Search in continuous space:

Gradient, Newton-Raphson, convexity

Xiaojin Zhu

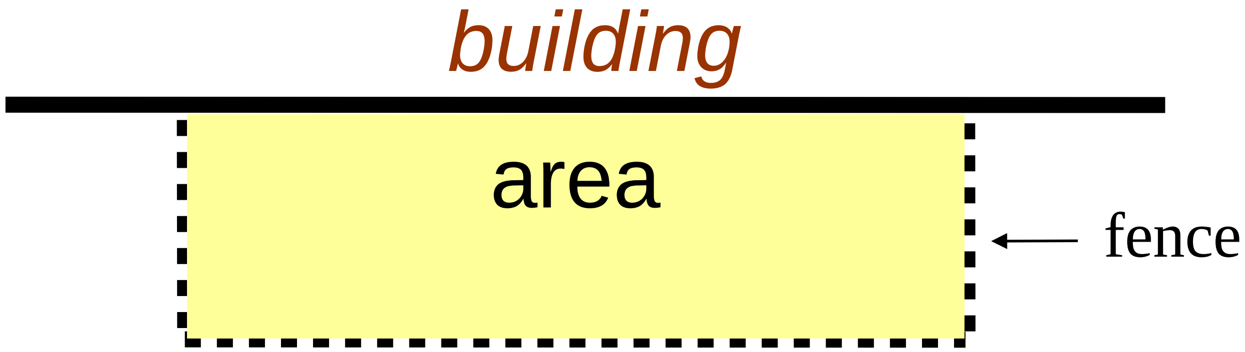
jerryzhu@cs.wisc.edu

Computer Sciences Department

University of Wisconsin, Madison

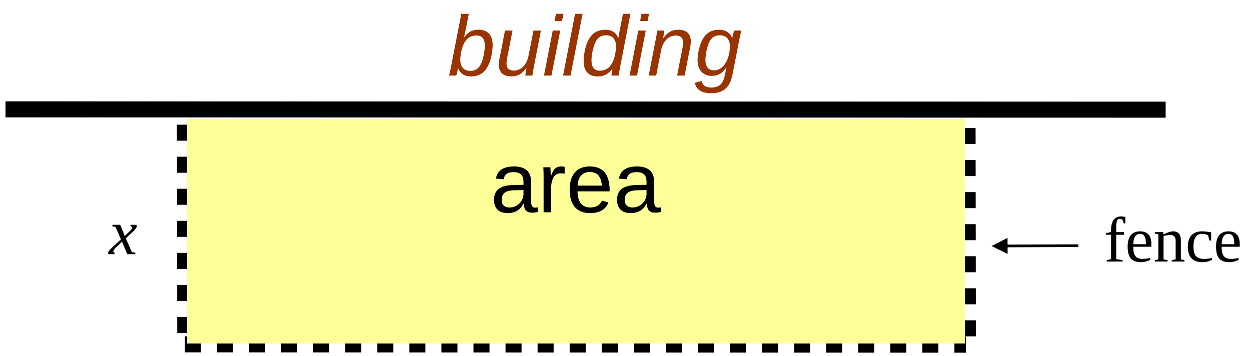
Optimization

- 100m fence, want to maximize area



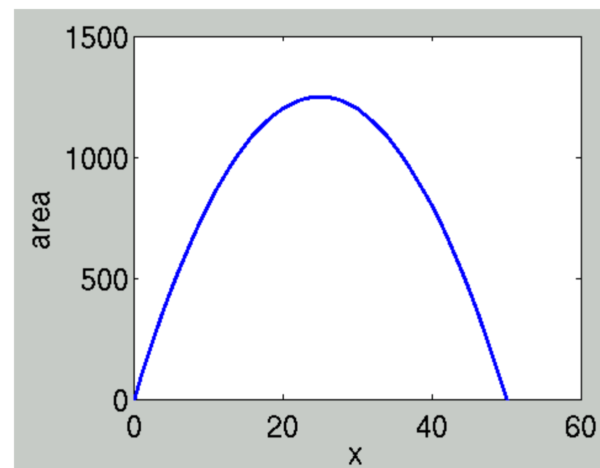
