# CE 191: Civil and Environmental Engineering Systems Analysis

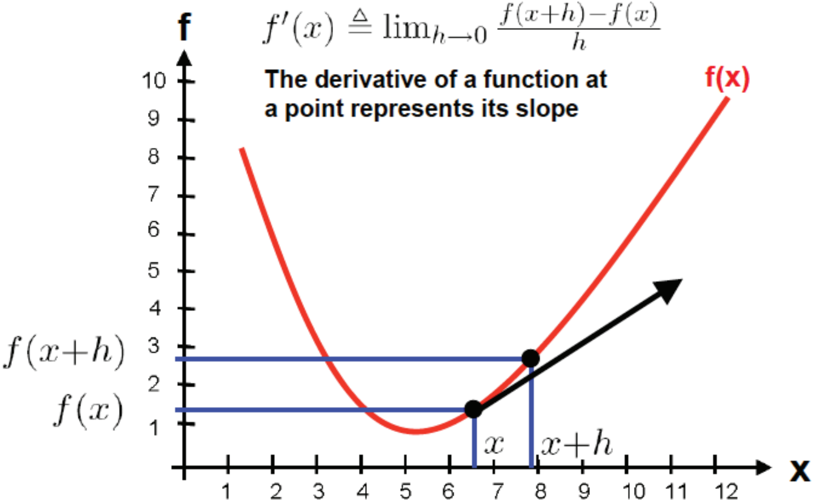
## LEC 11 : Gradient Descent

Professor Scott Moura
Civil & Environmental Engineering
University of California, Berkeley

Fall 2014

# Recall Definition of Derivative

$$f'(x) \triangleq \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$



# Gradient Descent

**Goal:** Find the minimum of a differentiable function, starting from an initial guess, which cannot be easily visualized.

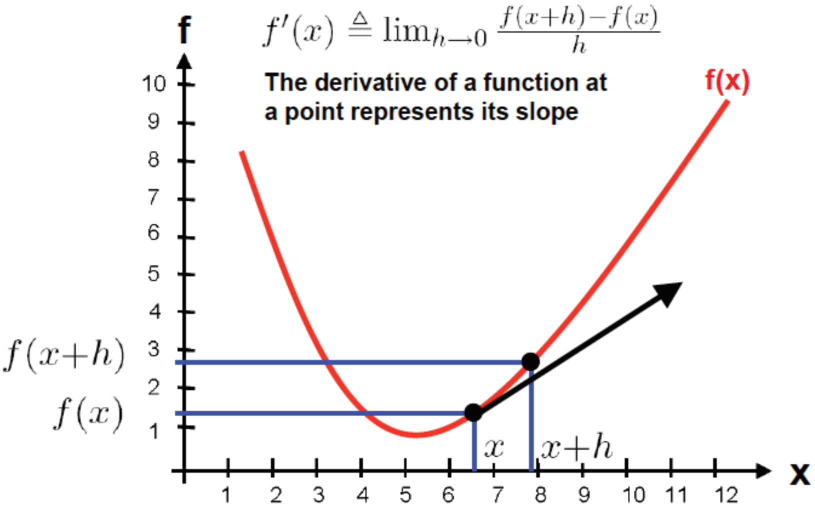
Exs:

$$
\begin{aligned}
f(x) &= \exp(\sin x^2) + \sqrt{x^4 + 3}\,\sin\left[\exp\left(\frac{-1}{1+\epsilon|x|}\right)\right] \\
g(x,y,z) &= \sin\left(\frac{1}{2}x^2 - \frac{1}{4}y^2 + \frac{1}{8}z^2\right)\cos\left(2x + 1 - e^y\right)
\end{aligned}
$$

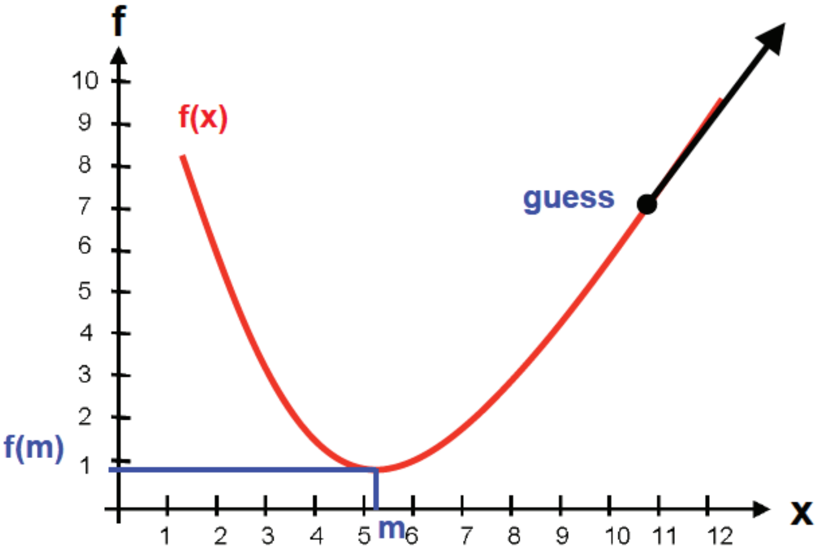
Main idea:

