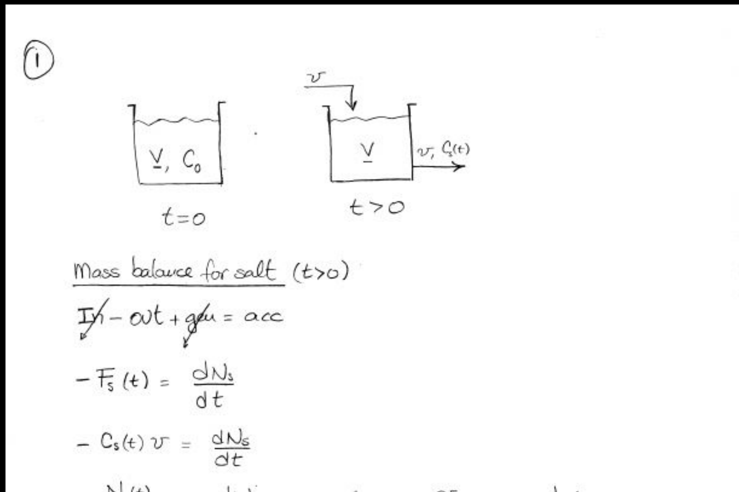
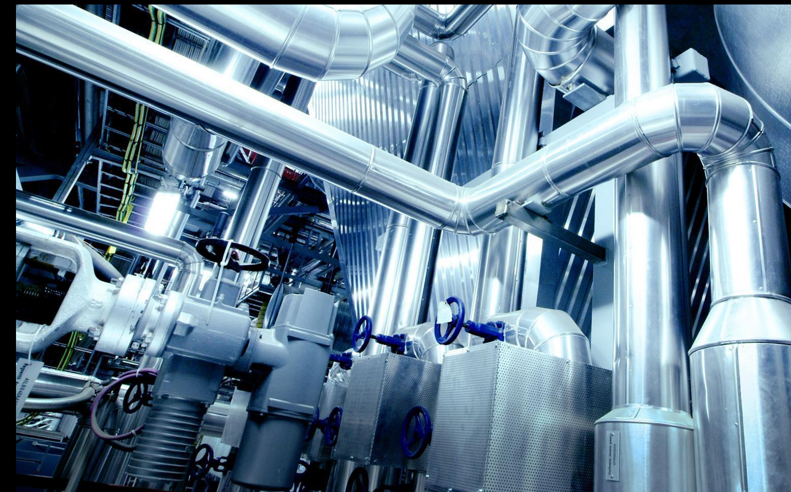




ChE352
Numerical Techniques for Chemical Engineers

Professor Stevenson

Lecture 15

Questions about pseudocode?

- Follows the structure of code
- Intended for human reading, not machine
- Must be **clear** and **logically correct**
- No points off for simple syntax errors
- Example:

```
x = [0, 1, 2]  # comments help
for i in range(0, len(x)):
    print sign(x[i])  # eg +/-
```

The optimal topic to learn

- Many problems in engineering (and life?) can be phrased as optimization:
  - Optimal conditions for a reactor
  - Optimal choice of reactor for a plant
  - Optimal plant for the world economy
  - Optimal studying for your GPA

Other examples?

A friendly objective function

- QALY the lightbulb
- Mascot of EA (Effective Altruism)
- "Quality-Adjusted Life Years" are a popular objective function for public health charities

Optimization formalism

Optimization problems are typically stated as:

$$\boxed{\left(P_1\right) \equiv \begin{array}{ll} \text{minimize:} & f\left(x\right) \\ \text{subject to:} & x \in \Omega \subset \mathbb{R}^n \end{array}}$$

Objective function

Constraint set

$$\text{Where:} \quad f : \mathbb{R}^n \to \mathbb{R}, \ \Omega = \left\{x : g\left(x\right) \leq 0, \quad h\left(x\right) = 0\right\}$$

Why "minimize"?

