

UNIVERSITAT DE BARCELONA
QuantumLabUB

PRÀCTIQUES EN EMPRESA
Grau de Física

Particles in a box

Jofre Vallès Muns

Tutors:
Carles Calero Borrallo
Montserrat Guilleumas Morell
Bruno Juliá Díaz

Març 2020 - Febrer 2021

Índex

1	Introducció	2
1.1	QuantumLabUB	2
2	Introducció i fonament teòric	3
2.1	"Particles in a box"	3
2.1.1	Simulation	3
2.1.2	Demo	5
2.1.3	Game	6
2.2	Equacions	7
2.2.1	Potencial $V(\vec{r}_i, \vec{r}_j)$	8
2.3	Mètode numèric	11
2.3.1	Càlcul matricial	13
2.4	Entropia	14
3	Desenvolupament del projecte	16
3.1	Metodologia	16
3.2	Idea original	16
3.3	Rendiment	17
3.3.1	<i>Velocity-Verlet</i>	17
3.3.2	<i>Close particle list</i>	18
3.3.3	Numba	19
3.4	Accessibilitat i estètica	20
4	Valoració i conclusions	22
	Referències	23
	Apèndix: Guia d'usuari	24

1. Introducció

Aquest document recull el meu treball fet des del març del 2020 fins al febrer del 2021 en l'assignatura de *Pràctiques en Empresa*, en el projecte Quantum-LabUB [1] del Departament de Física Quàntica i Astrofísica i el Departament de Física de la Matèria Condensada de la Universitat de Barcelona. Ha estat supervisat per Carles Calero Borralló, Montserrat Guilleumas Morell i Bruno Juliá Díaz i hi he dedicat un total d'unes 290 hores.

En aquest document trobareu tant el desenvolupament i explicació en detall del treball com el fonament teòric amb què treballa. A més, com que la finalitat d'aquest programa és molt divulgativa, s'ha inclòs com a apèndix la guia d'usuari, pensada perquè els professors que els interessi mostrar aquest programa als seus alumnes (o qualsevol persona que en tingui interès) tinguin unes instruccions fàcils per saber com funciona i com es pot accedir als elements fonamentals del programa.

1.1 QuantumLabUB

Des de fa uns anys, QuantumLabUB és un projecte en què estudiants cursant el grau de física desenvolupen codi amb intenció divulgativa. Inicialment, QuantumLabUB s'encarava en la divulgació d'elements de mecànica quàntica, però com que ja s'incorporava elements de mecànica clàssica en els projectes d'alguns anteriors alumnes i veient el potencial per explorar aquest camp, aquest any s'ha incorporat Carles Calero Borralló (Departament de Física de la Matèria Condensada) i s'ha pogut tractar la divulgació en l'àmbit de la mecànica estadística, creant així la secció ClassicalLabUB, enfocada en la divulgació de física clàssica, com és el cas del meu treball.

En aquest semestre he pogut participar en aquestes pràctiques juntament amb dos alumnes més, cadascú amb el seu projecte independent, l'Adrià Bravo i la Júlia Cabrera.

2. Introducció i fonament teòric

En aquesta secció es presenta el programa amb les seves funcions i s'explica el fonament teòric que hi ha darrere.

2.1 ”Particles in a box”

Particles in a box és el nom del programa resultant del projecte d'aquestes pràctiques. En la figura 2.1 es pot veure la pàgina que s'obre a l'iniciar el programa, que consta de dues seccions, *Simulation* i *Demo*:



Figura 2.1: Menú, pàgina per defecte en obrir el programa.

2.1.1 Simulation

La secció *Simulation* és on resideix la gran part del programa. Es tracta del lloc on l'usuari pot preparar les seves simulacions, modificant paràmetres i

variables i navegant entre els diferents menús. Després, les simulacions es poden computar i, un cop s'ha acabat el càlcul, es poden observar alhora que es mostren l'evolució de certes quantitats en gràfiques que evolucionen amb la simulació.