1. Make an initial guess
2. Compute the derivative at this point (i.e. slope)
3. Follow the direction of the slope (i.e. descend)
4. Stop when the slope becomes zero

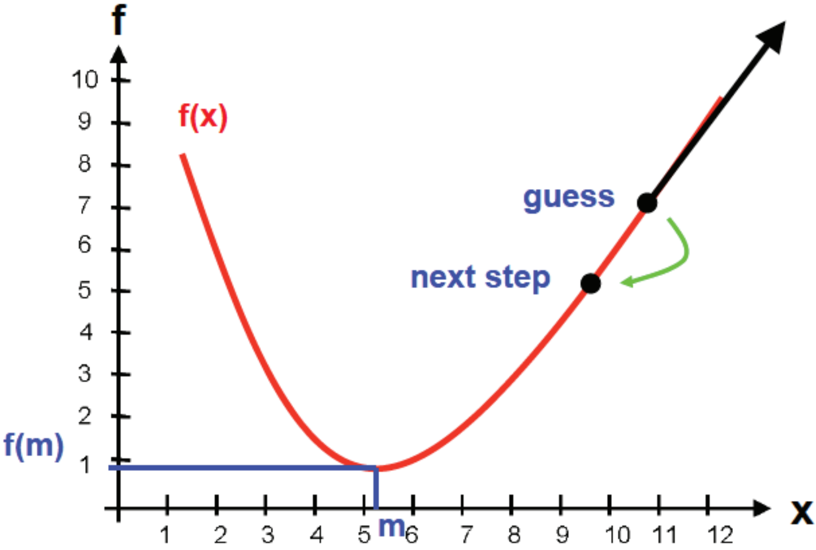
# Recall Definition of Derivative



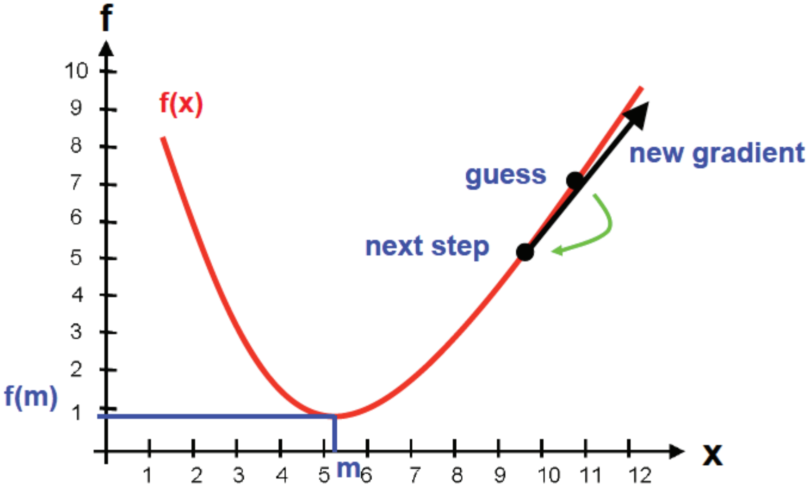
# Illustration of Gradient Descent



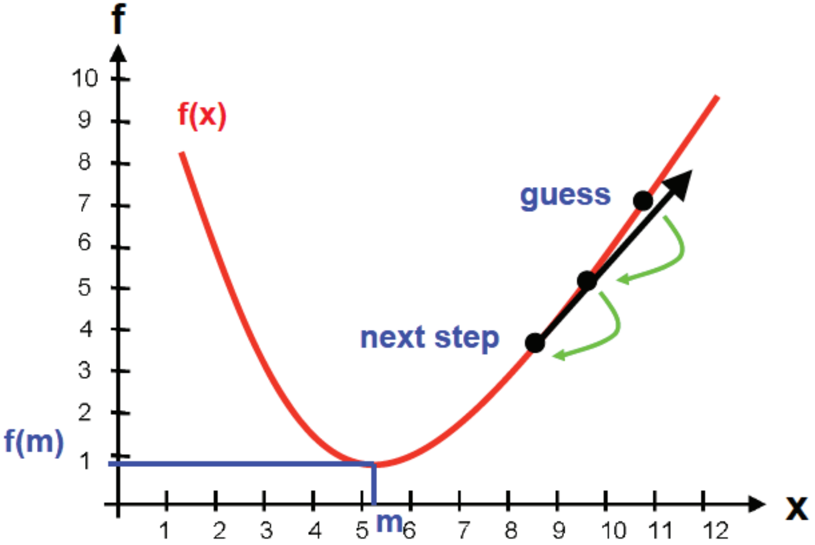
# Illustration of Gradient Descent



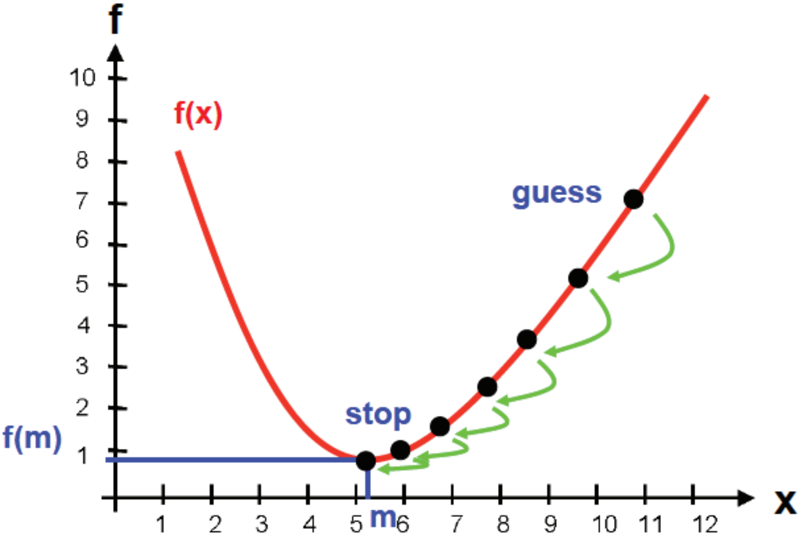
# Illustration of Gradient Descent



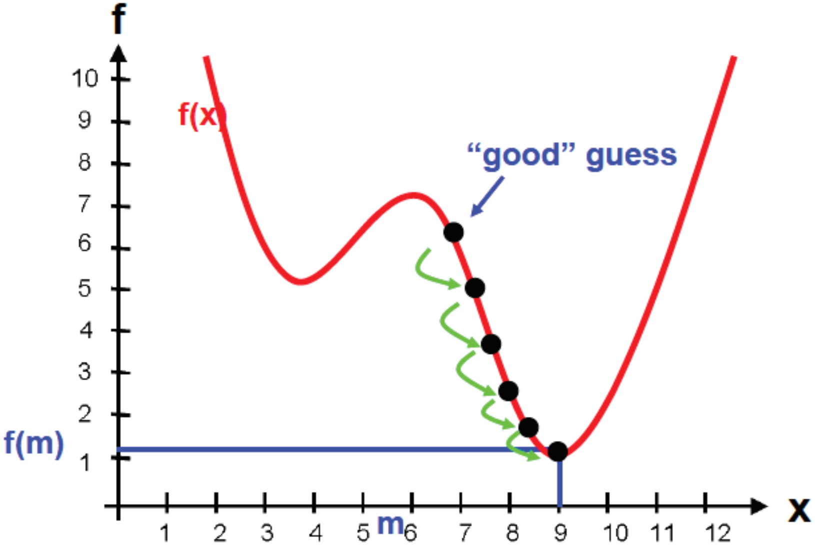
# Illustration of Gradient Descent



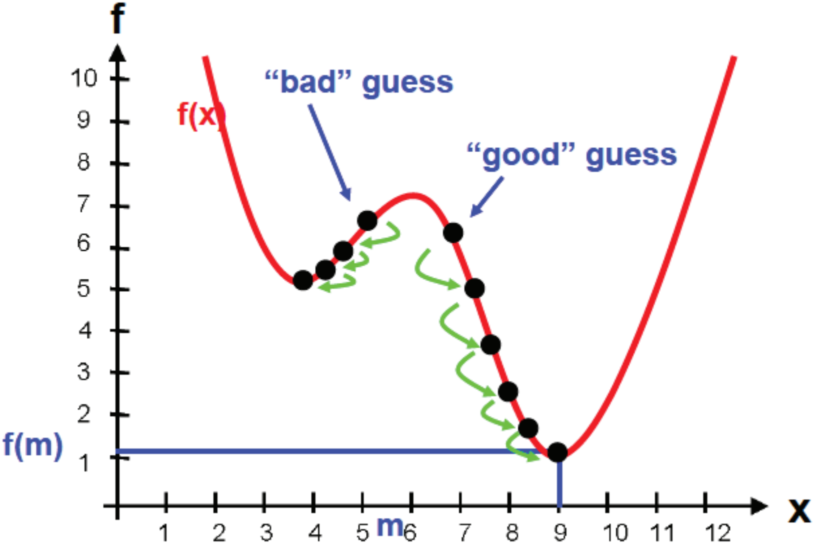