Optimization formalism

Optimization problems are typically stated as:

$$\boxed{\left(P_1\right) \equiv \begin{array}{ll} \text{minimize:} & f\left(x\right) \\ \text{subject to:} & x \in \Omega \subset \mathbb{R}^n \end{array}}$$

Objective function

Constraint set

Where: $f : \mathbb{R}^n \to \mathbb{R},\ \Omega = \left\{x : g\left(x\right) \le 0,\quad h\left(x\right) = 0\right\}$

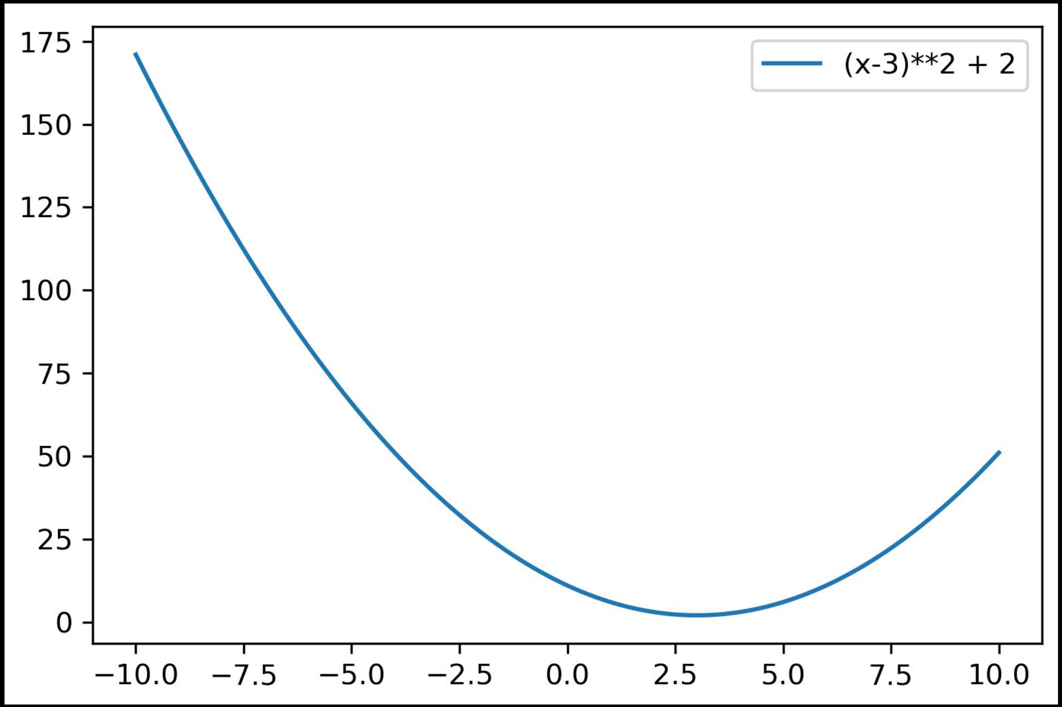
Or abbreviated as:

Problem name

$$\left(P_1\right) \equiv \begin{array}{ll} \min & f\left(x\right) \\ \text{s.t.} & x \in \Omega \end{array} \quad OR \quad \boxed{\min_{x\in\Omega} \quad f\left(x\right)} = v^*$$

Optimal value

Simple Minimization Example



If we constrain this problem to some bounds, what does the Extreme Value Theorem say?

$$\boxed{\min_{x\in\Omega} \quad f(x)} \quad \left(unconstrained\right)$$

$$f(x) = (x-3)^2 + 2, \quad \Omega = \mathbb{R}$$

$$\arg\min_{x\in\Omega} \quad f(x) = \boxed{x^* = ?}$$

$$\boxed{v^* = ?}$$

Optimal solution

What is constrained optimization?

- If Ω is all of R^n, then we have an <u>unconstrained optimization problem</u>: g(x) = h(x) = Ø (often we don't even write the Ω for unconstrained)
- If g(x) and/or h(x) are non-zero, then we have a <u>constrained optimization problem</u>
- Most problems in engineering have constraints
- There are tricks to turn constrained problems into unconstrained problems to solve them more easily

Multi-dimensional optimization

- Typically more than one variable to optimize
- Eg: why is this reactor this exact shape?
- Make a geometric representation of the shape (many variables)
- Optimize all variables together

Gradient-based optimization

- If we have the gradient $\nabla f(x)$, we can optimize functions up to **trillions** of variables
- Without the gradient, even 100 is tough

Defining the gradient

"Nabla"
aka "del"

$$\nabla \equiv \begin{bmatrix} \dfrac{\partial}{\partial x_1} \\ \dfrac{\partial}{\partial x_2} \\ \vdots \\ \dfrac{\partial}{\partial x_n} \end{bmatrix} \rightarrow \quad \forall f : \mathbb{R}^n \rightarrow \mathbb{R}, \quad \nabla f(x) = \begin{bmatrix} \dfrac{\partial f}{\partial x_1} \\ \dfrac{\partial f}{\partial x_2} \\ \vdots \\ \dfrac{\partial f}{\partial x_n} \end{bmatrix}$$

- First derivative for a scalar function in *n* dims
- Same shape (*n*) as the input vector x
- (What's the derivative of a vector function?)

