CS101: Intro to Computing
Fall 2015

Lecture 23

Administrivia

- Homework 14 ***after*** break (enjoy break!)
  - The last homework!
  - Coming Monday, Nov 30th (at the latest)
  - Due on the last day of class
- Final exam
  - December 15th 1:30pm-4:30pm (here)
  - Get approval for the conflict (email me)

REVIEW

Are you here?

a) Yes.
b) No.
c) Maybe.
d) I like turtles.

Course Summary (so far…)

1. Python fundamentals
2. Data wrangling
3. Data visualization
4. Simulation
5. Random processes
6. Optimization

DISCRETE OPTIMIZATION

Combinatorial Optimization

- Finite, but large number of solutions
  - e.g. Suitcase problem
- Methods:
  - Brute force
  - Random search
  - Global greedy search
  - Hill climbing (local greedy search)

Brute-force search

- Search through the ***entire*** domain of f
- Also called an “exhaustive search”

Random search

- Randomly sample the domain of f
- Might not find the *true* optimum
- Might find one that is “good enough”
- How can we sample from the domain of our problem?

Greedy Search

- Ignore the overall problem
- Pick the "best" partial solutions individually, one by one
- Keep picking until we can't improve anymore
- Need a heuristic for "best" items in our problem

Hill Climbing

- Keep track of our “current” solution
- Change one element of the solution
- If change improves the solution, this is new “current” solution
- Repeat until we can’t improve anymore

Steepest Ascent Hill Climbing

- Keep track of our “current” solution
- Incrementally change ***all*** elements of the solution
- Pick the change with best improvement
  - “Steepest” slope
- If change improves the solution, this is new “current” solution
- Repeat until we can’t improve anymore