# Illustration of Gradient Descent



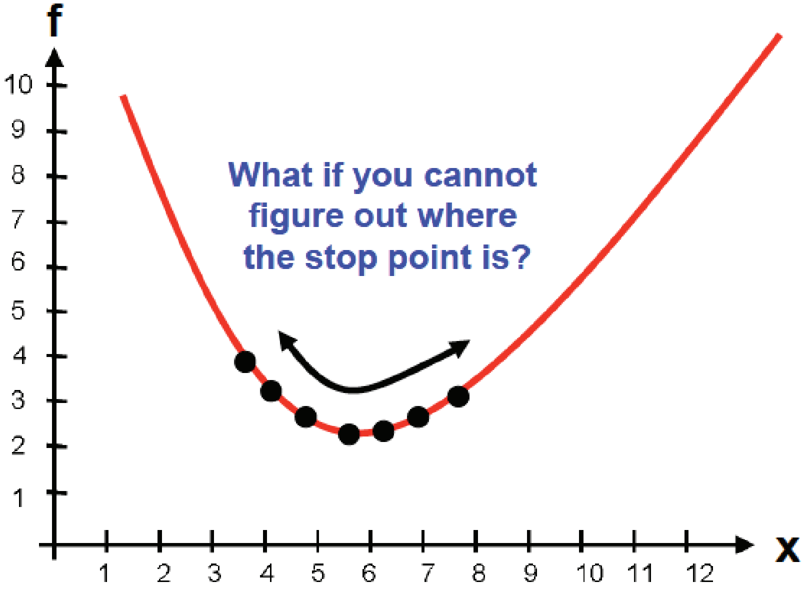
# Problem 1: Non-convex function



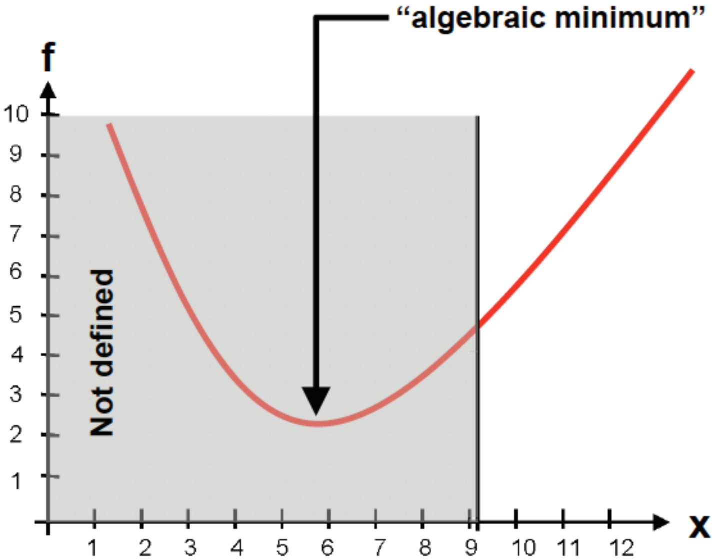
# Problem 1: Non-convex function



# Problem 2: How to stop?



# Problem 3: How to hit a wall?



Start with an initial guess

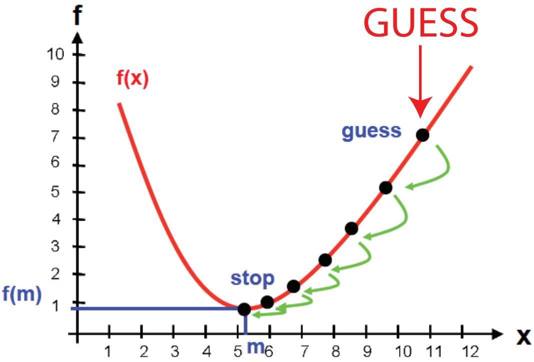
Repeat

– Determine a descent direction

– Choose a step size

– Update

Until stopping criterion is satisfied



# Gradient Descent Algorithm

Start with an initial guess

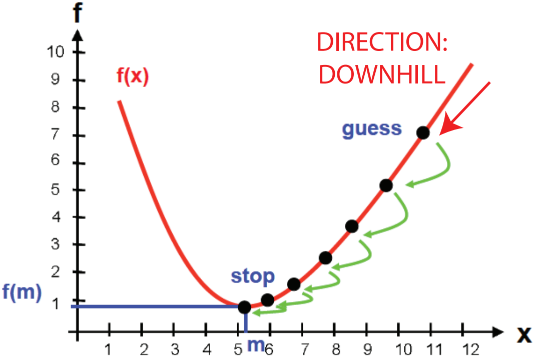
Repeat

– Determine a descent direction

