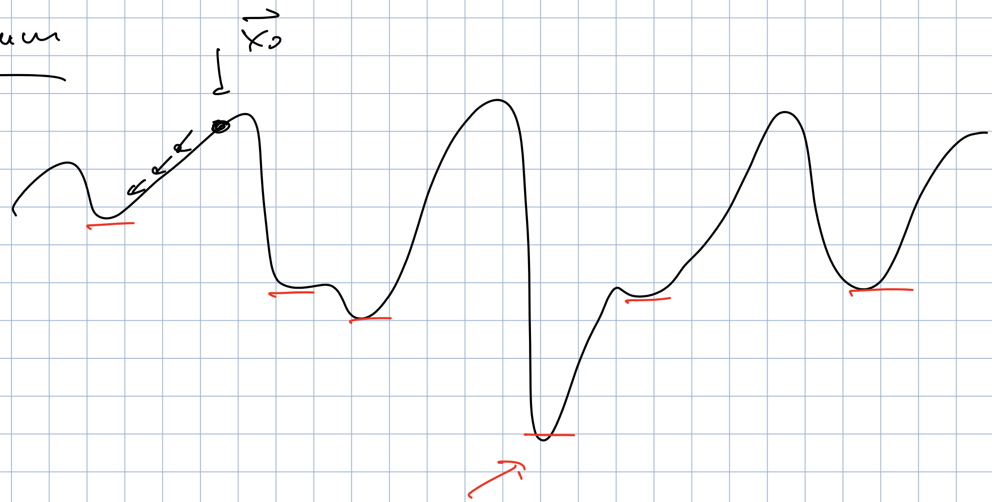
Numerical optimization

$$f : \vec{x} \in \mathbb{R}^N \to \mathbb{R}$$

$$\vec{x}^* = \arg\min \left(f(\vec{x}) \right)$$

absolute minimum

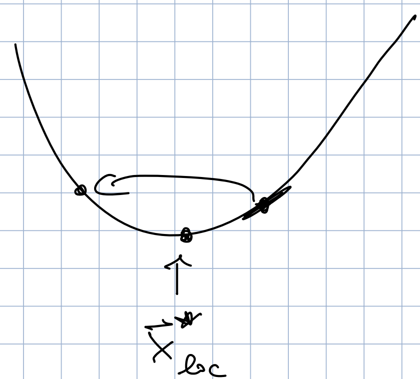
local minimum



Algorithm : standard gradient descent

$$\vec{x}_{k+1} = \vec{x}_k - \epsilon \vec{\nabla} f(\vec{x}_k)$$

ϵ sufficiently small



$$\left| f(\vec{x}_k) - f(\vec{x}^*_{loc}) \right| \sim O\left(\frac{1}{k}\right)$$

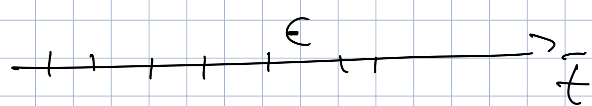
worst-case scenario

(1) Improvements : using physical insights

$$\dot{\vec{x}}$$

$$\vec{x} = \vec{x}(\tilde{t})$$

$$\frac{\vec{x}_{k+1} - \vec{x}_k}{\epsilon} = -\vec{\nabla} f$$



$$\dot{\vec{x}} = -\vec{\nabla} f = \vec{F}$$

$$m\ddot{\vec{x}} + \underbrace{\gamma \dot{\vec{x}}}_{\text{viscous drag}} - \vec{F} = 0 \quad \leftarrow \quad \text{Newton's}$$

t : $0, \Delta t, 2\Delta t, \ldots$

$$\vec{x}_k = \vec{x}(k\Delta t)$$

$$m \frac{\vec{x}_{k+1} + \vec{x}_{k-1} - 2\vec{x}_k}{(\Delta t)^2} + \gamma \frac{\vec{x}_{k+1} - \vec{x}_k}{\Delta t} = -\vec{\nabla} f$$

