Gases

ExactasPrograma

Facultad de Ciencias Exactas y Naturales, UBA

Verano 2020

GASES

En el siglo XVII
Boyle-Mariot

$$PV = k_1$$

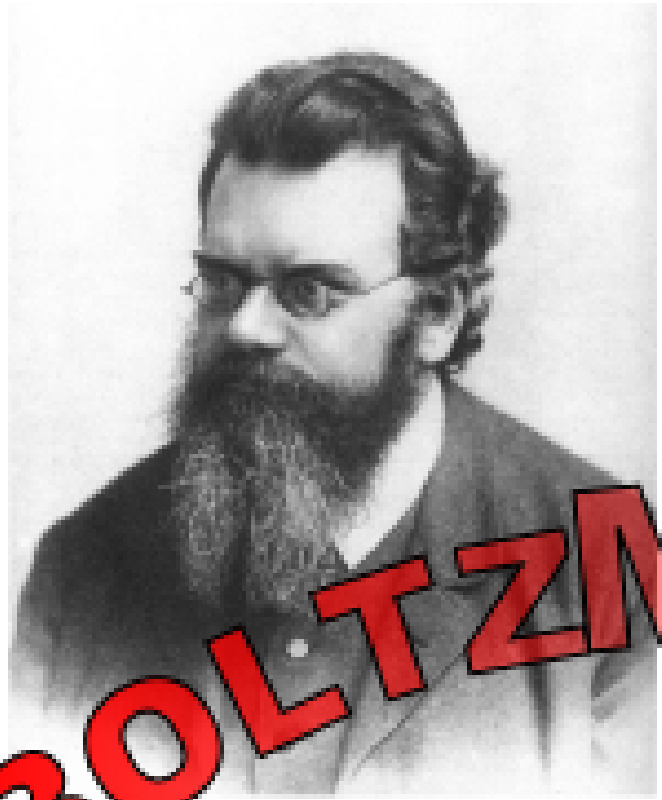
Gay Lussac

$$P/T = k_2$$

Ecuación de estado de un gas ideal

$$PV = nRT$$

y en eso llegó ...



(En realidad fueron unos cuantos, empezando por Daniel Bernoulli, J. C. Maxwell entre otros)

Idea absurda

- Los gases están compuestos por partículas que casi no interactúan entre ellas (salvo por choques esporádicos).
- La presión es producida por los choques de las partículas contra las paredes.
- La temperatura está relacionada con el promedio de energía cinética de esas partículas.

$$\frac{\sum_j^{npart} \frac{1}{2} m_j |V_j|^2}{npart} = \langle Ec \rangle = \frac{3}{2} K_B T \tag{1}$$

donde K_B es la constante de Boltzmann ($K_B = \frac{R}{N_A}$).

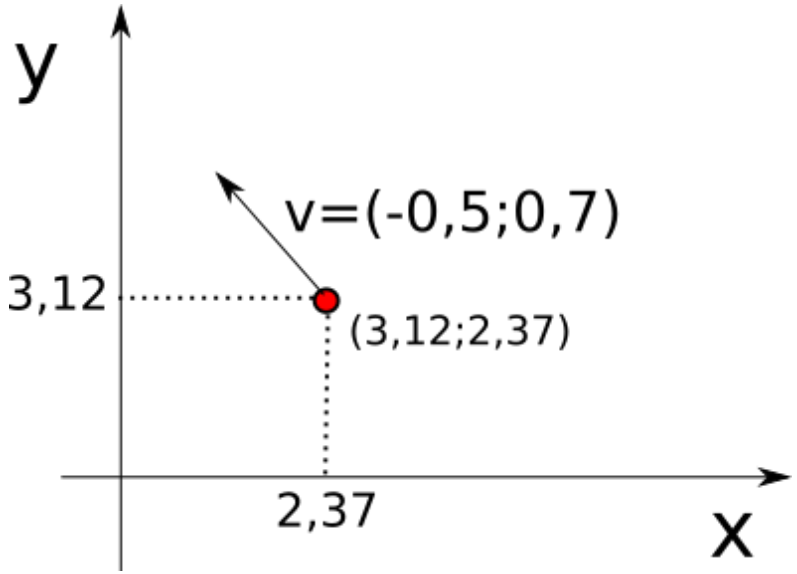
¿Cómo saber si el modelo está bien?