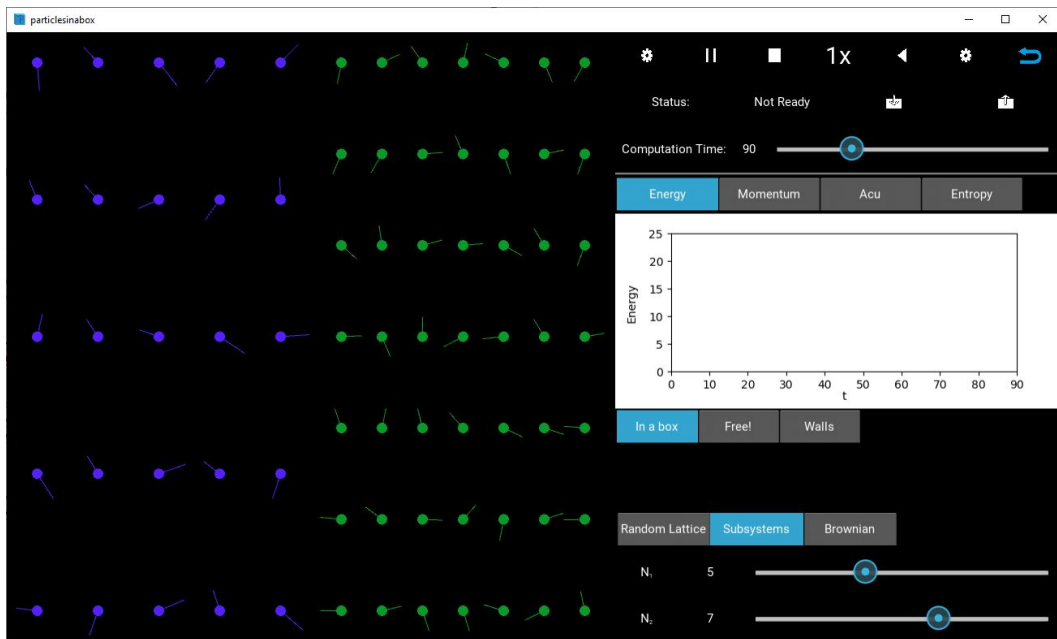


Figura 2.2: Imatge de la secció *Simulation*. En aquest cas es mostra una simulació del tipus *In a box* i *Subsystems*, amb 5×5 partícules del tipus 1 i 7×7 partícules del tipus 2.

Com es pot veure a la Figura 2.2 a la dreta, la simulació consta de 3 botons que controlen els límits de la simulació i 3 botons que controlen la distribució de les partícules. Els tres botons que controlen els límits de la simulació són *In a box*, *Free!* i *Walls*.

A *In a box*, les partícules interaccionen elàsticament amb els límits de la simulació; a *Free!* no interaccionen, les partícules tenen condicions periòdiques de contorn amb els límits; i finalment, a *Walls*, s'inclou una barrera elàstica amb un forat, amb la posició i la mida del forat variables per l'usuari.

Els tres botons que controlen la distribució de partícules són *Random Lattice*, *Subsystems* i *Brownian*. A *Random Lattice* es genera una graella de $N \times N$ partícules; a *Subsystems* es generen dues graelles de $N_1 \times N_1$ i $N_2 \times N_2$ partícules a temperatures diferents i a *Brownian* es genera una graella de $N \times N$ partícules petites més una partícula gran.

Així doncs, un cop escollit el tipus de simulació i els seus paràmetres, clicant el botó d'engranatge que es pot veure a la Figura 2.2 en la part superior, s'executa el càlcul de la simulació i es pot veure l'evolució del càlcul en el terminal. Un cop finalitzat el càlcul, es pot reproduir la simulació calculada amb el botó de *Play* de la Figura 2.2, i es poden observar l'evolució temporal de les diferents quantitats: l'energia, el moment, el moment acumulat i l'entropia.

Per més informació sobre com fer anar el programa i les seves funcionalitats anar a l'apèndix: Guia d'usuari.

2.1.2 Demo

La secció *Demo* es tracta d'una simplificació de la secció anterior, fent-la més entenedora i fàcil d'utilitzar que la primera: les simulacions ja venen computades prèviament i així, l'usuari pot simplement reproduir la simulació sense haver d'esperar el temps de càlcul que sol ser necessari.