β/ϵ

$$\underbrace{\left(\frac{m}{(\Delta t)^2} + \frac{\gamma}{\Delta t}\right)}_{\epsilon^{-1}} (\vec{x}_{k+1} - \vec{x}_k) = \frac{m}{(\Delta t)^2}(\vec{x}_k - \vec{x}_{k-1}) - \vec{\nabla} f$$

$$\vec{x}_{k+1} = \vec{x}_k + \beta(\vec{x}_k - \vec{x}_{k-1}) - \epsilon \vec{\nabla} f$$

"momentum" term ($\rightarrow$ inertia)

Polyak's heavy-ball method

still one has $|f(\vec{x}_k) - f(\vec{x}^*)| \sim O\left(\frac{1}{k}\right)$

(w.c. scenario)

② Nesterov's accelerated gradient descent

$\vec{x}_1$

$\vec{y}_1 = \vec{x}_1$

$\vec{x}_2 = \vec{y}_1 - \epsilon \vec{\nabla} f(\vec{y}_1)$

$\vec{y}_2 = \vec{x}_2 + \left(\frac{k-1}{k+2}\right)\Big|_{k=2} (\vec{x}_2 - \vec{x}_1) \leftarrow$ Nesterov's point

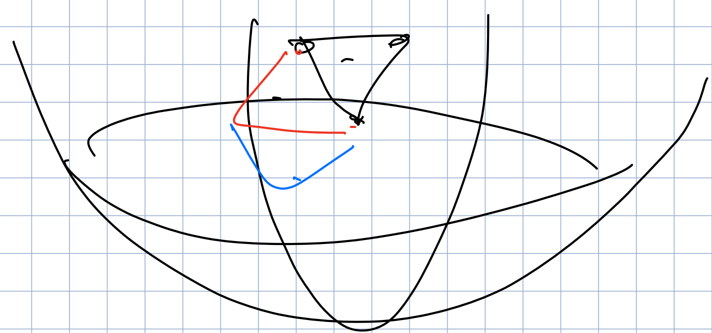
$$\vec{x}_3 = \vec{y}_2 - \epsilon \vec{\nabla} f(\vec{y}_2)$$

$\dots$

$$\boxed{\begin{aligned} \vec{y}_n &= \vec{x}_n + \left(\frac{n-1}{n+2}\right)\left(\vec{x}_n - \vec{x}_{n-1}\right) \\ \vec{x}_{n+1} &= \vec{y}_n - \epsilon \vec{\nabla} f(\vec{y}_n) \end{aligned}}$$

one can prove : $\left| f(\vec{x}_n) - f(\vec{x}^*) \right| \sim O\left(\frac{1}{k^2}\right)$

without calculating $\vec{\nabla} f$: simplex method



$\vec{x}$ continuously varying variable

Discrete variables

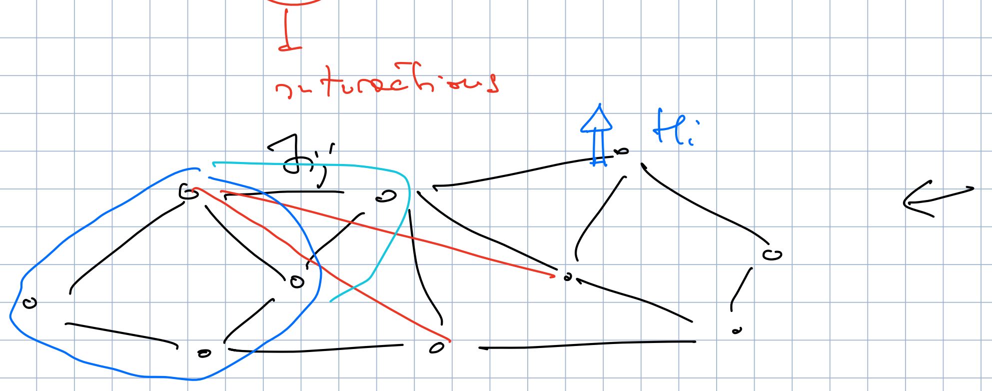
$$\vec{x} \to \vec{\sigma} = (\sigma_1, \sigma_2, \dots, \sigma_N)$$