– Choose a step size

– Update

Until stopping criterion is satisfied



Start with an initial guess

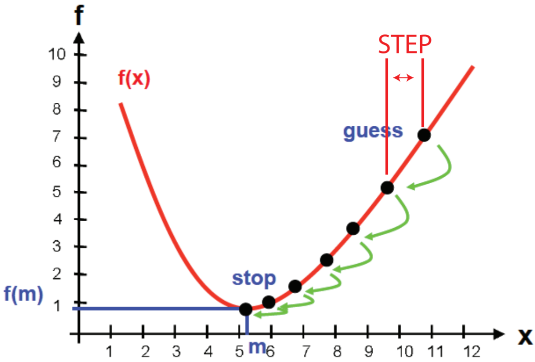
Repeat

– Determine a descent direction

– Choose a step size

– Update

Until stopping criterion is satisfied



Start with an initial guess

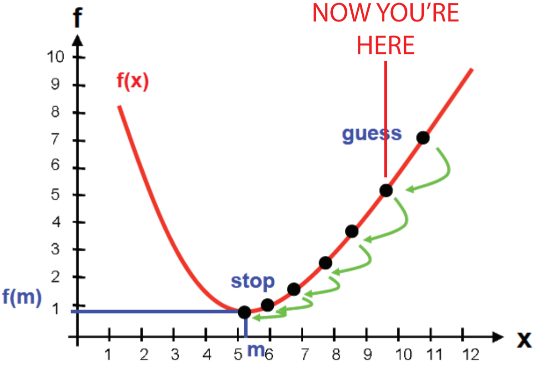
Repeat

– Determine a descent direction

– Choose a step size

– Update

Until stopping criterion is satisfied



Start with an initial guess

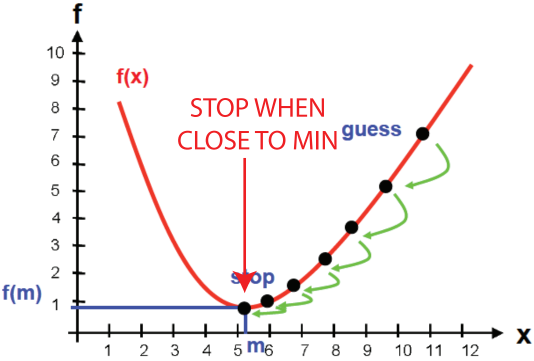
Repeat

– Determine a descent direction

– Choose a step size

– Update