Optimization

- 100m fence, want to maximize area



$$f(x) = x(100 - 2x) = -2x^2 + 100x$$

$$\frac{\partial f(x)}{\partial x} = -4x + 100$$



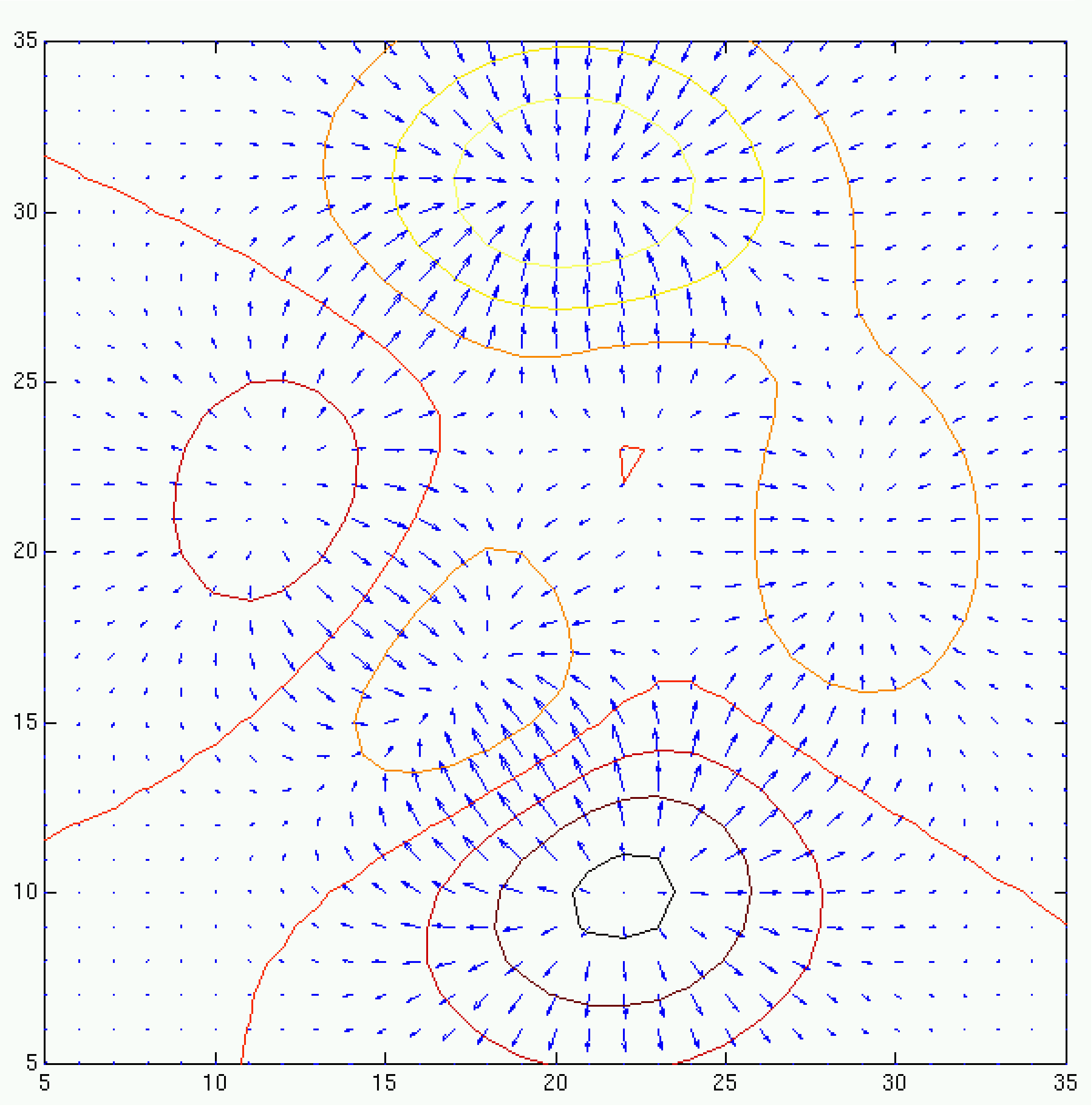
Continuous space

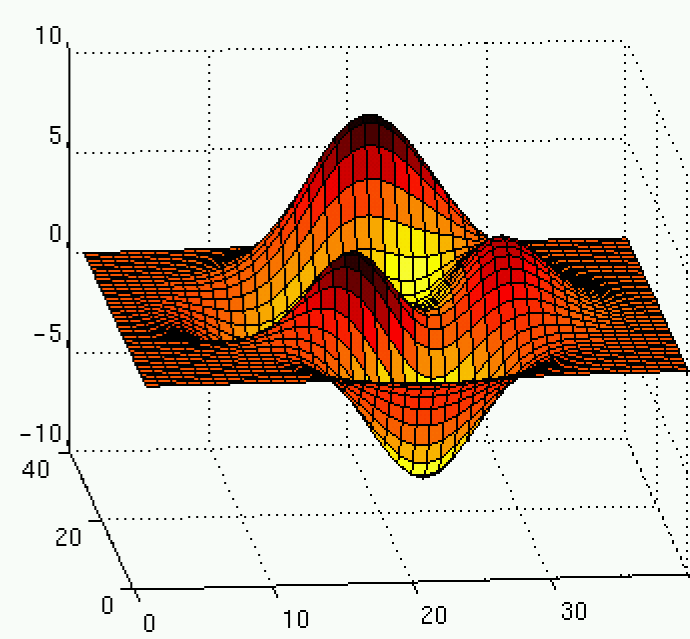
- Find state $x=(x_1, x_2, \ldots, x_m) \in R^m$ that minimizes $f(x)$