$\sigma_i = \pm 1$ Ising spin

Energy of an Ising model

magnetic fields

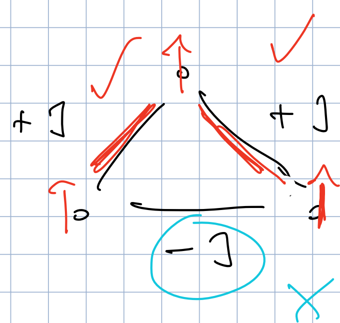
$$f(\vec{\sigma}) = -\sum_{ij} J_{ij} \sigma_i \sigma_j - \sum_i H_i \sigma_i$$

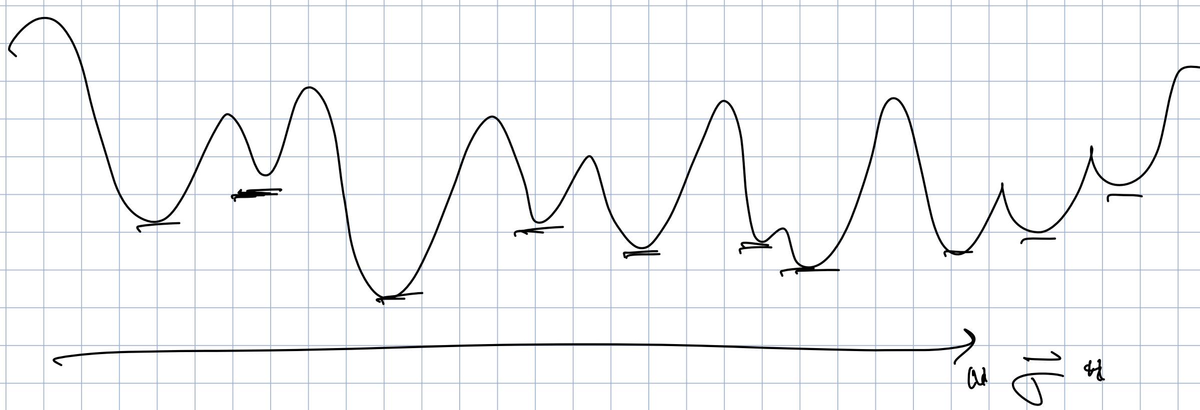


Competing interactions: do not admit a unique optimum minimizes all interaction terms

"frustration"

Ex.





frustration + disorder (randomness)

↳ "spin glasses"

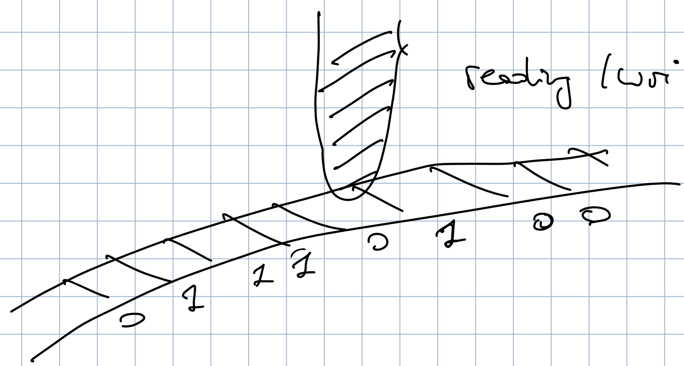
Examples of minimization of functions of discrete variables:

· Combinatorial optimization

Primer on computational complexity

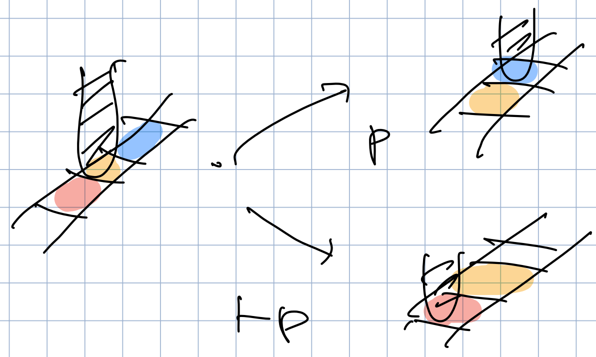
"Tools" to solve a problem :

