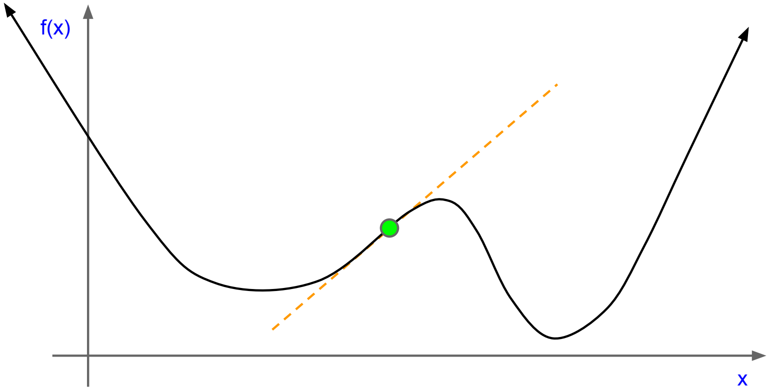
# Optimization in one variable: convex objective



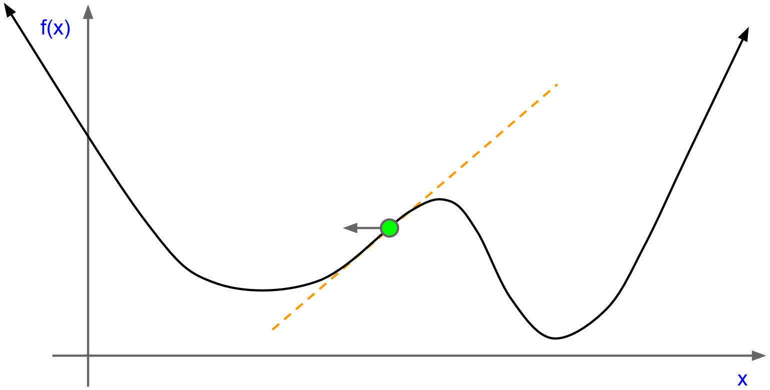
# Key definitions

- *domain*: space for input variable $x$
- *range*: space for output of objective function $f(x)$
- *critical point*: $f'(x) = 0$
- *local minimizer*: $f'(x) = 0$ and $f''(x) > 0$
- *local maximizer*: $f'(x) = 0$ and $f''(x) < 0$
- *saddle point*: $f'(x) = 0$ and $f''(x) = 0$
- *global minimizer*: $x^*$ such that $f(x^*) \leq f(x)$ for all $x$ in domain

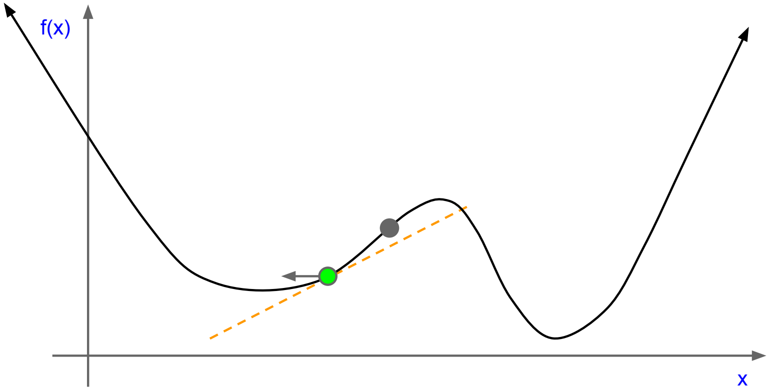
# Optimization in one variable: algorithm basics



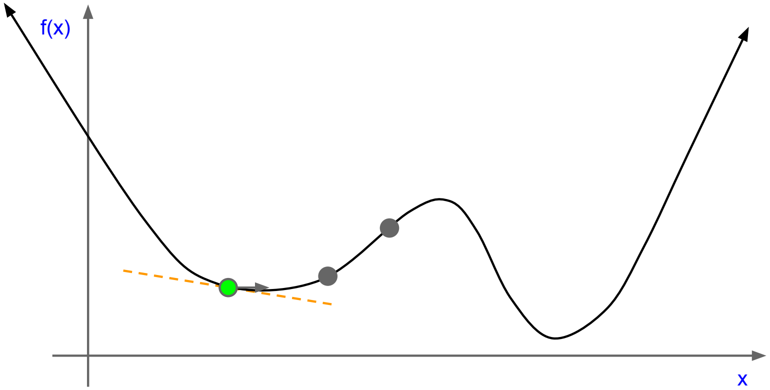
# Optimization in one variable: algorithm basics