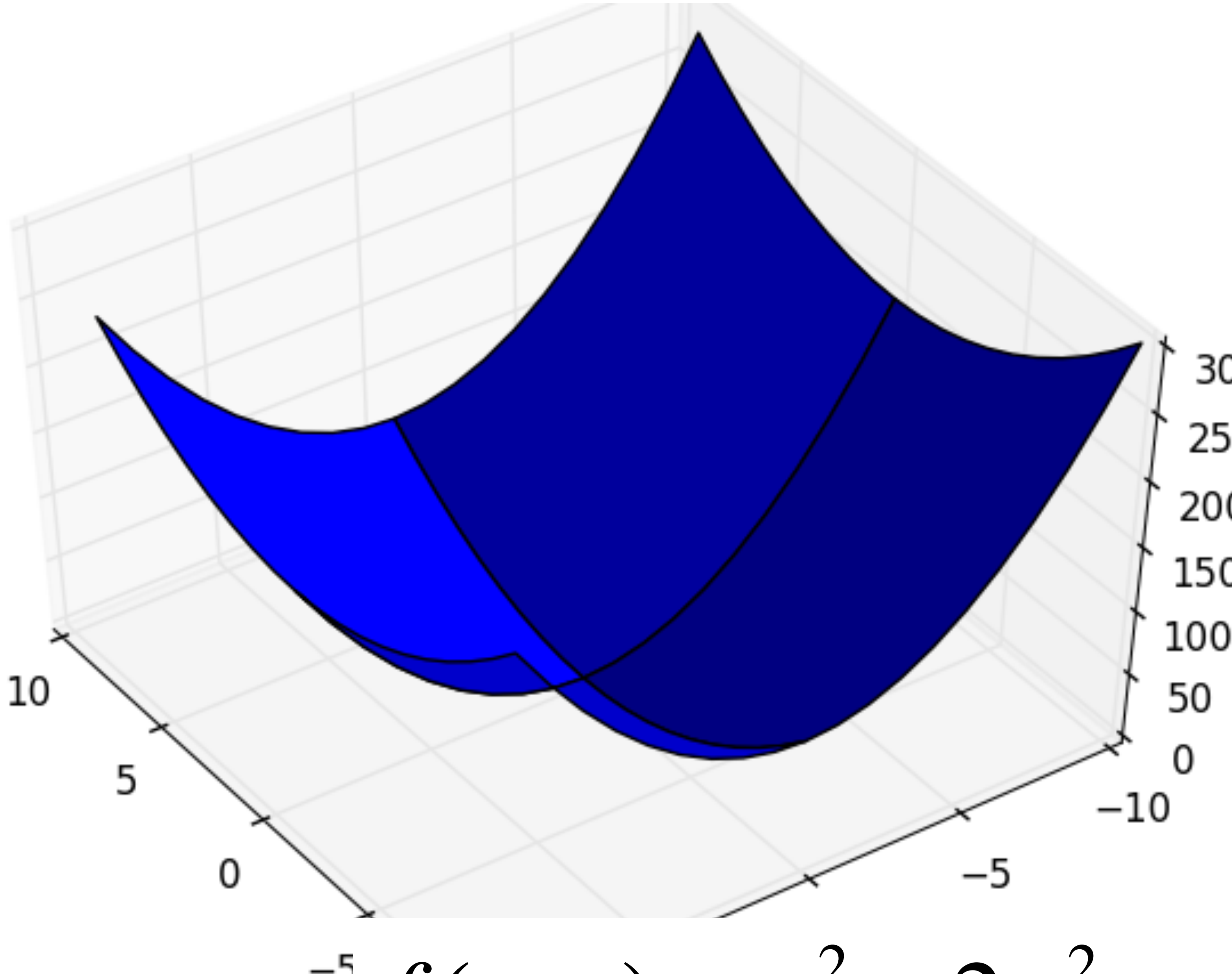
CONTINUOUS OPTIMIZATION

Continuous Optimization

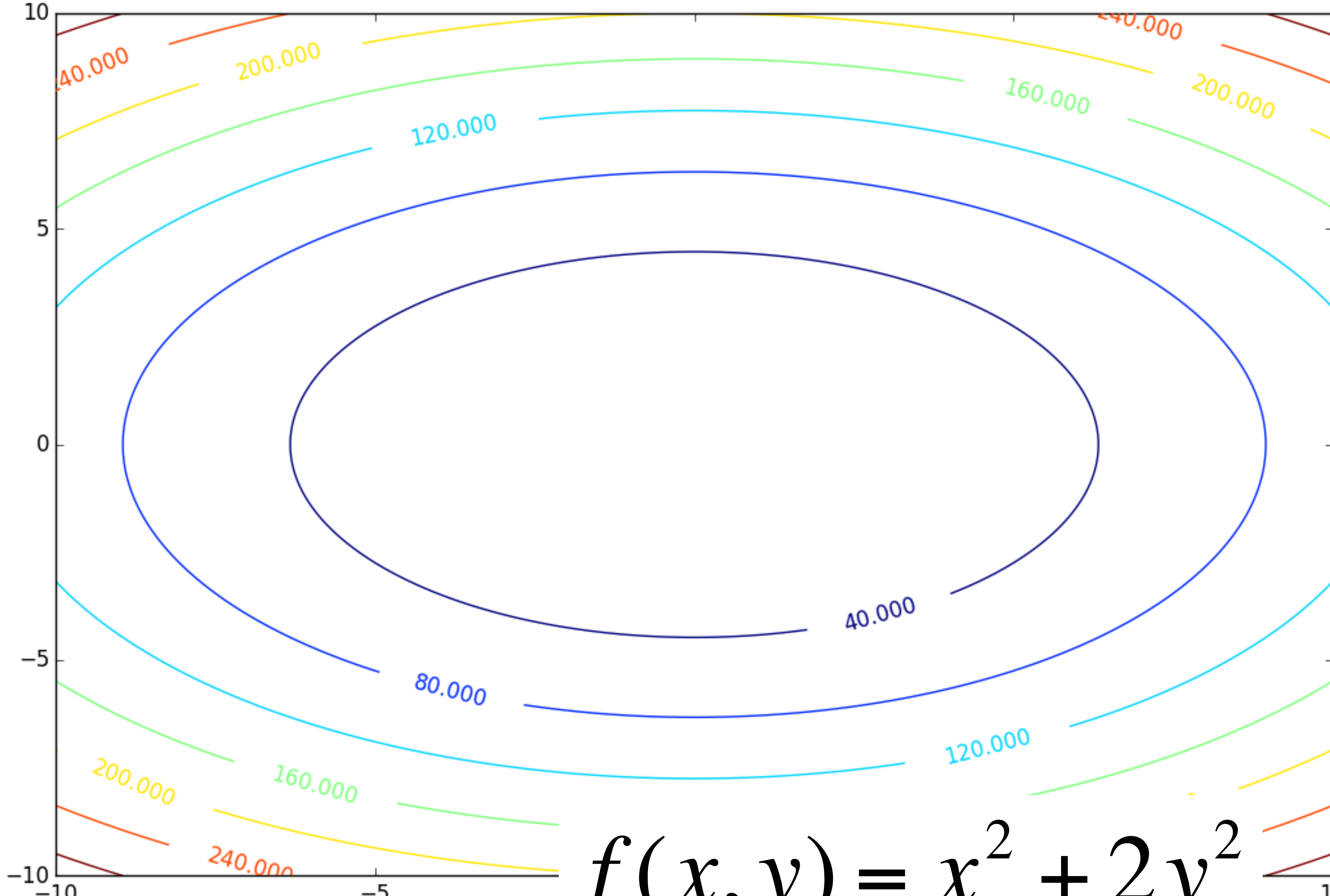
- Goal of optimization: given a function f(x), find x such that f(x) is minimized
- What about functions like this?