- The (partial) derivative $\frac{\partial f}{\partial x_i}$
- The gradient is the vector

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_m} \right)$$

- The gradient points to “higher ground” in f

Gradient





Where gradient vanishes...

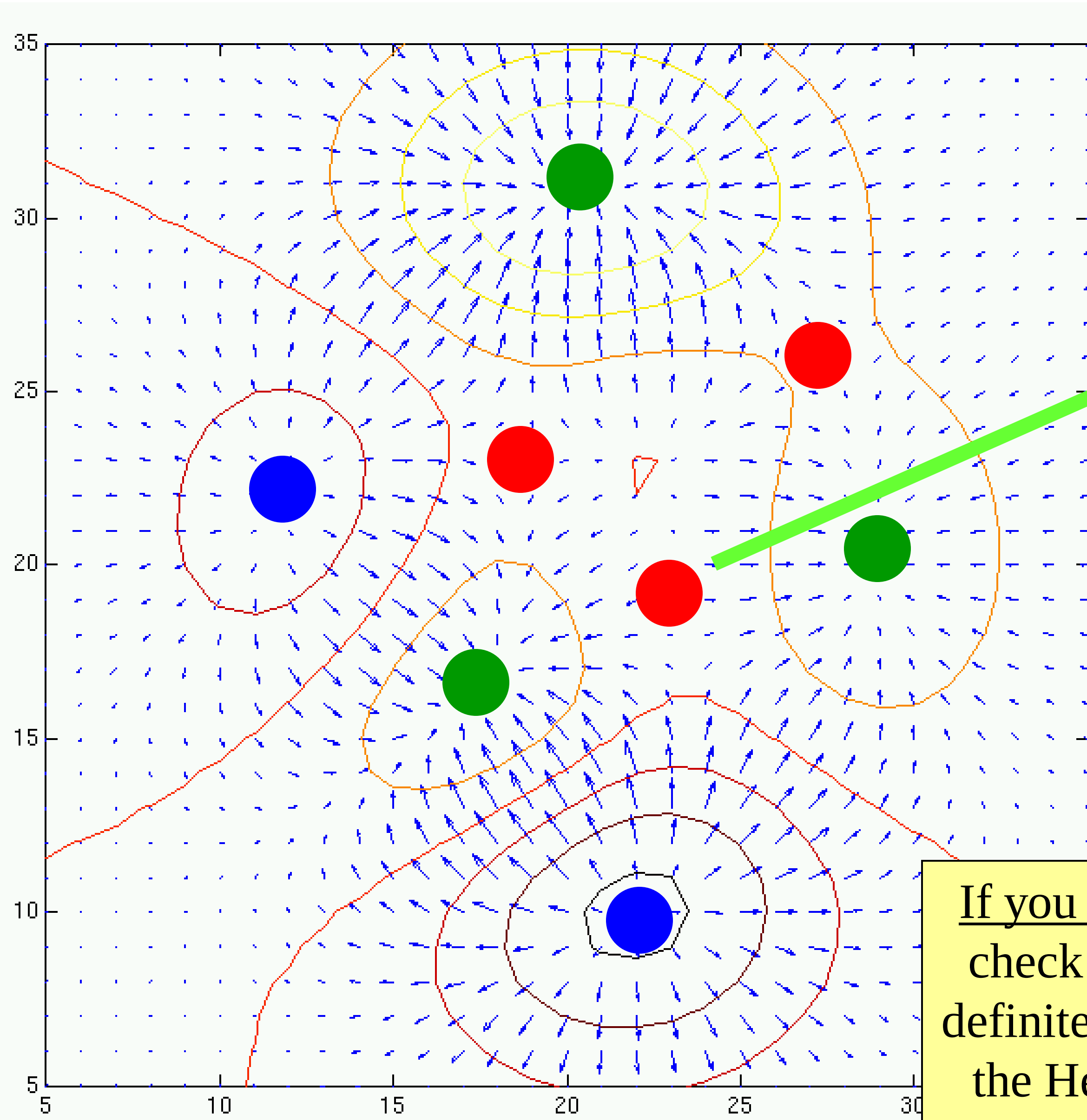
- Find state $x=(x_1, x_2, \ldots, x_m) \in R^m$ that minimizes $f(x)$