Until stopping criterion is satisfied



# Gradient Descent Algorithm

Start with an initial guess

Repeat

– Determine a descent direction

– Choose a step size

– Update

Until stopping criterion is satisfied

**Psuedo-code:**

guess $x_0$

direction = $-f'(x)$

step = $h > 0$

$x_{k+1} = x_k - h \cdot f'(x_k)$

stop when $f'(x_k) \approx 0$

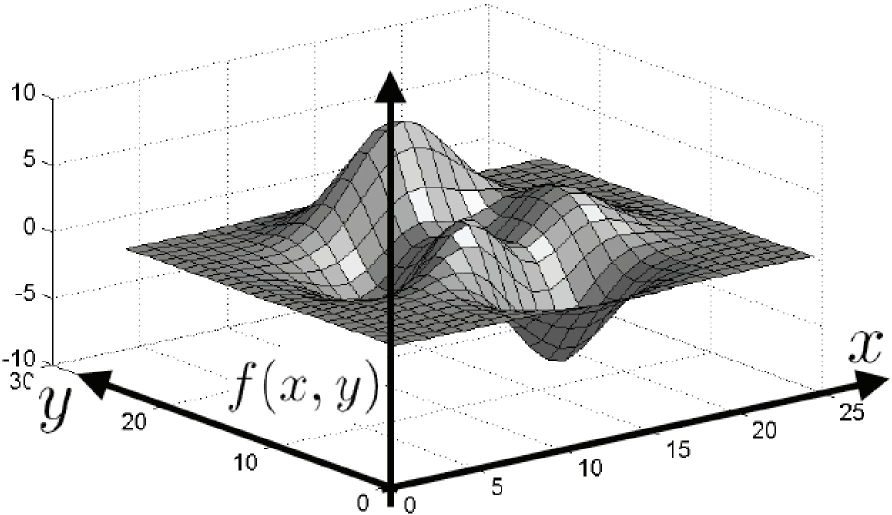
# Gradient in 2D

Definition of gradient in 2D

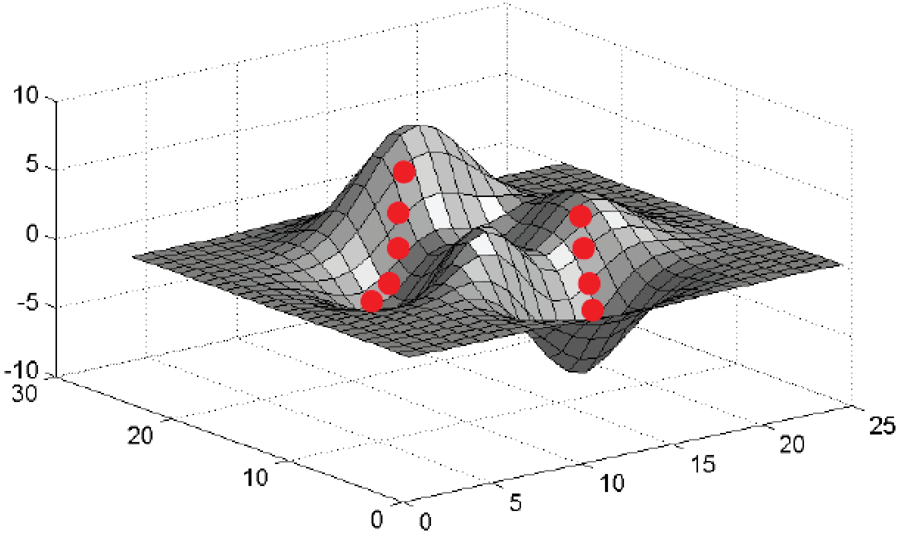
$$\nabla f(x,y) = \begin{bmatrix} \frac{\partial f}{\partial x}(x,y) \\ \frac{\partial f}{\partial y}(x,y) \end{bmatrix}$$

