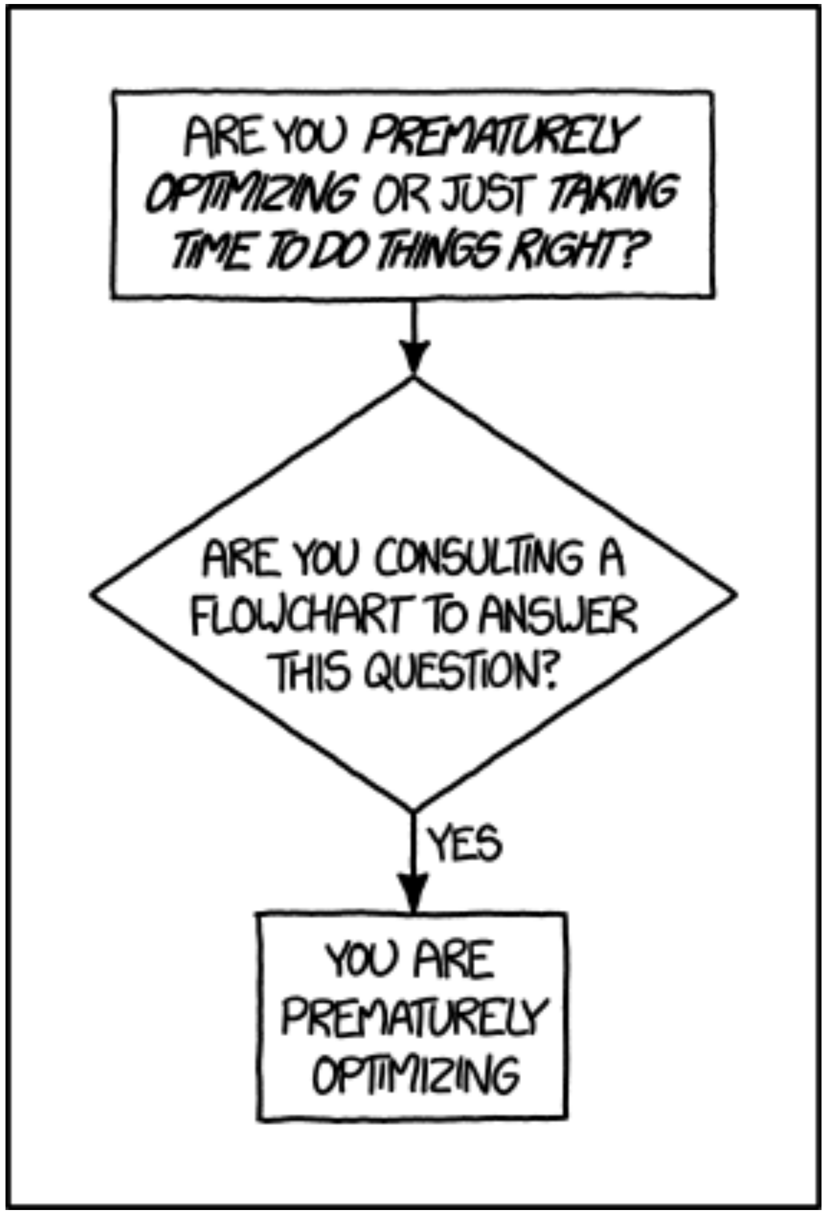
Applications of Mathematics in Computer Science (MACS)

LECTURE #4: ALGORITHMIC DIFFERENTIATION



The CS Problem:

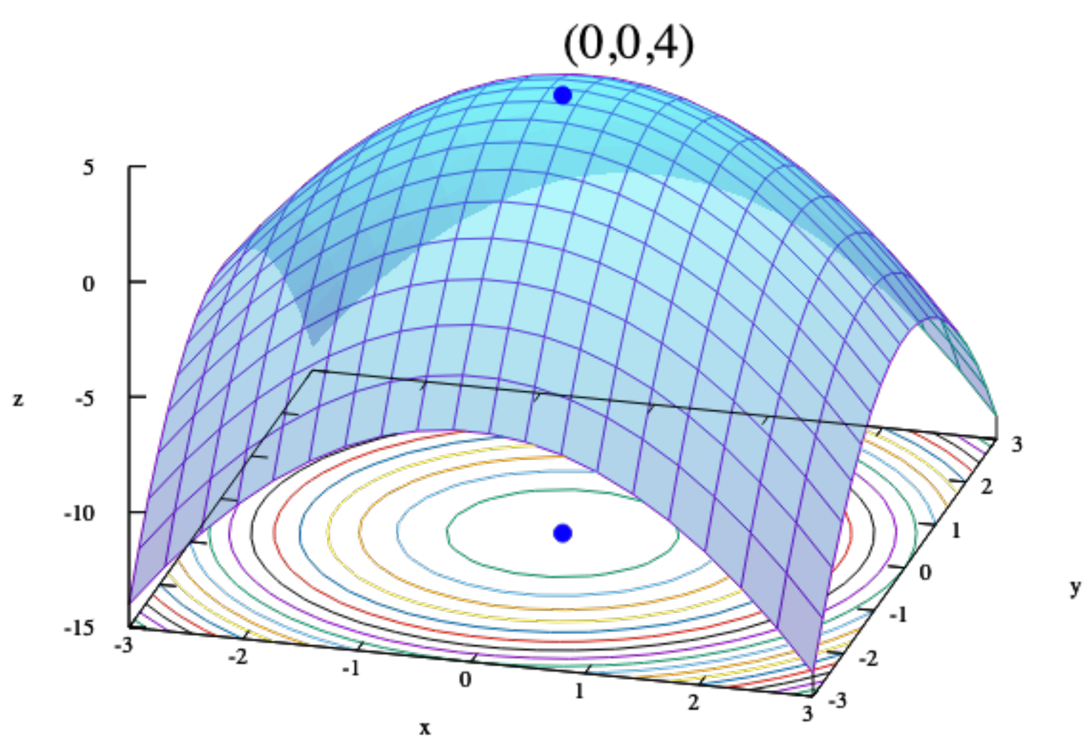
Optimization: finding the best solution

- shortest path in a graph
- Most economical packaging
- Most effective vaccination strategy

Optimization

- Given: a function $f : A \to \mathbb{R}$
- Find: $x_0 \in A$ such that:
  - $f(x_0) \leq f(x)$ for all $x \in A$ (minimization)

  or
  - $f(x) \leq f(x_0)$ for all $x \in A$ (maximization)



$$z = -(x^2 + y^2) + 4$$

$$\max_{(x,y)} z = (0,0)$$

How to solve an optimization problem?

Stochastic optimization

try many values randomly

How to solve an optimization problem?