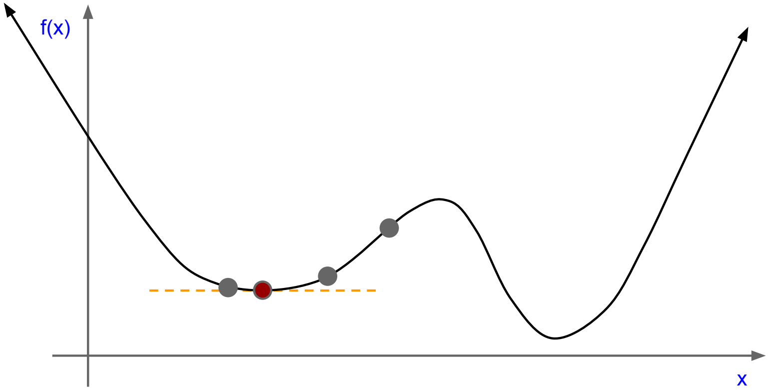
# Optimization in one variable: algorithm basics



# Optimization in one variable: algorithm basics



# Optimization in one variable: algorithm basics



# Optimization in one variable: algorithm basics

# Optimization in one variable: algorithm basics

- Start with an initial guess $x_0$

# Optimization in one variable: algorithm basics

- Start with an initial guess $x_0$
- Goal: generate sequence that converges to solution

$$x_0, x_1, x_2, x_3, \cdots \to x^*$$

# Optimization in one variable: algorithm basics

- Start with an initial guess $x_0$
- Goal: generate sequence that converges to solution

$$x_0, x_1, x_2, x_3, \cdots \to x^*$$

- Notation for sequence and convergence: $\{x_k\} \to x^*$

# Optimization in one variable: algorithm basics

- Start with an initial guess $x_0$
- Goal: generate sequence that converges to solution

$$x_0, x_1, x_2, x_3, \cdots \to x^*$$

- Notation for sequence and convergence: $\{x_k\} \to x^*$
- Key algorithm property: ***descent condition***

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

## Optimization in one variable: algorithm basics

- Start with an initial guess $x_0$
- Goal: generate sequence that converges to solution

$$x_0, x_1, x_2, x_3, \cdots \to x^*$$

- Notation for sequence and convergence: $\{x_k\} \to x^*$
- Key algorithm property: ***descent condition***

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

- Technical algorithm property: ***convergence to solution***

$$|x_{k+1} - x_k| \to 0 \text{ if and only if } f'(x_k) \to 0 \text{ and } \lim_{k\to\infty} f''(x_k) \geq 0$$

# Optimization in many variables

$$\text{minimize} \quad f(x) \in C^2 : \mathbf{R}^n \to \mathbf{R}$$

- $x$ is an $n$-dimensional vector of real variables
- $f(x)$ is the objective function (twice continuously differentiable)
  - First derivative or gradient of $f$ is written $\nabla f(x)$
  - Second derivate or Hessian of $f$ is written $\nabla^2 f(x)$
- We are looking for a point $x^*$ such that $\nabla f(x) = 0$ and $\nabla^2 f(x) \succeq 0$. Note that this is a *local* optimizer
  - $\nabla^2 f(x) \succeq 0$ means that all the eigenvalues of $\nabla^2 f(x)$ are non-negative

# The gradient $\nabla f(x)$ in 2 variables

Vector of variables:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Gradient of $f$:

$$\nabla f(x) = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \end{bmatrix}$$

# The Hessian $\nabla^2 f(x)$ in 2 variables

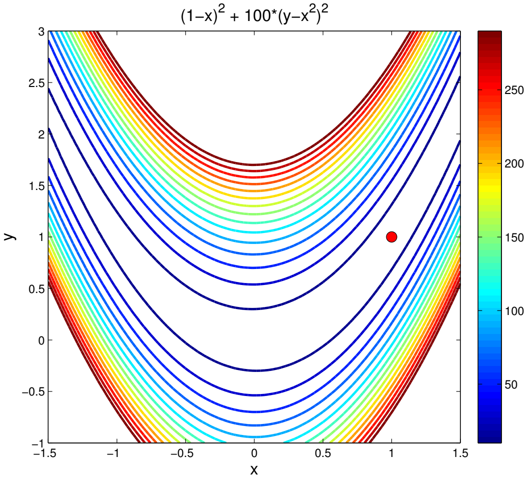
$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{bmatrix}$$

