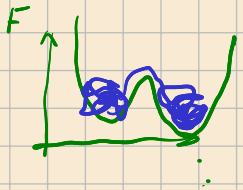
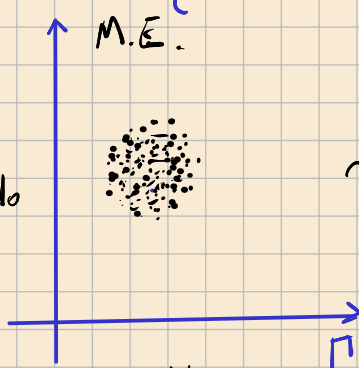


* Conclusión: ¿por qué funciona MD? Veamos el espacio de fases



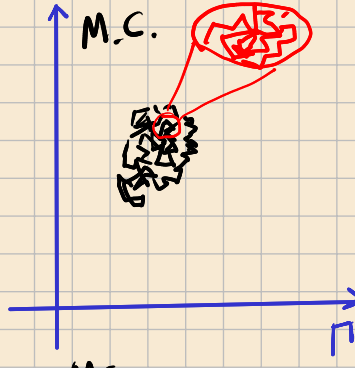
Un punto.
↓
un microestado
del sistema



Ensemble

- micro Estados que recorre el sistema en una condición física

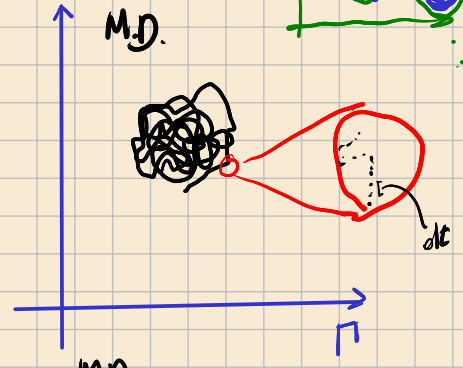
~



MC

- cadena de Markov
- muestreamos con configuraciones al azar
- Assume ergodicidad

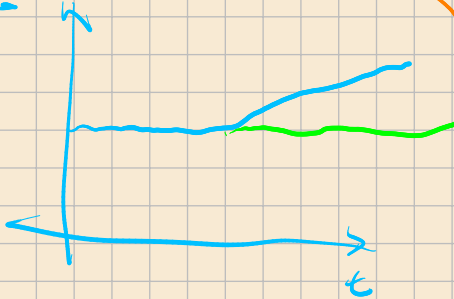
~



MD

- trayectoria clásica en el espacio de fases
- Las configuraciones salen de integrar ec's de Newton
- Assume ergodicidad

Ej: Sistema aislado



Newton, $E = \text{cte.}$

Ensemble Microcanónico.

Programa MD básico

loop temporal

program md

call init

t=0

do while (t.lt.tmax)

→ call force(f,en)

→ call integrate(f,en)

t=t+delt

call sample

enddo

stop

end

simple MD program

initialization

MD loop

determine the forces

integrate equations of motion

sample averages

$$Z_{max} = N_{steps} \times dt$$

\vec{r}_0, \vec{v}_0

DO $i=1, N_{steps}$

Medir variables físicas. Funciones de \vec{r}_i y \vec{v}_i

Bib: Frenkel & Smit

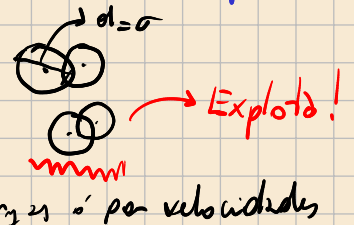
Initialization: Asignamos posiciones y velocidades iniciales

- Las partículas deben tener posiciones compatibles con la estructura que se quiere simular.

- Puede ser el agua → líquido, gas

- En una estructura cúbica simple (sólido)

- Velocidades al azar: puede obtenerse de una distribución de Boltzmann compatible con una T → v_{norm} es Ziggurat



por fuerzas o por velocidades

Reescalado de velocidades

Pueden determinarse las velocidades usando $u_{ini} \in [-0.5, 0.5]$

Tenemos para la "T instantánea" $k_B T(t) \equiv \frac{\sum_{i=1}^N m v_{\alpha,i}^2(t)}{N_f}$

Temperatura deseada

⇒ Podemos reescalar todas las velocidades con el factor $\sqrt{\frac{T}{T(t)}}$

⇒ Así la primera configuración tiene velocidades tales que la temperatura es la T deseada.