Generalization of derivative in two dimensions

# Gradient Descent in 2D



# Gradient Descent in 2D



# Gradient Descent Algorithm in 2D

Start with an initial guess

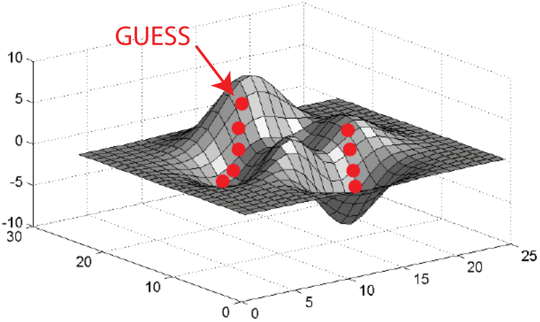
Repeat

– Determine a descent direction

– Choose a step size

– Update

Until stopping criterion is satisfied



# Gradient Descent Algorithm in 2D

Start with an initial guess

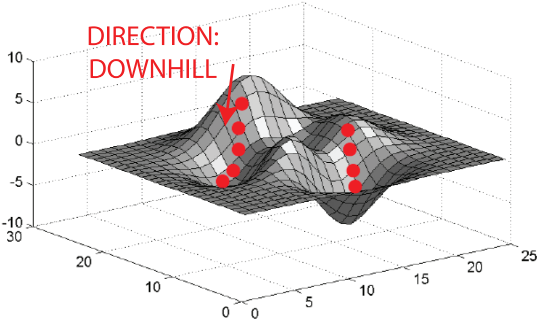
Repeat

– Determine a descent direction

– Choose a step size

– Update

Until stopping criterion is satisfied



# Gradient Descent Algorithm in 2D

Start with an initial guess

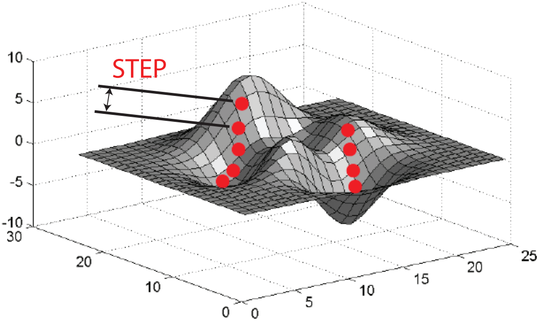
Repeat

– Determine a descent direction

– Choose a step size

– Update

Until stopping criterion is satisfied



# Gradient Descent Algorithm in 2D

Start with an initial guess

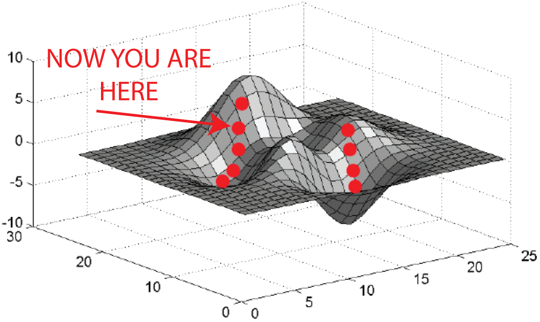
Repeat

– Determine a descent direction