# Let's look at an example

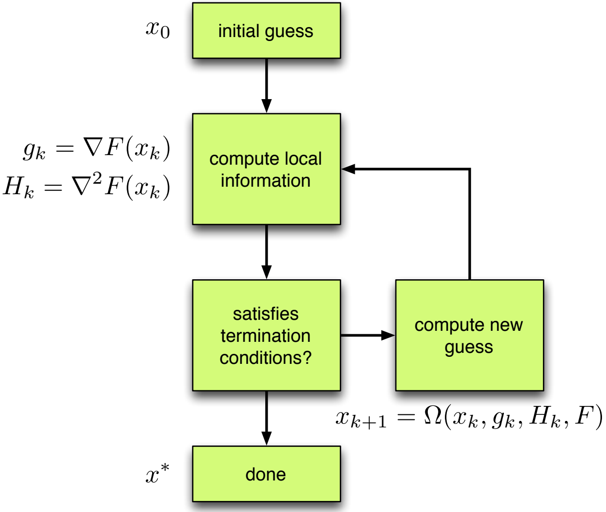
The Rosenbrock function:

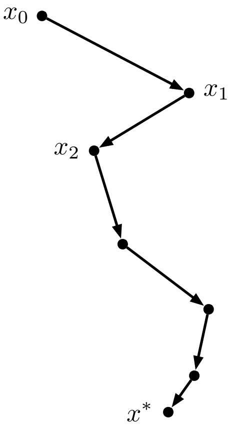
$$f(x, y) = (1 - x)^2 + 100 \left(y - x^2\right)^2$$

# Rosenbrock contours



# Basic optimization algorithm



# Line search algorithms

1. compute a search direction $p_k$
   - for minimization, $p_k$ must be a descent direction, that is $p_k^T g_k < 0$
2. select a step length $\alpha_k$ along $p_k$ such that $f(x_k + \alpha_k p_k) < f(x_k)$
   - (we need more technical requirements here)
3. update the guess $x_{k+1} \leftarrow x_k + \alpha_k p_k$

# Example line search algorithms

Algorithm:

$$x_{k+1} \leftarrow x_k + \alpha_k p_k$$

Gradient descent:

$$p_k = -g_k = -\nabla f(x_k)$$

Modified Newton's method:

$$p_k = -(H_k + \lambda_k I)^{-1} g_k = (\nabla^2 f(x_k) + \lambda_k I)^{-1} \nabla f(x_k)$$

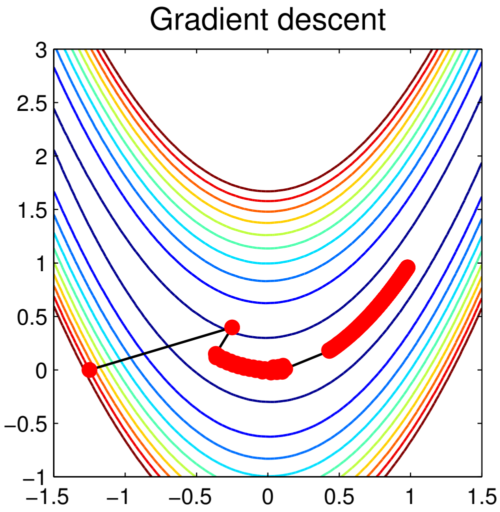
# Step length selection: backtracking

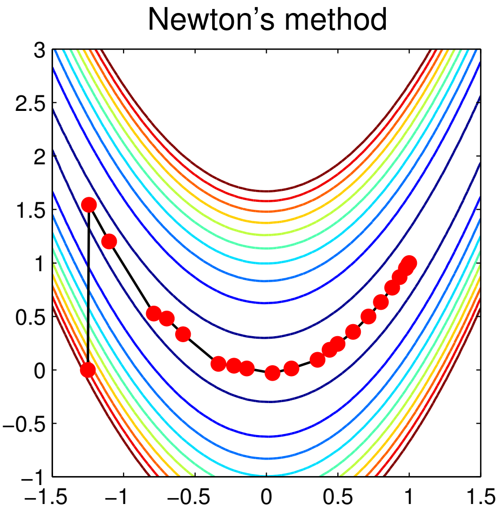
Goal: given $p_k$ find $\alpha$ such that $f(x_k + \alpha p_k) < f(x_k)$.