Deterministic Turing machine (DTM)



Non-deterministic Turing machine (NTM)

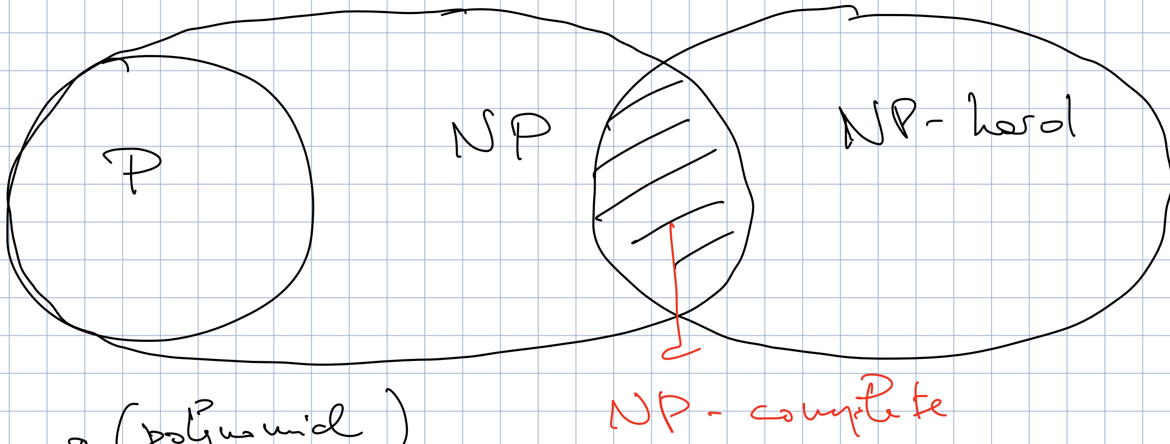
TM which can branch in its operations



A problem is solved by a NTM if one of its branches reaches a solution.

Complexity classes for computational problems

conjecture : P ⊂ NP ?



P → (polynomial)

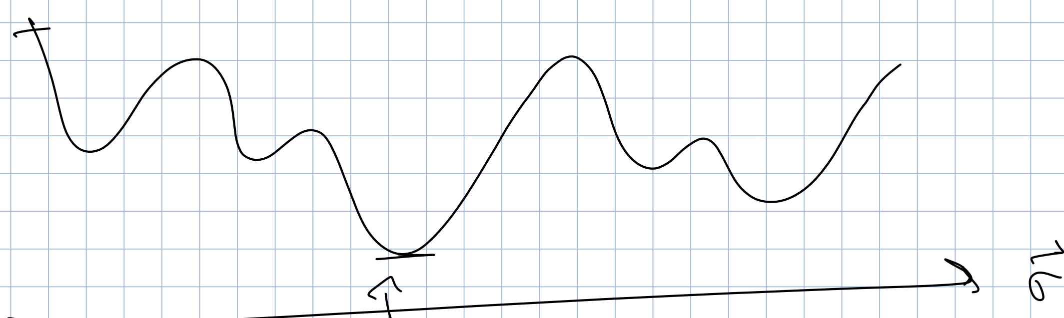
P: probs. which can be solved in poly time by a DTM

NP: (non-deterministic polynomial): probs that can be solved in poly time by a NTM. its solution can be checked in poly time using DTM