Tengo dos opciones:

1. Hago un modelo muy simplificado y resuelvo con lápiz y papel.
2. Hago un modelo menos simplificado y **simulo** usando una computadora.

¡A modelar!

- Cómo dijo un gran filósofo contemporáneo “paso a paso”. Empecemos con una partícula en una caja.
- Primera simplificación: **sólo dos dimensiones**.
- Ahora la posición tiene dos componentes: x e y (ó puedo llamar $pos(x, y)$).
- A la velocidad le pasa lo mismo: V_x y V_y (o $V(v_x, v_y)$).



Puedo tratar x e y independientemente

Ya estuvimos jugando con MRUV, volvemos a usarlo:

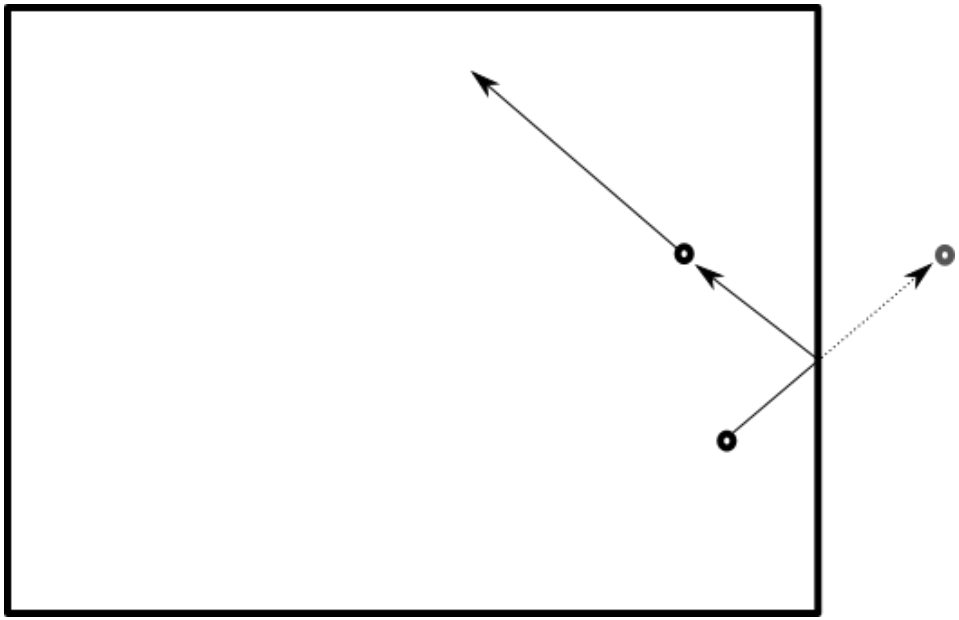
$$x(t + \Delta t) = x(t) + v_x(t)\Delta t + \frac{1}{2}a_x \Delta t^2$$

$$y(t + \Delta t) = y(t) + v_y(t)\Delta t + \frac{1}{2}a_y \Delta t^2$$

donde $a_x = f_x/m$ y $a_y = f_y/m$

Ahora pongámosla en una caja

Hay que tener en cuenta los choques con las paredes:

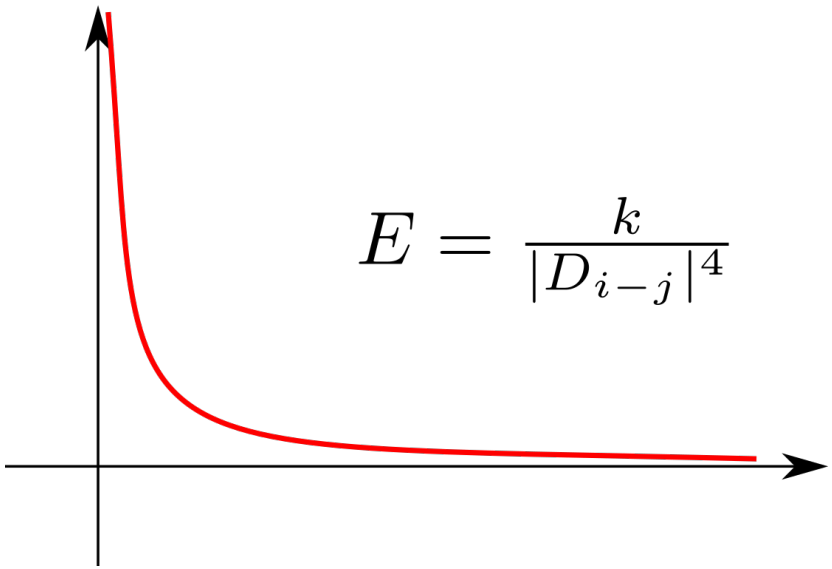


Una manera:

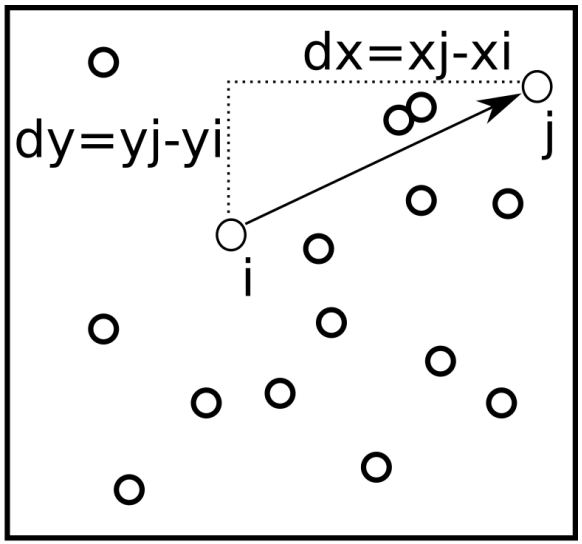
- Si $x > xmax \Rightarrow vx = -vx$
- y $x = x - 2(x - xmax)$
- y a las cosas de "y" no les pasa nada.

Bolas que chocan

En el modelo de gases las partículas (átomos o moléculas) pueden chocar.
Una forma de hacerlo es poniendo un potencial repulsivo entre pares:



$$E = \frac{k}{|D_{i-j}|^4}$$



Que la fuerza esté contigo

- D es la distancia entre las partículas i y j:

$$|D_{i-j}| = \sqrt{(x(i) - x(j))^2 + (y(i) - y(j))^2} \tag{2}$$

- La fuerza que siente la partícula i por su interacción con la partícula j:

$$F_x^j(i) = -\frac{\partial E}{\partial X(i)} = 4k\frac{(x(i) - x(j))}{|D_{i-j}|^6} \tag{3}$$

- La fuerza total sería:

$$F_x^{total}(i) = 4k\sum^{j\neq i}\frac{(x(i) - x(j))}{|D_{i-j}|^6} \tag{4}$$

$$F_y^{total}(i) = 4k\sum^{j\neq i}\frac{(y(i) - y(j))}{|D_{i-j}|^6} \tag{5}$$

1. Definir las posiciones y velocidades iniciales.
2. Calcular las fuerzas para cada partícula:

```
for i in range(npart):
  Fx[i]=0
  for j in range(npart):
    if i !=j:
      Fx[i] += 4*k*(x[i]-x[j])/|Dij|**6
```