$$f(x, y) = x^2 + 2y^2$$

- We could (and ***did*** in HW13) turn it into a discrete problem



$$f(x,y) = x^2 + 2y^2$$



$$f(x, y) = x^2 + 2y^2$$

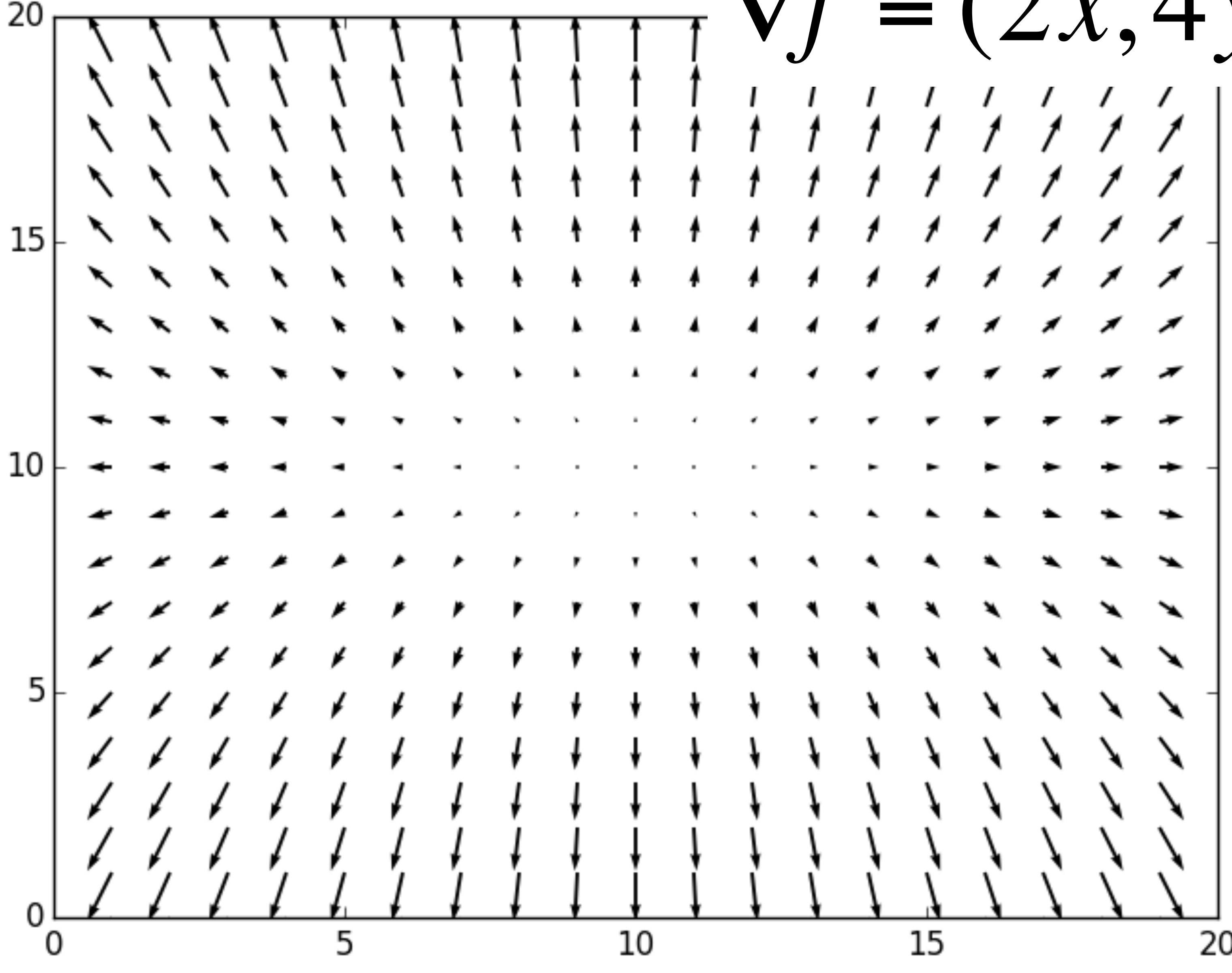
Gradient

- Points in the direction of greatest rate of increase of a function
- Partial derivative of f in each dimension

$$f(x, y) = x^2 + 2y^2$$

$$\nabla f = (2x, 4y)$$

$$\nabla f = (2x, 4y)$$



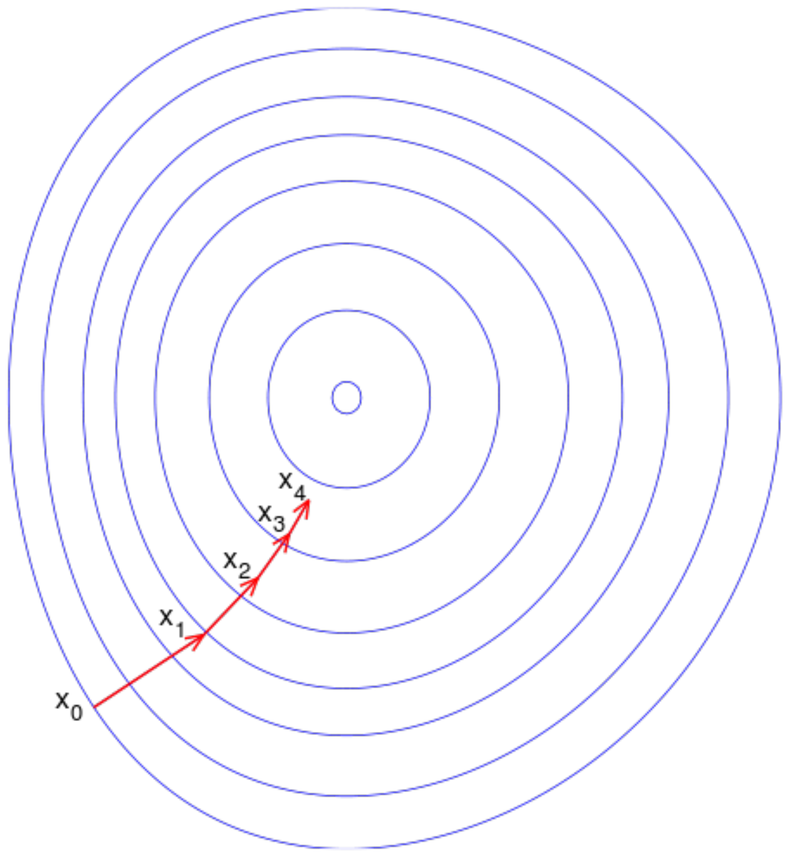
Gradient Descent

- A function decreases ***fastest*** along its negative gradient
- Follow the gradient by a ***small*** amount to gradually approach the minimum