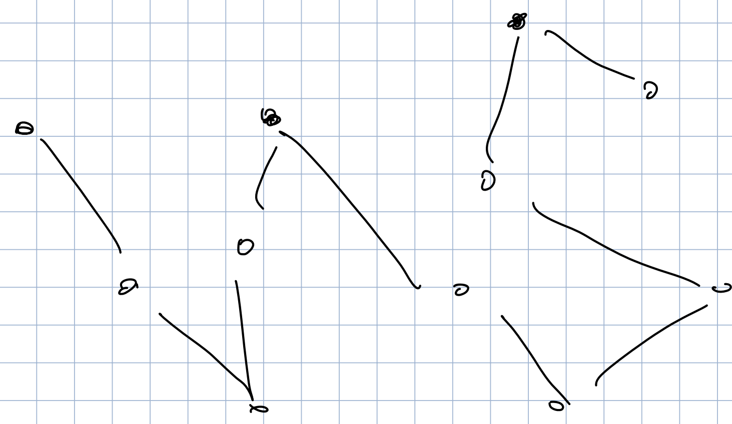
NP-hard: probs as hard as NP ones, such that all problems in NP can be mapped efficiently onto them in polynomial time, but NP-hard probs. may not admit a verification of their solution in poly time using a DTM.

Examples of $\boxed{\text{NP-hard}}$ problems

minimization



Travelling salesman problem



?

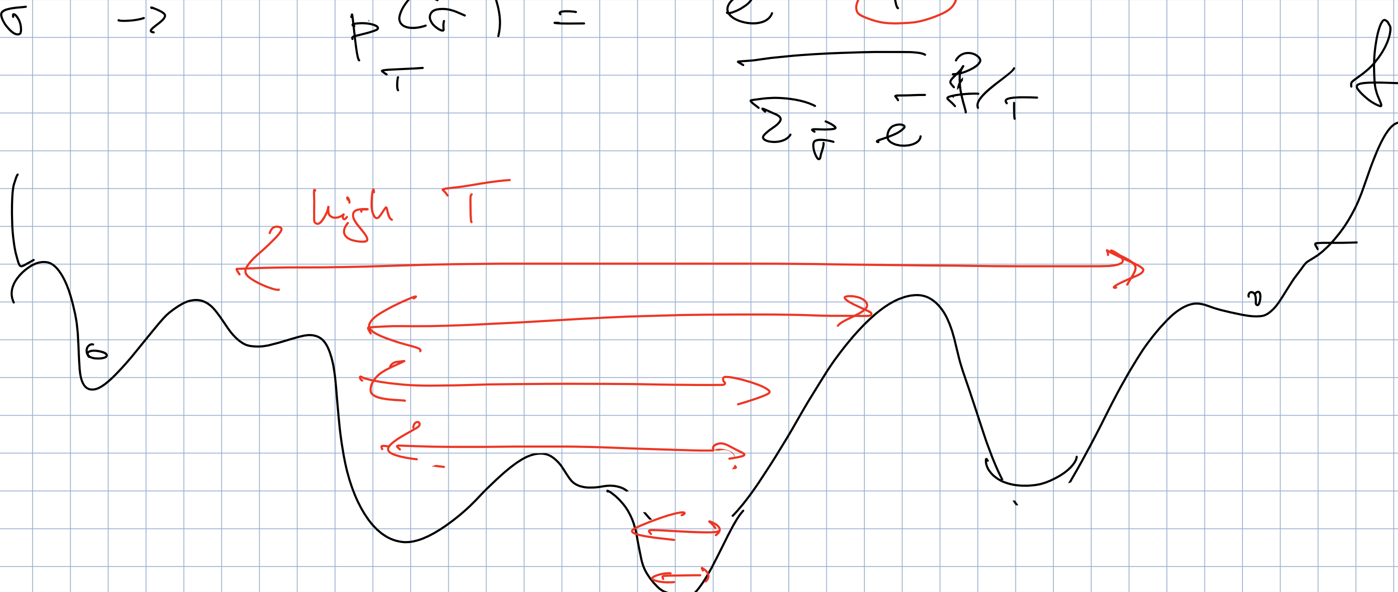
many other problems ?