* Puede repetirse para varios pasos de dinámica

Cálculo de la fuerza

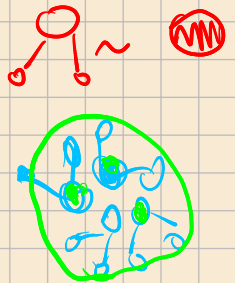
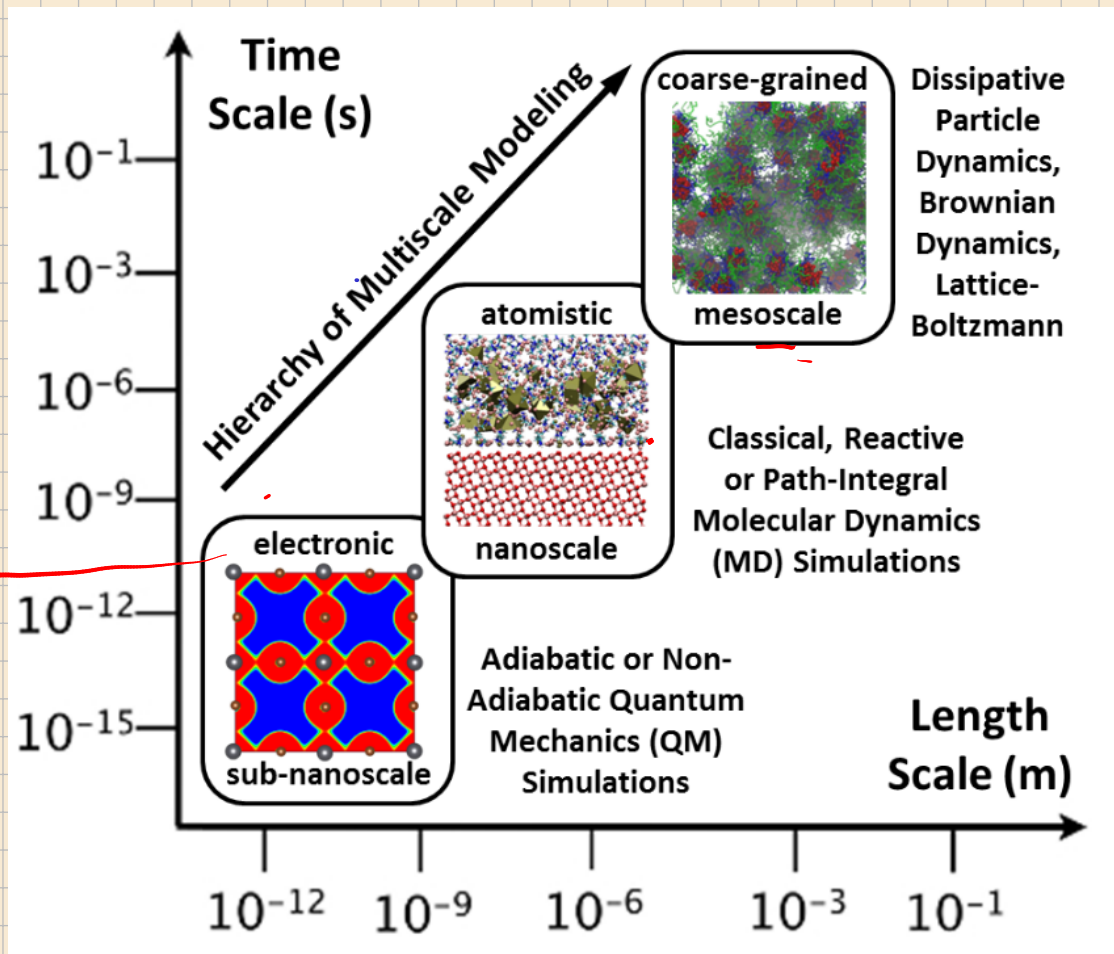
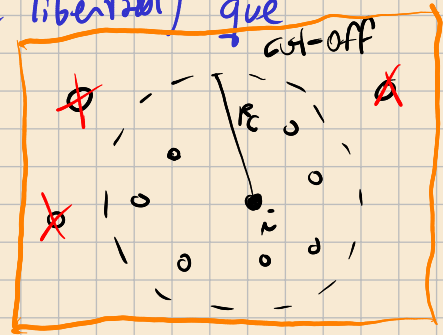


- Parte más pesada del programa (puede ser $\sim 80\%$ del tiempo o más)
- Si consideramos interacciones aditivas de 2 pares (Ej: Lennard-Jones)
- \Rightarrow Para N partículas debemos evaluar $\frac{N(N-1)}{2} \sim N^2$ distancias entre pares.