$$x_{new}, y_{new} =$$

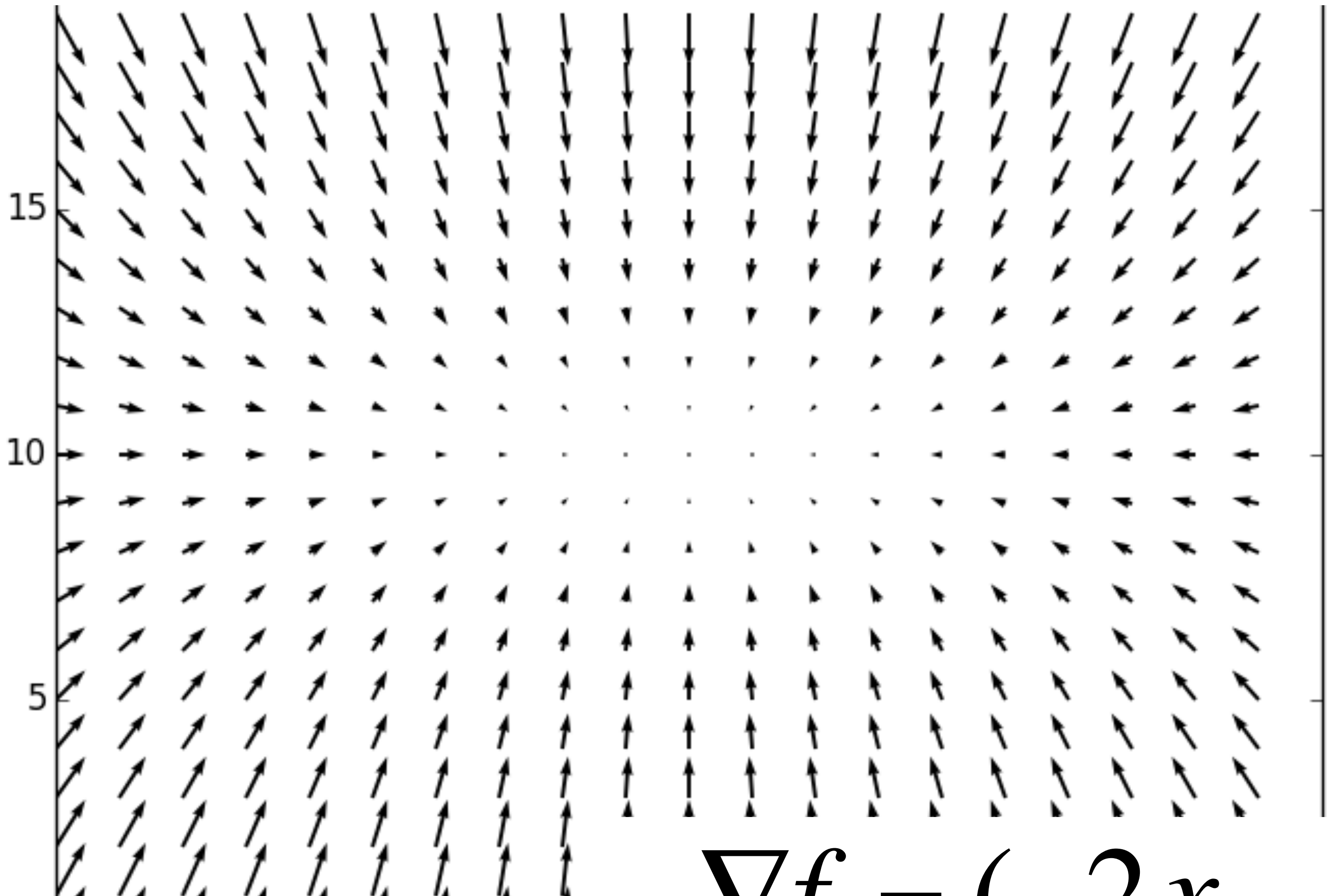
$$x_{old}, y_{old} - \gamma \nabla f(x_{old}, y_{old})$$