Approach to minimization of functions of discrete variables (and not only)

=> "simulated annealing"

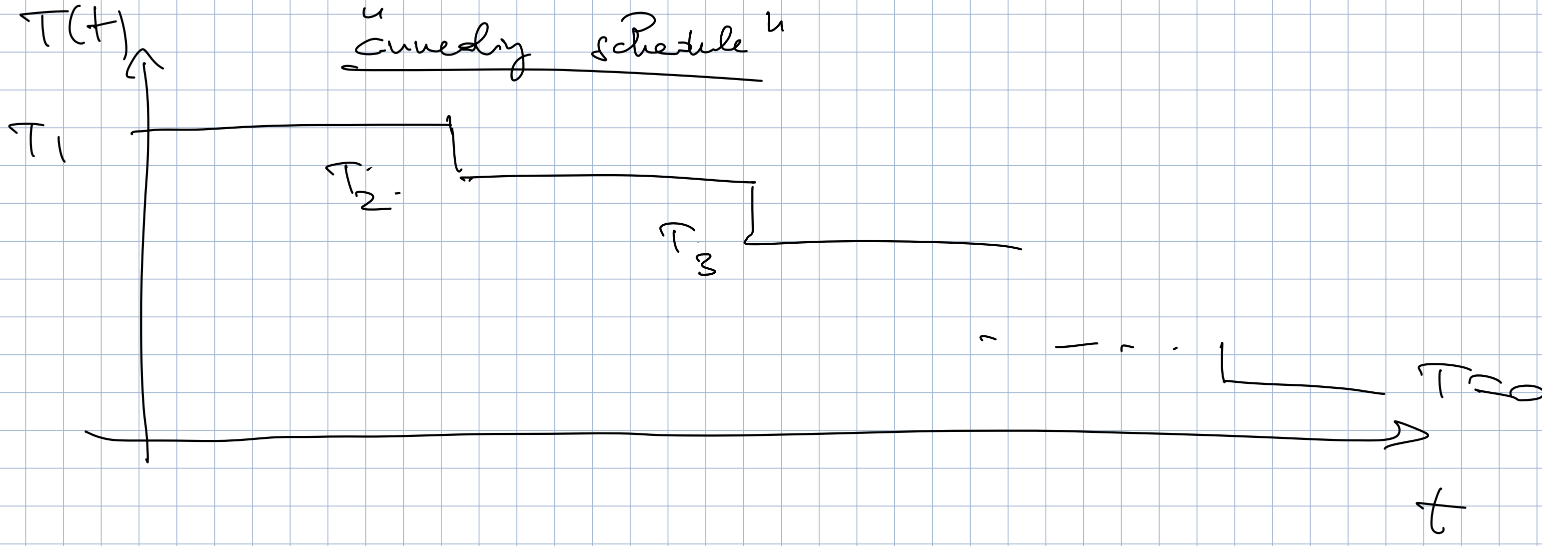
$f(\vec{\sigma}) \to$ energy

$$\vec{\sigma} \to p_T(\vec{\sigma}) = \frac{e^{-\frac{f(\vec{\sigma})}{T}}}{\sum_{\vec{\sigma}} e^{-f/T}}$$



In practice: use MC to explore configurations $\vec{\sigma}$

"annealing schedule"



Simulated annealing may fail?

Markov chain fails to reproduce the statistics of $p_T(\vec{\sigma})$

$$\tau \simeq O(\exp(N))$$ at least below some characteristic T

Proof: simulated annealing converges to the absolute minimum if

$$T(t) \simeq \frac{cN}{\log t}$$

$$t \simeq \exp\left(\frac{cN}{T}\right)$$

(5) Numerical solution to ordinary differential equations (ODEs)

ODE $\quad \vec{x}(t)$

$$\vec{F}(\vec{x}(t), \dot{\vec{x}}, \ddot{\vec{x}}, \dots, \vec{x}^{(n)}; t) = 0$$