Solve Linear Systems via Gradients

- We can pose the solution to Ax = b as an optimization problem of finding a vector r = Ax – b which is zero (if A is invertible) or with norm(r) as small as possible
- We call r the residual vector:

$$\underset{r,x\in\mathbb{R}^n}{\text{minimize}} \quad \|r\|$$

$$\text{subject to:} \quad r = Ax - b$$

Rearrange & simplify ||r|| ...

$$x^* = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left\| Ax - b \right\|_2 = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \sqrt{\left(Ax - b\right)^T \left(Ax - b\right)}$$

$$\overset{\substack{norms\\ are>0}}{=} \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left(Ax - b\right)^T \left(Ax - b\right)$$

What's "argmin"?

$$= \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left(Ax - b\right)^T Ax - \left(Ax - b\right)^T b$$

$$= \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \left(Ax\right)^T Ax - b^T Ax - \left(\left(Ax\right)^T - b^T\right) b$$

$$= \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} x^T A^T Ax - b^T Ax - x^T A^T b + b^T b$$

Making an Equivalent Problem

Since we don't care what the actual value of the residual is, just where it is minimized, we can ignore the constant b^Tb:

$$x^* = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \quad x^T A^T Ax - b^T Ax - x^T A^T b - \cancel{b^T b}$$

$$= \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \quad x^T A^T Ax - b^T Ax - x^T A^T b$$

$$= \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \quad x^T A^T Ax - 2b^T Ax$$