Gradient Descent



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

Gradient Descent



$$-\nabla f = (-2x, -4y)$$

Gradient Descent

```
cx,cy=10.0,10.0   # current x and y
gamma=.01         # rate of movement

for i in range(1000):
    cx-=gamma*2*(cx)
    cy-=gamma*4*(cy)
print cx,cy
```

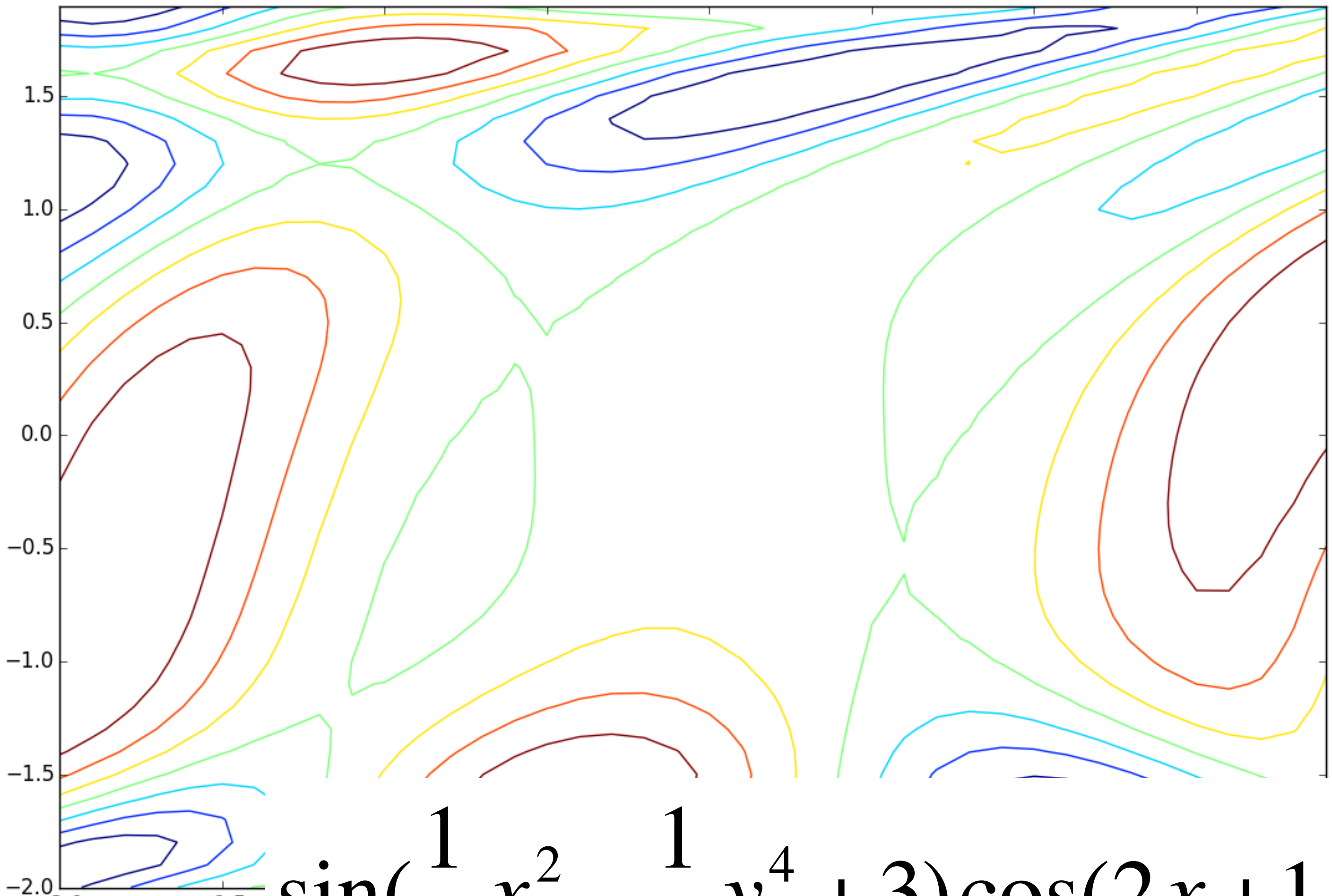
Gradient Descent

```
cx,cy=10.0,10.0   # current x and y
gamma=.01         # rate of movement
epsilon=.00001
while improvement<epsilon:
    ox,oy=cx,cy
    cx-=gamma*grad_fx(cx)
    cy-=gamma*grad_fy(cy)
    improvement=distance((cx,cy),(ox,oy))
```