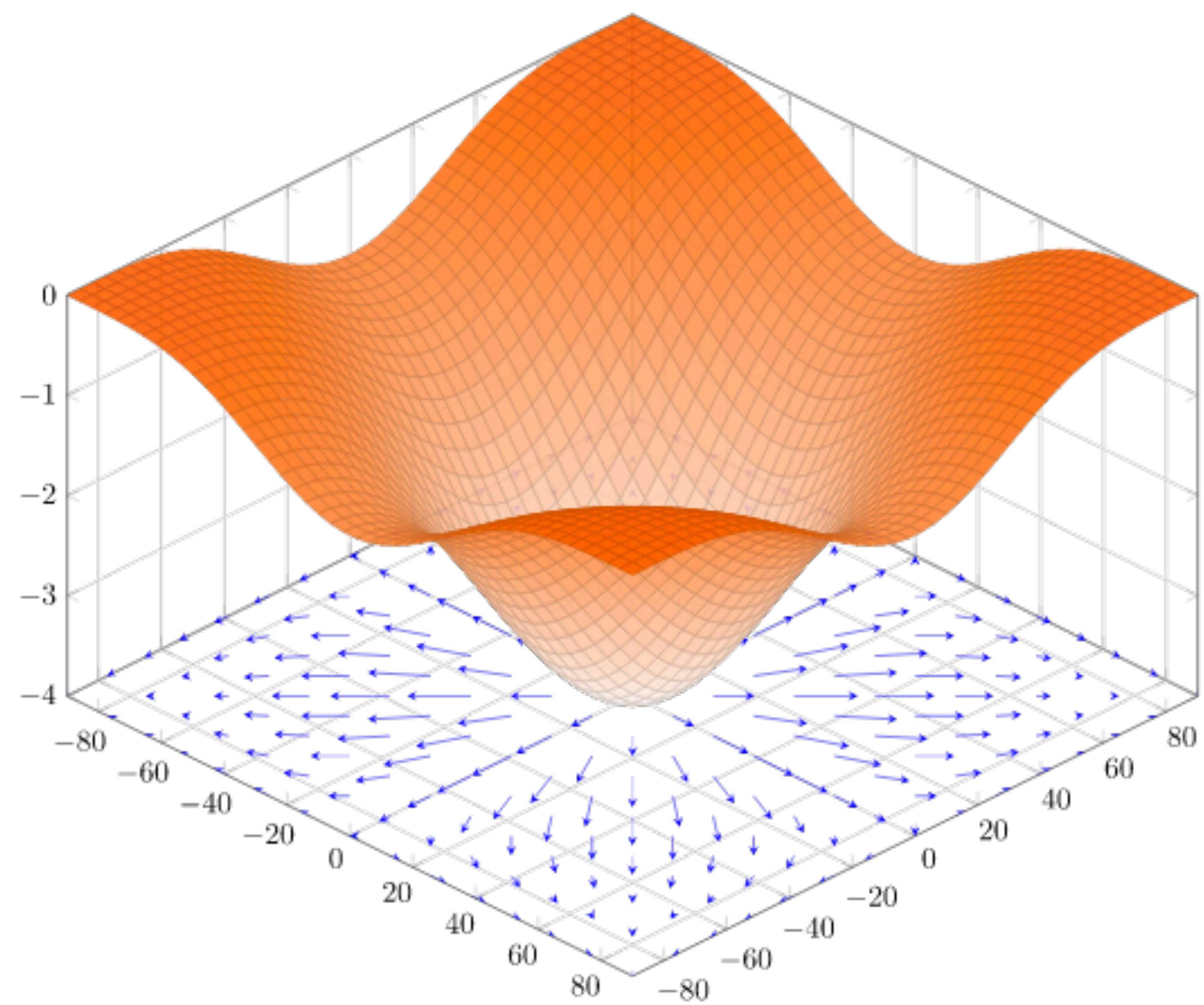
Gradient-based optimization

Differentiation based

Gradient:

direction of fastest ascent

$$\nabla f = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \end{pmatrix}$$



How to solve an optimization problem?