Let $P = A^TA$

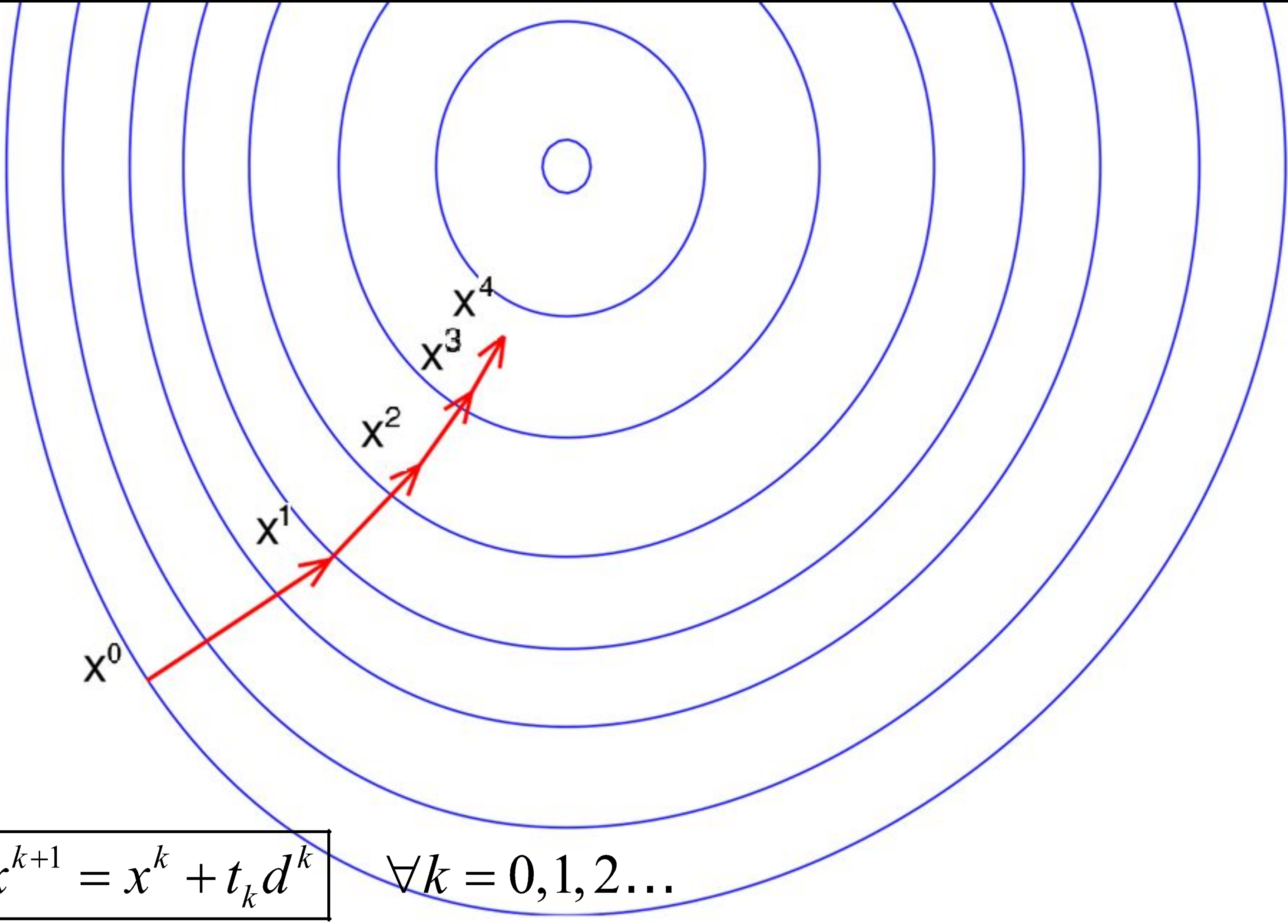
Let $c = 2A^Tb$

$$= \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \quad \boxed{x^T Px - c^T x} = \underset{x\in\mathbb{R}^n}{\operatorname{argmin}} \quad \boxed{f(x)}$$

Iterative optimizers

$$x^{k+1} = x^k + t_k d^k$$

- x^{k+1} is the next guess
- x^k is the current guess
- d^k is the <u>search direction</u> (e.g. the gradient)
- t_k is the <u>step size</u>



$$\boxed{x^{k+1} = x^k + t_k d^k} \qquad \forall k = 0,1,2\ldots$$

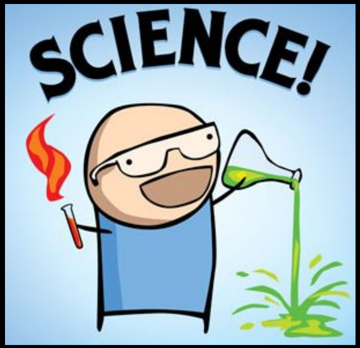
Method of Steepest Descent

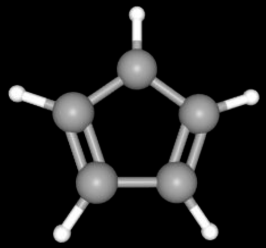
- The simplest gradient-based optimizer is steepest descent
- Search direction = negative of gradient

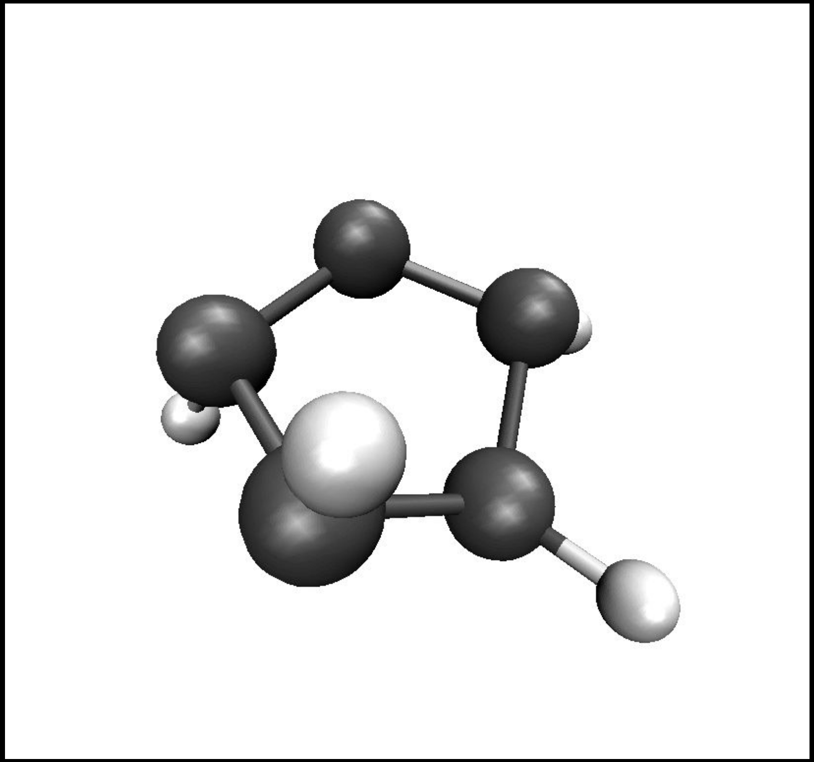
$$\boxed{d^k = -\nabla f\left(x^k\right)}$$