Procedure: start with initial guess $\alpha > 0$ (use $\alpha = 1$ for Newton's method)

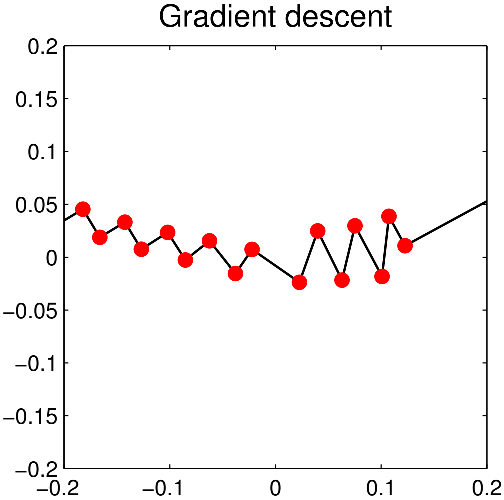
1. if $f(x_k + \alpha p_k) < f(x_k)$, then return $\alpha$, otherwise continue
2. decrease $\alpha$ by some factor $0 < \delta < 1$: $\alpha \leftarrow \delta\alpha$
3. repeat

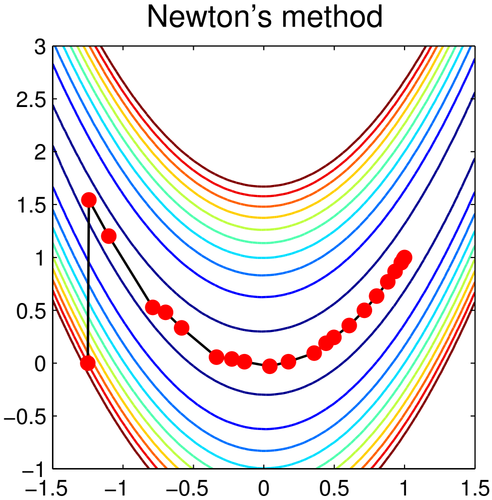
# Optimization on rosenbrock function



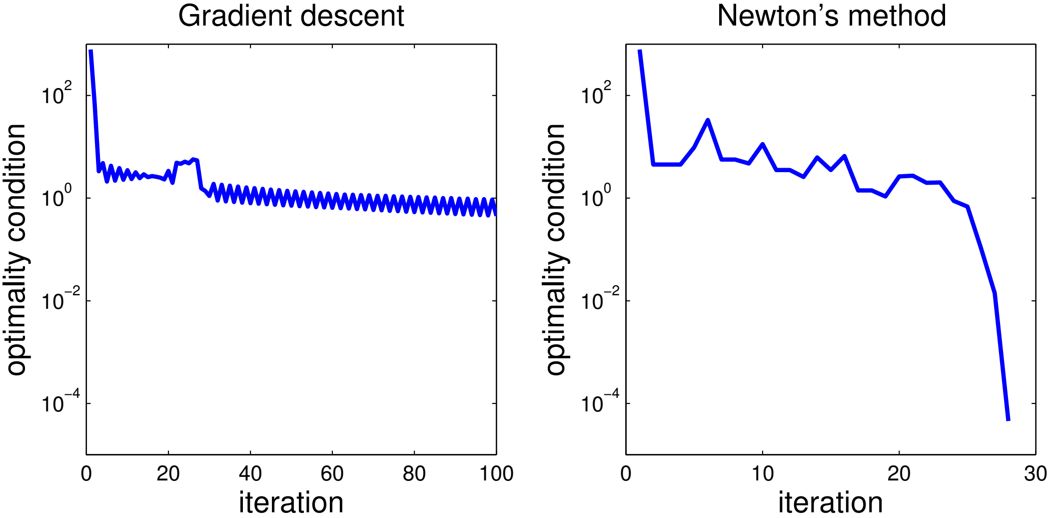



# Optimization on rosenbrock function



# Optimization on rosenbrock function



# Considerations in selecting optimization algorithms

- Computational cost/scale of objective function
- Computational cost of linear algebra associated with optimization algorithm
- Accuracy requirement in your application

# Two very important optimization problems

- linear least squares
- non-linear least squares

# Constraints

- basic idea of constraints
- work through example from multivariate calculus
- introduce idea of multipliers
- equality constraints
- inequality constraints
- linear constraints
- nonlinear constraints

# Penalty and barrier methods

- introduce a penalty into the objective to penalize constraint violation

# Linear programming

# Discrete variables

- Mixed integer programming
- Scheduling problems

# What's next

- Nonlinear programming
- Convex modelling
- Study of algorithms
- Modeling languages
- Automated differentiation