Gradient descent

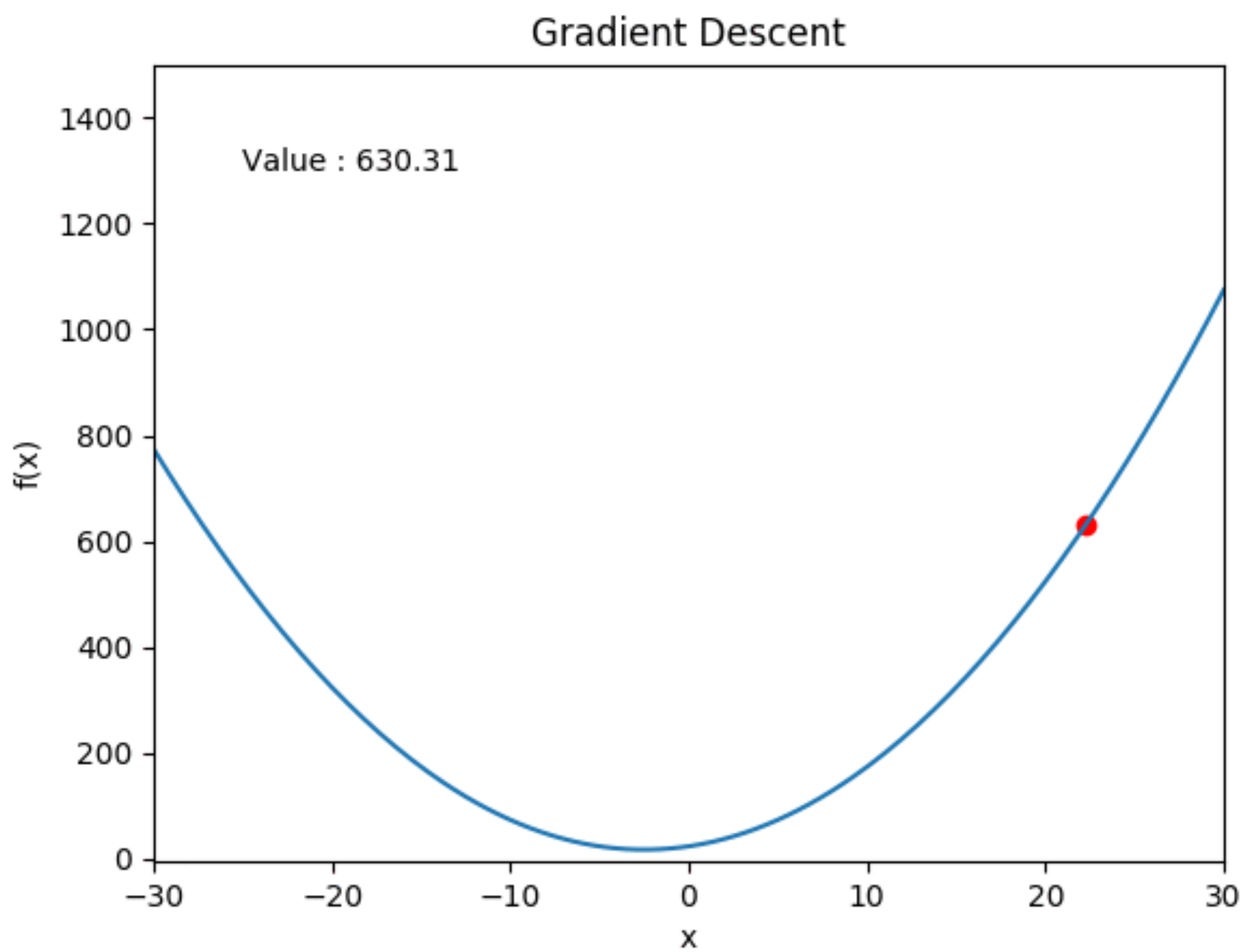
With $\nabla f(x, y, \cdots)$ we can optimize much faster

Repeat:

- $x \leftarrow x - \delta \nabla f(x)$
- $\delta \leftarrow \gamma \delta$

for:

- δ is small
- $\gamma = 1 - \varepsilon$



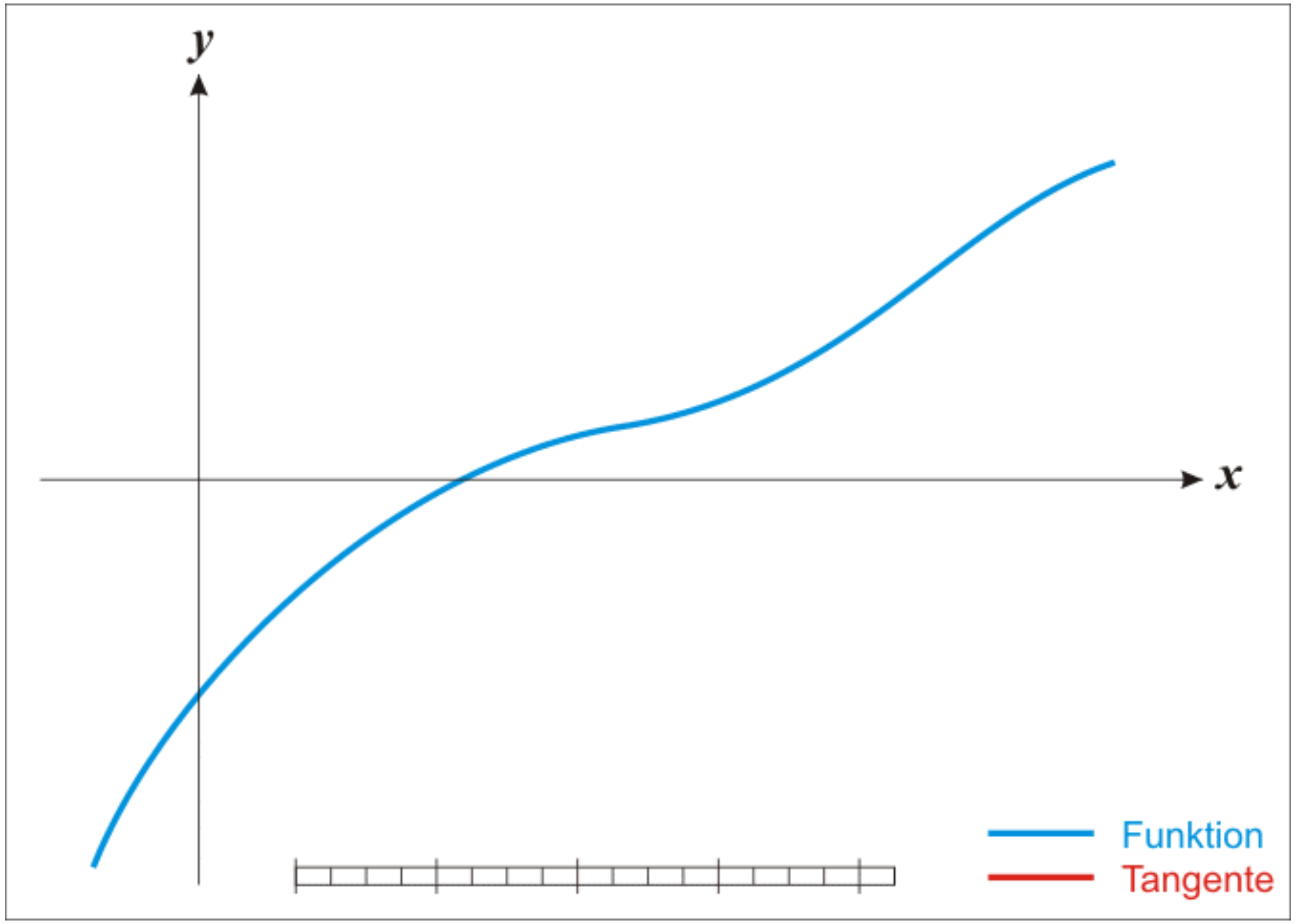
How to solve an optimization problem?

Newton's method

Repeat:

- $$x \leftarrow x - \frac{f'(x)}{f''(x)}$$

converges faster than Gradient Descent



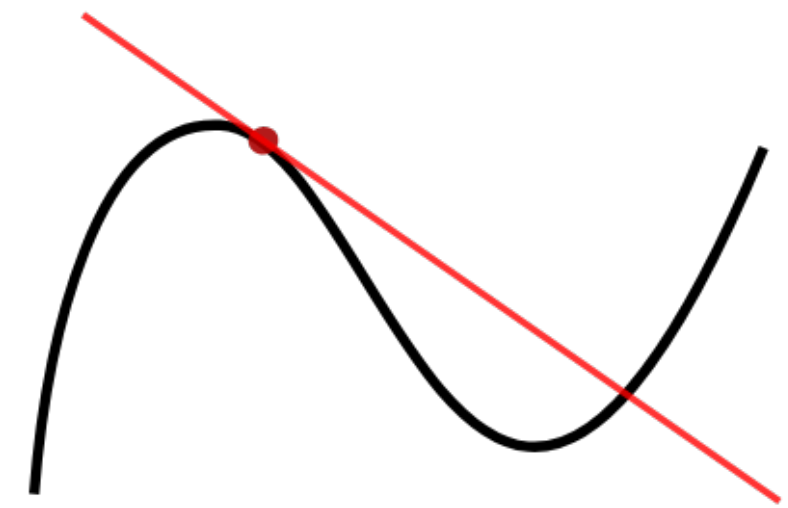
But how do we obtain derivatives?

$$\nabla f(x) \qquad f'(x) \qquad f''(x)$$

The math technique:
Differentiation

Differentiation

Slope of the function's tangent



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

Leibniz's notation: $\dfrac{df}{dx}$

Partial derivatives: $$\frac{\partial f(x_1, x_2, \ldots, x_n)}{\partial x_i}$$

Differentiation Rules

- Constants: if $f(x) = C$ then $f'(x) = 0$
- Sum: $(f(x) + g(x))' = f'(x) + g'(x)$
- Product: $(f(x) \cdot g(x))' = f'(x)g(x) + f(x)g'(x)$
- Quotient: $(\frac{1}{f(x)})' = -\frac{f'(x)}{f(x)^2}$
- Chain rule: $f(g(x))' = f'(g(x))g'(x)$
- Trigo: $\sin(x)' = \cos(x)$, $\cos(x)' = -\sin(x)$

Differentiation Examples

$$3' = 0$$

$$x' = 1$$

$$(x^2)' = (xx)' = x'x + xx' = 2x$$

$$\tan(x)' = \left(\frac{\sin(x)}{\cos(x)}\right)' = \frac{\sin^2(x) + \cos^2(x)}{\cos^2(x)} = \frac{1}{\cos^2(x)}$$

But how do computers differentiate?

Numerical differentiation

1. Choose h
2. Compute $f(x)$
3. Compute $f(x+h)$
4. Return $\dfrac{f(x+h)-f(x)}{h}$

Problems:

h too large — truncation error

h too small — roundoff error

Symbolic differentiation

1. Represent function symbolically.
2. Apply differential rules.

Problems:

Cannot handle complex constructors (loops/conditional/recursion)

Code swell: $\left(\frac{\log(x)+\exp(x)}{\log(x)\exp(x)}\right)' =$

$$-\frac{\exp(-x)(\log(x)+\exp(x))}{\log(x)} - \frac{\exp(-x)(\log(x)+\exp(x))}{x\log^2(x)} + \frac{\exp(-x)(\exp(x)+\frac{1}{x})}{\log(x)}$$

Algorithmic differentiation

- NOT symbolic differentiation
- NOT numerical differentiation
- differentiates ANY code
- Computation costs of $f'(x)$ and $f(x)$ are similar

Algorithmic differentiation

```
def f(a, b):
  c = a*b
  d = sin(c)
  return d
```

```
def f(a, da, b, db):
  c, dc = a*b, da*b + a*db
  d, dd = sin(c), dc * cos(c)
  return d, dd
```

Program trace

f

```
def f(a, b):
   c = a*b
   if c > 0:
      d = log(c)
   else:
      d = sin(c)
   return d
```

f(2, 3)

```
a=2, b=3
c=a*b=6

d=log(c)=1.791


return d=1.791
```

f(2, 3),df(2,3)/da

```
a=2, b=3, da=1, db=0
c=a*b=6,  dc=da*b + a*db=3

d=log(c)=1.791, dd=dc*(1/c)=0.5


return d=1.791, dd=0.5
```

Let's try…