Truco

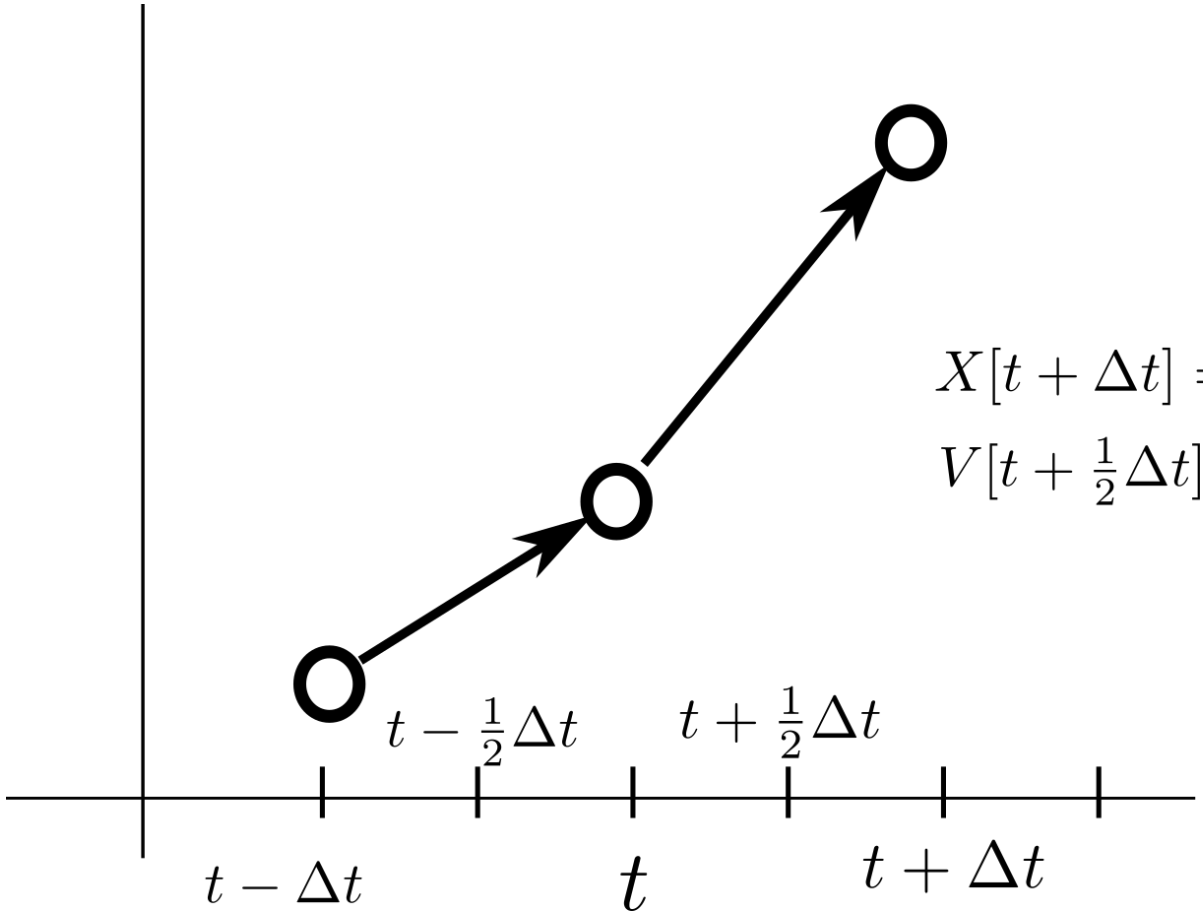
Puedo usar $D2_{i-j} = (x_i - x_j)^2 + (y_i - y_j)^2$.
Entonces: $|D_{i-j}|^6 = {D2_{i-j}}^3$.

3. calculo las velocidades nuevas según:
   $V_x[i] = V_x[i] + \frac{F_x[i]}{m} * dt$
4. Luego calculo las nuevas posiciones (MRU):
   $x[i] = x[i] + V_x[i] * dt$
5. veo si chocan con las paredes y vuelvo al punto 2.

Creer o reventar

Esto es equivalente a los que hicieron en MRUV

Sólo si están preguntones



$$X[t + \Delta t] = X[t] + V[t + \tfrac{1}{2}\Delta t] * \Delta t$$
$$V[t + \tfrac{1}{2}\Delta t] = V[t - \tfrac{1}{2}\Delta t] + a[t] * \Delta t$$

Para escribir en un archivo

Para poder ver las dinámicas vamos a usar VMD (Visualizing Molecular Dynamics).
Tenemos que escribir los datos en un archivo.
primero lo abrimos con: salida=open(”salida.xyz”,”w”)

```
salida=open("salida.xyz","w")
```

“salida” es el nombre con el cual lo vamos a llamar adentro del código, salida.xyz es el nombre del archivo que generará y con “w” le decimos a Python que lo vamos a escribir.
Luego hacemos el print casi como siempre:

```
print ("quiero escribir a un archivo", file=salida)
```

Escribiendo un archivo `.xyz`

Formato:

- En la primer línea va el número de partículas (digamos `npart`)
- Luego una línea vacía.
- luego `npart` líneas con: tipo de átomo coordx coordy coordz (separado por espacios)

```
print (npart,file=salida)
print (" ",file=salida)
for jj in range(npart):
    print ("6",x[jj],y[jj],"0", file=salida)
```