Limitations

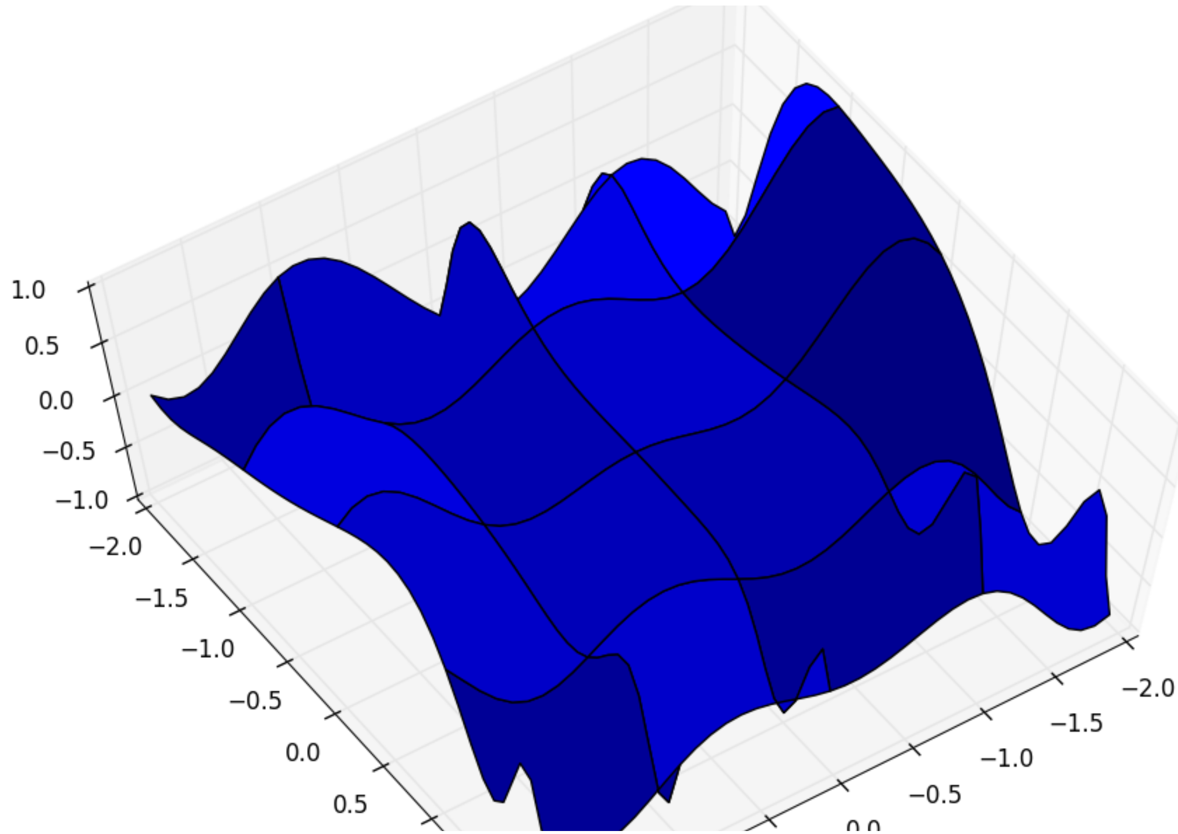
- Surface must be convex
- Can’t use it if we can’t compute gradient
- Can be slow to converge