\Rightarrow Sin "trucos" la evolución de la fuerza escala $\boxed{\sim N^2}$

* Esto limita el número de partículas (o grados de libertad) que pueden incluirse en una simulación

Escala y grados de libertad en una simulación



Integración de las Ecuaciones de Movimiento: Algoritmo de Verlet

* Necesitamos un algoritmo para integrar las ecuaciones de Newton

* Debe ser rápido y suficientemente preciso. El algoritmo de Verlet, uno de los primeros sugeridos, cumple muy satisfactoriamente ambos requerimientos.

Desarrollo de Taylor en $+\Delta t$ y $-\Delta t$:

$$\begin{cases} r(t+\Delta t) = r(t) + \underbrace{v(t)}_{\frac{\partial x}{\partial t}} \Delta t + \underbrace{\frac{f(t)}{2m}}_{\frac{\partial^2 x}{\partial t^2}} \Delta t^2 + \frac{\Delta t^3}{3!} \left. \frac{\partial^3 x}{\partial t^3} \right|_t + \mathcal{O}(t^4) & (1) \\ r(t-\Delta t) = r(t) - v(t) \Delta t + \frac{f(t)}{2m} \Delta t^2 - \frac{\Delta t^3}{3!} \frac{\partial^3 x}{\partial t^3} + \mathcal{O}(t^4) & (2) \end{cases}$$

Sumando (1) y (2):

$$r(t+\Delta t) + r(t-\Delta t) = 2r(t) + \frac{f(t)}{m} \Delta t^2$$

Reescribiendo:

$$r(t+\Delta t) \simeq 2r(t) - r(t-\Delta t) + \frac{f(t) \Delta t^2}{m} \quad (3)$$

Algoritmo de posiciones de Verlet

* El error es del orden Δt^4

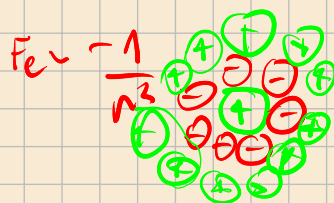
* No usa velocidades en principio \Rightarrow se reformula como velocity Verlet para tener expresiones explícitas de las velocidades

Restando (1) y (2):

$$r(t+\Delta t) - r(t-\Delta t) = 2v(t) \Delta t + \mathcal{O}(\Delta t^3)$$

$$v(t) = \frac{r(t+\Delta t) - r(t-\Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2) \quad (4)$$

Algoritmo de Verlet . velocidades



Algoritmo Velocity-Verlet

* Hay varias implementaciones equivalentes del algoritmo de Verlet
 * Velocity-Verlet es uno de los más utilizados (y es el que usaremos)

Forma final del algoritmo Velocity-Verlet:

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m} \Delta t^2$$

$$v(t+\frac{1}{2}\Delta t) = v(t) + \frac{1}{2} \Delta t \frac{f(t)}{m}$$
 Computar $f(t+\Delta t)$ y $a(t+\Delta t) = \frac{f(t+\Delta t)}{m}$

$$v(t+\Delta t) = v(t+\frac{1}{2}\Delta t) + \frac{1}{2} \Delta t a(t+\Delta t) = r(t+\Delta t), v(t+\Delta t), f(t+\Delta t)$$
 - computo cantidades físicas

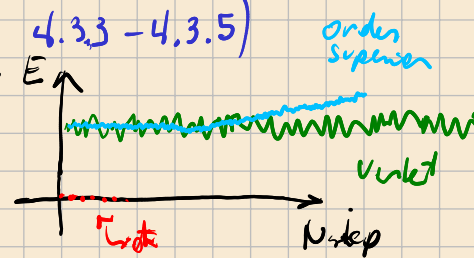
$r(t), v(t), f(t)$ (*)
 Sirve para implementación y la práctica

all force

* Existen algoritmos de orden superior (implican más tiempo de máquina)

* Verlet es un algoritmo reversible temporalmente → tiene estabilidad numérica a tiempos largos (ver: Frenkel y Smith 4.3.3-4.3.5)

Algoritmo symplectico



Potenciales de interacción típicos

$$U(R) = \sum_{\text{bonds}} k_r (r - r_{eq})^2$$

bond



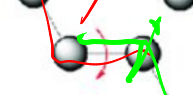
$$+ \sum_{\text{angles}} k_\theta (\theta - \theta_{eq})^2$$

angle



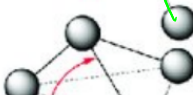
$$+ \sum_{\text{dihedrals}} k_\phi (1 + \cos[n\phi - \gamma])$$

dihedral



$$+ \sum_{\text{impropers}} k_\omega (\omega - \omega_{eq})^2$$

improper



$$+ \sum_{i < j}^{\text{atoms}} \epsilon_{ij} \left[\left(\frac{r_m}{r_{ij}} \right)^{12} - 2 \left(\frac{r_m}{r_{ij}} \right)^6 \right]$$