(only time derivatives)

n-th order differential equation

$\Rightarrow$ set of 1-st order differential equations

$$\begin{cases} \vec{y}_1(t) = \vec{x}(t) \\ \vec{y}_2(t) = \dot{\vec{x}}(t) = \dot{\vec{y}}_1(t) \\ \vec{y}_3(t) = \ddot{\vec{x}}(t) = \dot{\vec{y}}_2(t) \\ \dots \\ \vec{y}_{n+1}(t) = \vec{x}^{(n)}(t) = \dot{\vec{y}}_n \end{cases}$$

$$\vec{G}(\underbrace{\vec{y}_1, \vec{y}_2, \dots, \vec{y}_{n+1}}_{\vec{y}}; \underbrace{\dot{\vec{y}}_1, \dot{\vec{y}}_2, \dots, \dot{\vec{y}}_n}_{\dot{\vec{y}}}, \dot{\vec{y}}_{n+1}; t) = 0$$

$$\vec{G}(\vec{y}, \dot{\vec{y}}; t) = 0 \leftarrow$$

In physics: Cauchy problems

$$\boxed{\begin{aligned} \dot{\vec{y}} &= \vec{f}(\vec{y}; t) \\ \vec{y}(0) &= \vec{y}_0 \end{aligned}}$$

Ex. Newton's equation

$$\ddot{\vec{x}} = -\frac{\vec{F}}{m}$$

$$\vec{y}_1 = \vec{x} \qquad \vec{y}_2 = \dot{\vec{x}}$$

$$\begin{cases} \dot{\vec{y}}_1 = \vec{y}_2 \\ \dot{\vec{y}}_2 = -\dfrac{\vec{F}(t)}{m} \end{cases}$$

$$\dot{\vec{y}} = \vec{f}(\vec{y}; t)$$

Good news : numerical integration of a Cauchy problem is not hard

aim : numerical approx. to the solution

$$\| \vec{y}_{num}(t) - \vec{y}_{ex}(t) \| < \epsilon$$

num. exact

$t \in [0, t_0]$

$$\vec{y} = (y_1, y_2, \ldots, y_N)$$

$\hookrightarrow N = \#$ of dimensions

For some cases $A \gg 1$

depending on the integration scheme

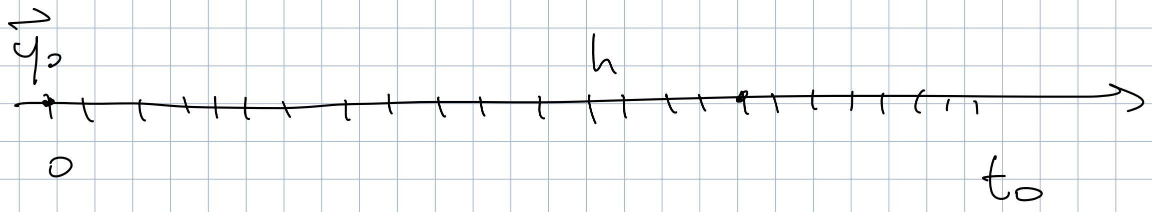
$$T \simeq A \frac{N}{\epsilon^{\alpha}} \frac{t_0}{t}$$

T time to solution

$\alpha \sim \mathcal{O}(1)$

depends on the integration scheme

Numerical integration scheme : <u>discretization of time</u>



h : time step $\qquad t_n = nh \qquad \vec{y}_n = \vec{y}(nh) = \vec{y}(t_n)$

<u>explicit integration scheme</u> = rule of update

$$\vec{y}_{n+1} = \vec{\mathcal{Y}}_n(\vec{y}_n, \vec{y}_{n-1}, \dots, \vec{y}_0 ; h, \vec{f})$$

implicit integration scheme

$$\vec{y}_{n+1} = \vec{\mathcal{Y}}_n(\vec{y}_{n+1}, \vec{y}_n, \vec{y}_{n-1}, \dots, \vec{y}_0 ; h, \vec{f})$$

<u>Exception</u> to time discretization : linear problem

$$\dot{\vec{y}}(t) = A\vec{y}$$

diagonalise $A \rightarrow U^{+} A U = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_N)$

$$\vec{\eta}(t) = U \vec{\xi}(t)$$

$$\vec{\xi}(t) = \left(\xi_1(0)\, e^{\lambda_1 t}, \ \xi_2(0)\, e^{\lambda_2 t}, \ \ldots, \ \ldots \right)$$