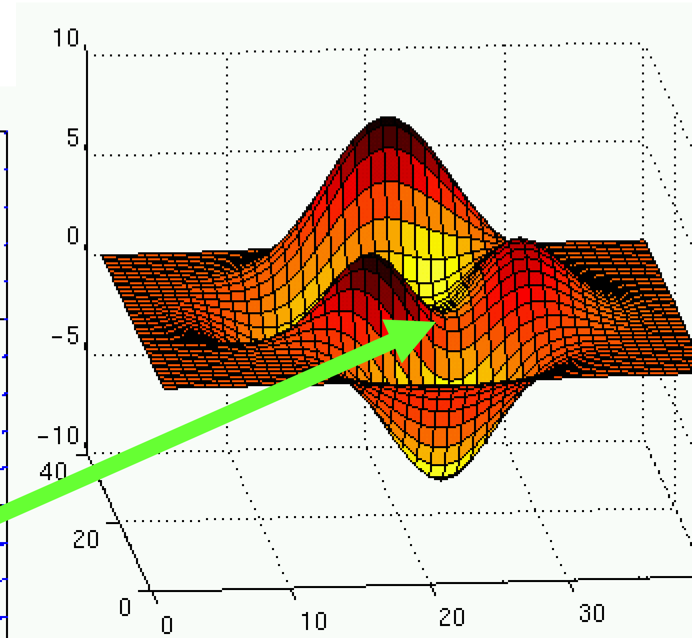
$$\Rightarrow \nabla f = 0$$

(= find x where the gradient disappears)

- Then you have to check whether it is a minimum or maximum, or saddle point

$\nabla f = 0$





Maximum
Minimum
Saddle point

<u>If you know</u>: check semi-definiteness of the Hessian

<u>If you don't</u>: check blah-blah of the blah

Gradient descent

- Often can't solve $\nabla f = 0$ in closed form
- But we can compute ∇f at any point
- **Heuristic**: move along the gradient in a small step

$$x \leftarrow x - \alpha \nabla f(x)$$

  - α is the "step size"
    - Too small: very slow
    - Too large: overshoot
    - Ideas?
- Analogous to hill climbing, finds local optimum.