– Choose a step size

– Update

Until stopping criterion is satisfied



# Gradient Descent Algorithm in 2D

Start with an initial guess

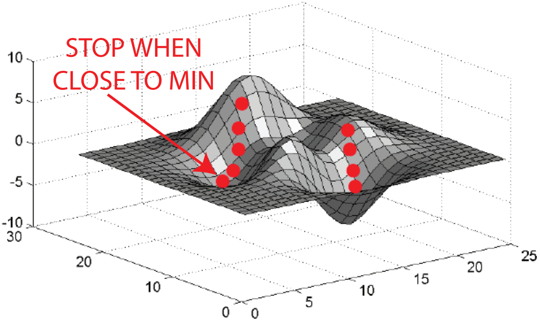
Repeat

– Determine a descent direction

– Choose a step size

– Update

Until stopping criterion is satisfied



# Gradient Descent Algorithm in 2D

Start with an initial guess

Repeat

– Determine a descent direction

– Choose a step size

– Update

Until stopping criterion is satisfied

**Psuedo-code:**

guess $x_0$

direction = $-\nabla f(x)$

step = $h > 0$

$x_{k+1} = x_k - h \cdot \nabla f(x_k)$

stop when $\nabla f(x_k) \approx 0$

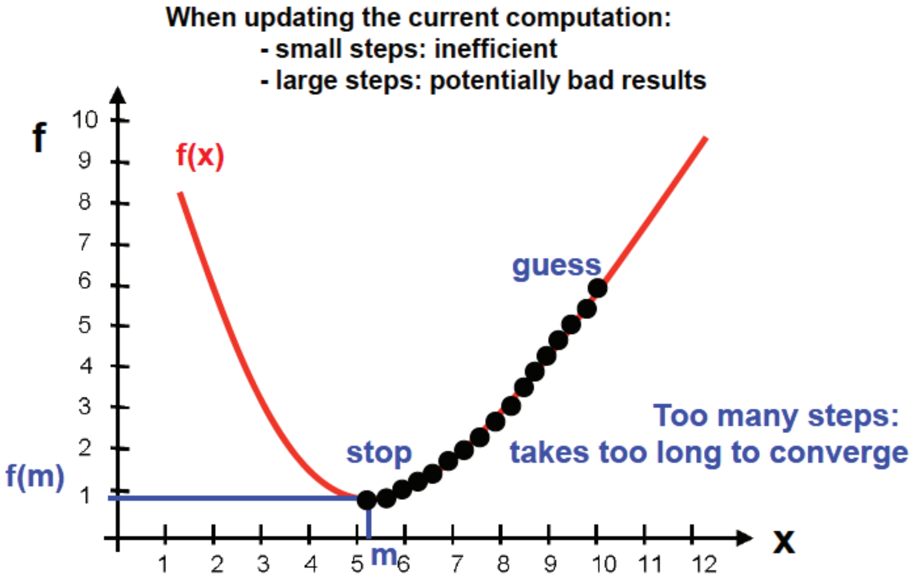
# Gradient in Multiple Dimensions

Can be generalized to multiple dimensions

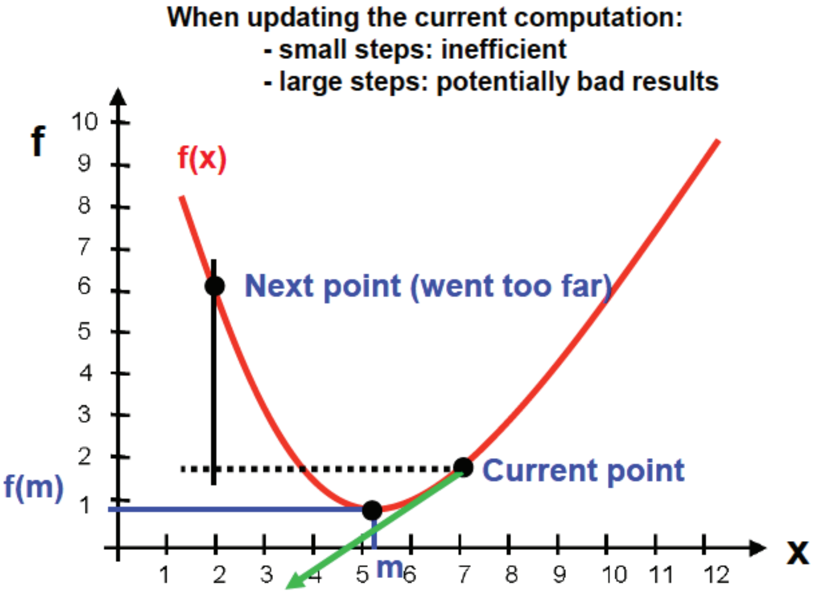
$$\nabla f(x_1, x_2, \cdots, x_N) = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_N}\right]$$