van der Waals

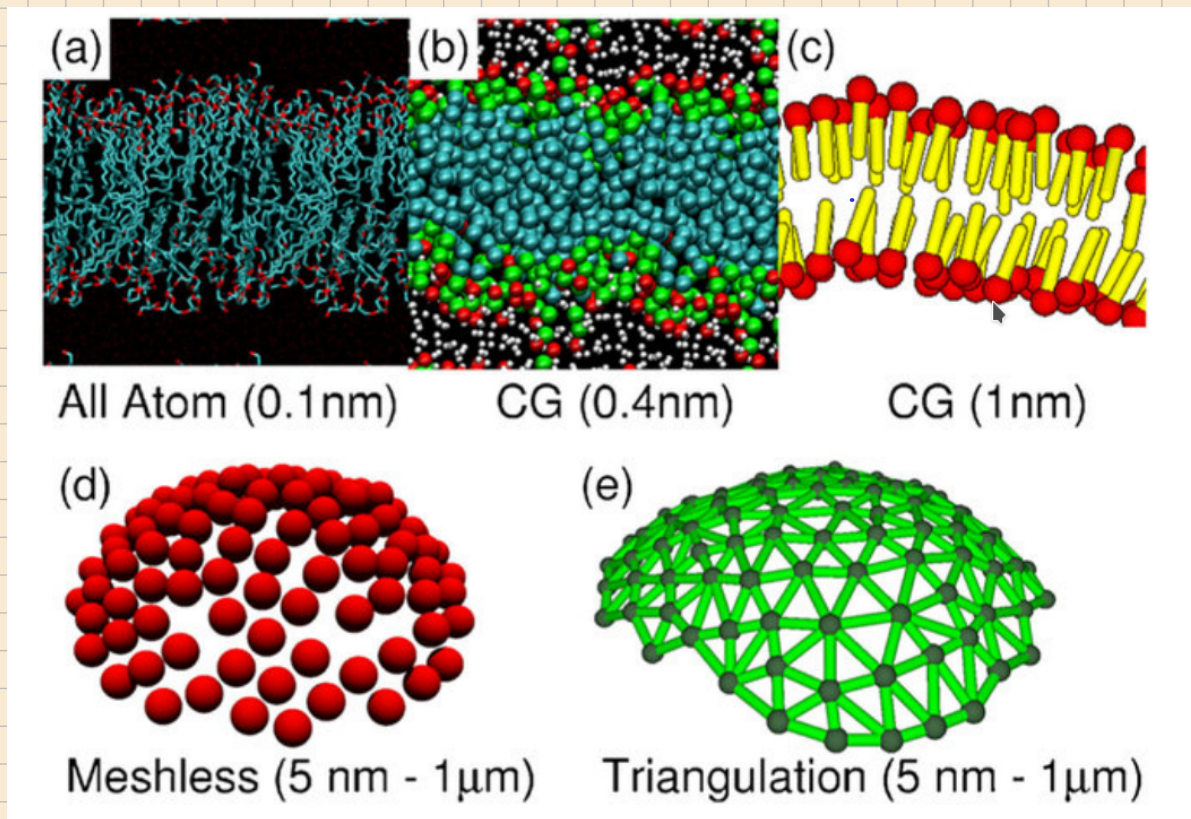


$$+ \sum_{i < j}^{\text{atoms}} \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}}$$

electrostatic



Potenciales de grado grueso



NUEVO!

Potenciales mejorados por Machine Learning (MLiPS)

nature
COMMUNICATIONS

ARTICLE

DOI: 10.1038/s41467-018-06169-2

OPEN

Towards exact molecular dynamics simulations with machine-learned force fields

Stefan Chmiela¹, Huziel E. Sauceda², Klaus-Robert Müller^{1,3,4} & Alexandre Tkatchenko⁵

Molecular dynamics (MD) simulations employing classical force fields constitute the cornerstone of contemporary atomistic modeling in chemistry, biology, and materials science. However, the predictive power of these simulations is only as good as the underlying interatomic potential. Classical potentials often fail to faithfully capture key quantum effects in molecules and materials. Here we enable the direct construction of flexible molecular force fields from high-level ab initio calculations by incorporating spatial and temporal physical symmetries into a gradient-domain machine learning (sGDML) model in an automatic data-driven way. The developed sGDML approach faithfully reproduces global force fields at quantum-chemical CCSD(T) level of accuracy and allows converged molecular dynamics simulations with fully quantized electrons and nuclei. We present MD simulations, for flexible molecules with up to a few dozen atoms and provide insights into the dynamical behavior of these molecules. Our approach provides the key missing ingredient for achieving spectroscopic accuracy in molecular simulations.

→ Buen modelo para vibraciones moleculares de alta energía.