Gradient descent without a gradient

- Sometimes can't compute ∇f
- e.g. $f(x)$ is returned by some black-box.
- Simulate the gradient at x (empirical gradient)

$$x = (x_1, x_2, \ldots x_m)$$

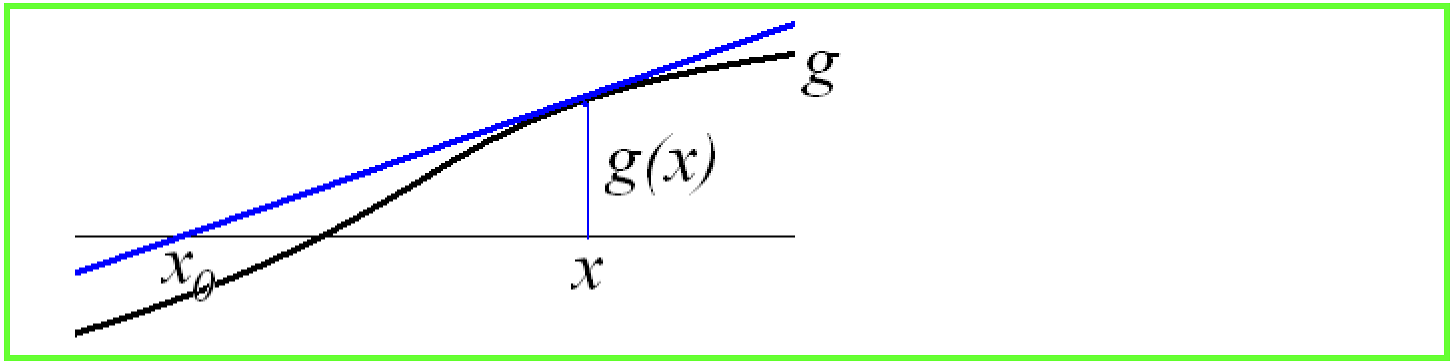
$$x' = (x_1 + \varepsilon, x_2, \ldots x_m)$$

$$\frac{\partial f}{\partial x_1} \approx \frac{f(x') - f(x)}{\varepsilon}$$

Similarly for $x_2 \ldots x_m$

Simple Newton-Raphson in 1-D

- A smart way to choose step size α
- Find roots $g(x)=0$
- To find min or max of $f(x)$, work on $g(x)=f'(x)$
- Assume near linearity of $g()$
  - $g(x) \approx g(x_0) + (x-x_0)\, g'(x_0)$ (1st order Taylor)
  - $g(x) \approx g(x_0) + (x-x_0)\, g'(x)$ (assumed near linear)
  - $x_0 \approx x - g(x) / g'(x)$
  - Make it iterative: $x \leftarrow x_0$
- Can overshoot; Finds local optimum.



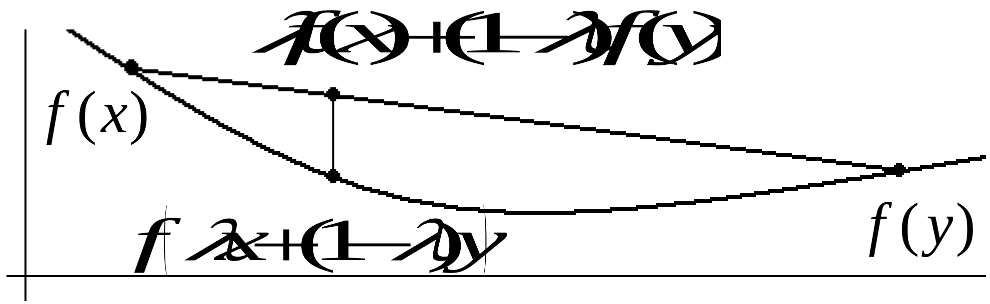
Convexity

- How nice it would be

  if there is one and only one minimum.

- It's so if *f*() is convex.

$$f(\lambda x + (1-\lambda) y) \le \lambda f(x) + (1-\lambda) f(y)$$



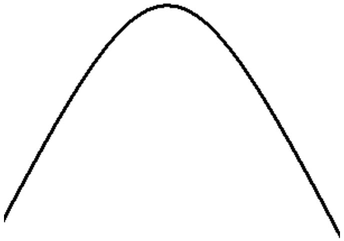
- In 1-D: $f''(x) > 0$ for all x. Example: $f(x)=x^2$
- In high-D: the Hessian at all *x* is positive semi-definite
- Even gradient descent will find the global minimum*

* with small α

Convexity

- A concave function has a global maximum
  - An upside-down convex function

How to remember:
con**cave** is like a
cave



- Much research
- Spend time to formulate your problem as a convex function!