Generalization of derivative in two dimensions

# Problem 1: Choice of Step

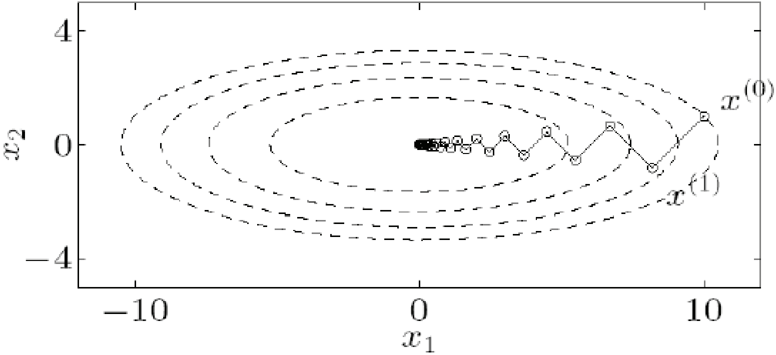


# Problem 1: Choice of Step



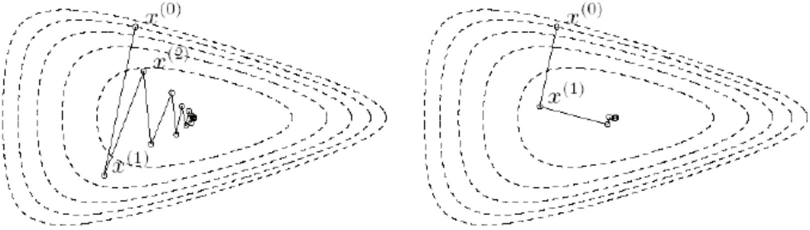
# Problem 2: Ping pong effect

$$f(x_1, x_2) = \frac{1}{2}\left(x_1^2 + \gamma x_2^2\right), \qquad \gamma > 0$$



# Problem 2: Ping pong effect

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




backtracking line search

exact line search

More info: Section 9.2 of Boyd's CVX Textbook

# Problem 3: Stopping Criterion

One intuitive criterion

$$|f'(x_k)| \leq \epsilon, \qquad \text{for some small } \epsilon > 0$$

Or in multiple (e.g. $N$) dimensions

$$\|\nabla f(x_k)\|_2 \leq \epsilon, \qquad \text{for some small } \epsilon > 0$$

where the 2-norm, expanded, is

$$\|\nabla f(x_k)\|_2 = \sqrt{\sum_{i=1}^{N} \left[\frac{\partial f}{\partial x_i}(x_k)\right]^2} = \sqrt{\left[\frac{\partial f}{\partial x_1}(x_k)\right]^2 + \left[\frac{\partial f}{\partial x_2}(x_k)\right]^2 \cdots \left[\frac{\partial f}{\partial x_N}(x_k)\right]^2} \leq \epsilon$$

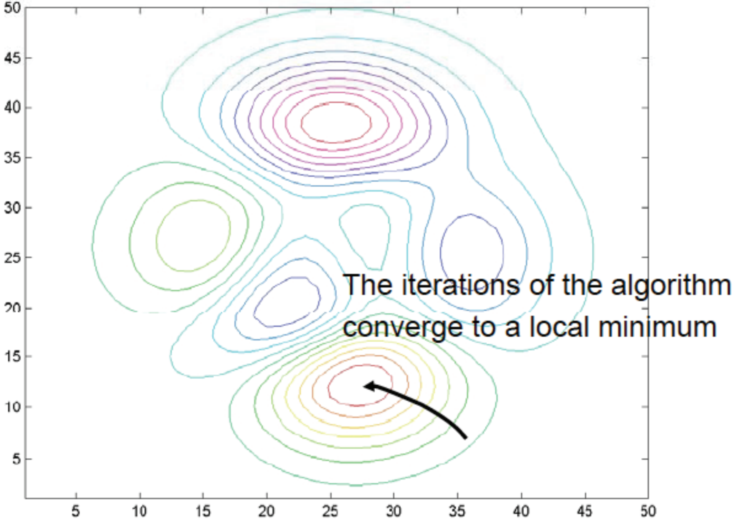
# Some Solutions

Several methods exist to alleviate these three problems

- Linear search methods (e.g. backtracking or exact line search)
- Normalized Steepest Descent
- Newton steps

**Fundamental Problem:** Local Minima

# Example of Local Minima



# Additional Reading

Boyd & Vandenberghe, Chapter 9

Papalambros & Wilde, Section 4.5, 4.6

EE 227 - Convex Optimization with Prof. El Gaouhi