Limitations



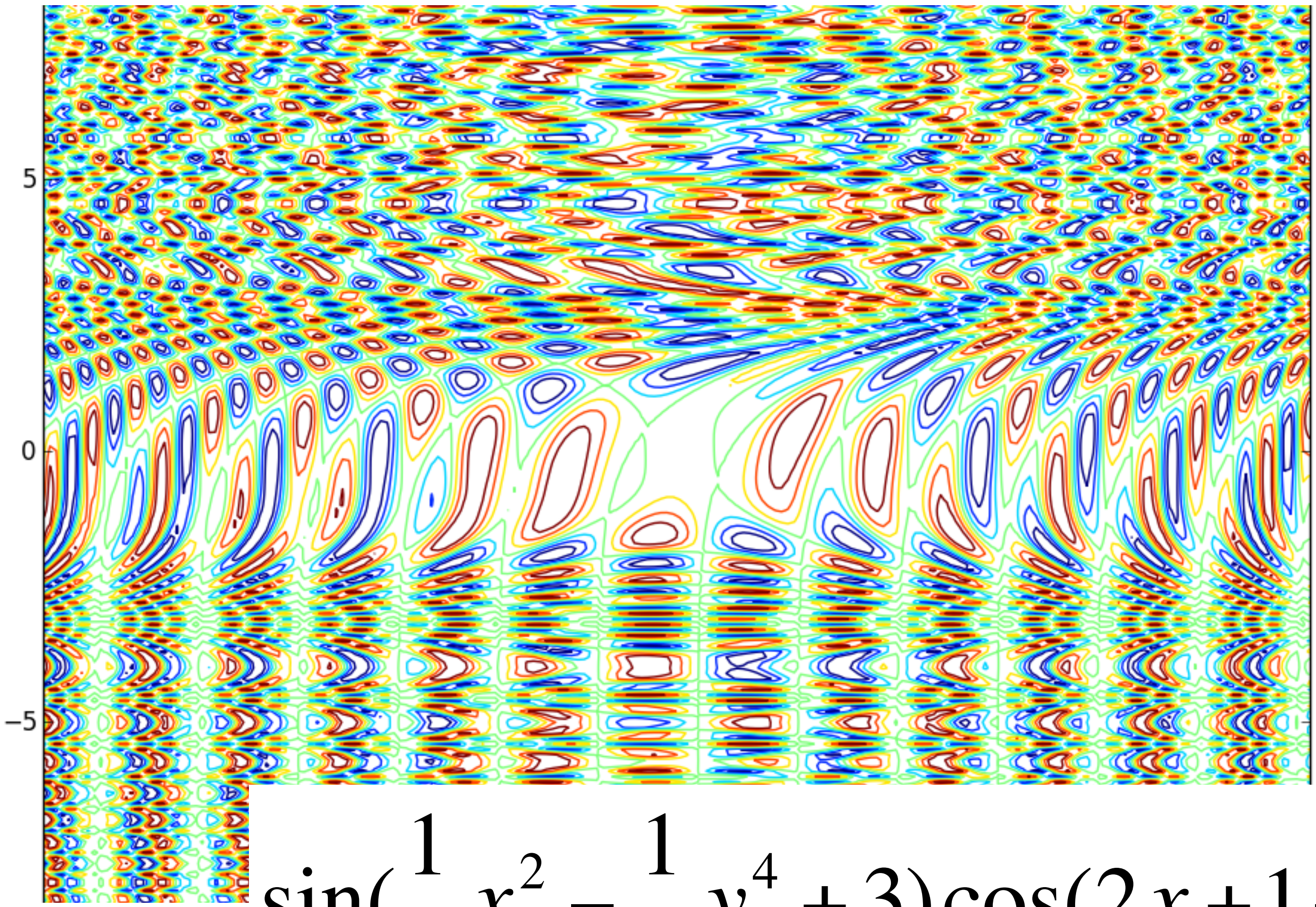
$$\sin\left(\frac{1}{2}x^2 - \frac{1}{4}y^4 + 3\right)\cos(2x + 1 - e^y)$$

Limitations



$$\sin(\frac{1}{2}x^2 - \frac{1}{4}y^4 + 3)\cos(2x + 1 - e^y)$$

Limitations



$$\sin(\frac{1}{2}x^2 - \frac{1}{4}y^4 + 3)\cos(2x + 1 - e^y)$$