- This direction is guaranteed to be a descent direction (f goes down, by definition)
- Obviously effective: we go in the direction in which f goes down the fastest

Molecule shape discovery



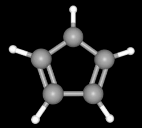




- Preferred shape is near minimum energy ΔU (aka E)
  - Why just E not ΔG?
  - $\Delta G = \Delta H - T\Delta S$
  - $\Delta H = \Delta U + P\Delta V$
  - $\Delta G \approx \Delta H \approx \Delta U = E$
- Can calculate $E(\mathbf{x})$ and $\nabla E(\mathbf{x})$ with quantum mechanics
- Given $E(\mathbf{x})$ and $\nabla E(\mathbf{x})$, how can we find $\mathbf{x}_{min}$?

Follow the forces ⇨ find x_{min}

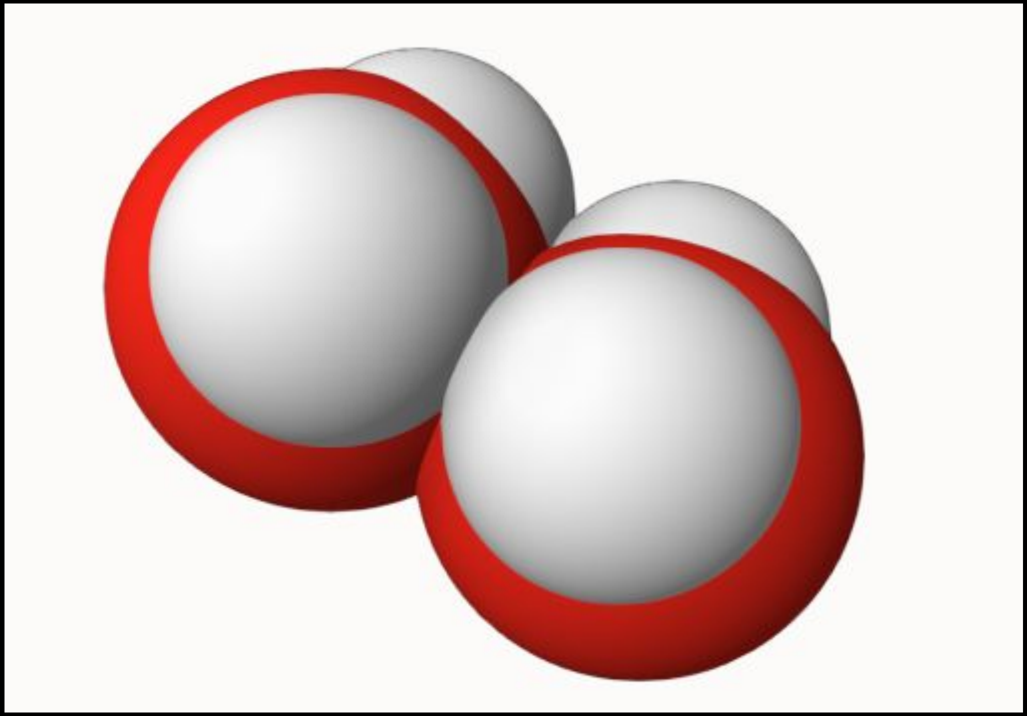
- Coords vector **x** represents all atoms
- Potential energy E(**x**) is a scalar
- ∇E(**x**) has same dimensions as **x**
- Each component of -∇E(**x**) is a force
- Follow the forces: <u>gradient descent</u>

x = 

	x	y	z
C	2.80	0.45	0.01
C	2.00	-0.12	0.00
C	3.61	-0.12	0.03
C	2.30	-1.07	0.01
C	3.30	-1.07	0.00
H	2.80	1.07	0.02
H	1.41	0.06	0.00
H	4.20	0.06	0.01
H	1.94	-1.58	0.00
H	3.67	-1.58	0.01

Where do we get F?

https://www.kaggle.com/allaboutchemistry/xtb-water-ivp



To view, download the xyz file and open it using: https://molstar.org/viewer/