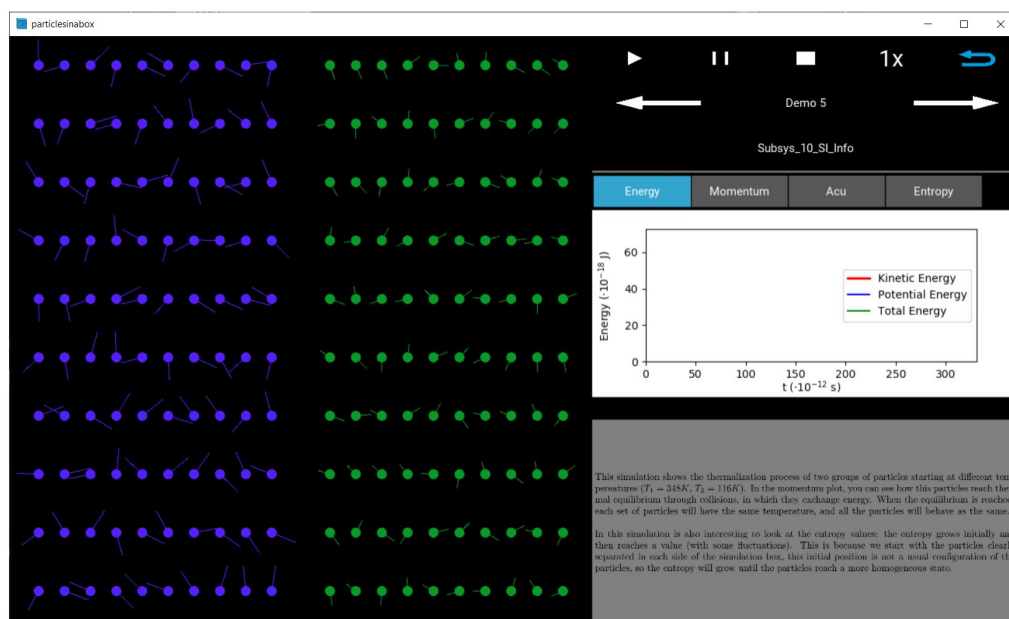


Figura 2.3: Imatge de la secció *Demo*.

Aquestes simulacions pregravades s'han escollit de tal manera que cadascuna mostri algun resultat interessant i diferent, permetent observar directament

els resultats que s'obtenen de les simulacions d'interaccions de partícules. A més, s'ha incorporat un petit text descriptiu a la part inferior a la dreta, que explica breument què es pot veure en la simulació carregada.

Un element interessant és que totes aquestes simulacions pregravades es poden obrir també des de la secció *Subsystems*, podent així observar els paràmetres inicials i corresponents a cada simulació. També, si l'usuari genera una simulació que troba interessant a l'apartat de *Subsystems*, pot decidir guardar-la i la podrà reproduir sempre que vulgui tant a la secció de *Subsystems* com a *Demo* (òbviament en aquest cas no hi haurà l'explicació teòrica que hi ha a les simulacions pregravades que venen amb el programa).

Tot això, fa que aquesta secció sigui especialment interessant per l'usuari que vol veure el comportament de les interaccions entre les partícules sense submergir-se en els detalls de la simulació (ni en el temps de càlcul, que pot ser llarg en alguns casos).

2.1.3 Game

La secció *Game* és una secció que pretenia incloure un element de joc al programa (vegeu Figura 2.4), es tracta d'un joc on l'usuari mou una barrera (on les partícules xoquen elàsticament) amb el control del teclat de l'ordinador. El joc tenia una inspiració en el "Dimoni de Maxwell", on l'usuari hauria d'actuar com el "Dimoni" i deixar passar només les partícules amb velocitat més gran, fent així que es redueixi l'entropia global i, per tant, violant el segon principi de la termodinàmica.

El principal repte a l'hora de fer aquesta secció era crear una simulació on es resolgués les equacions igual que en les altres seccions, però fer-ho en temps real. Tot i optimitzar amb profunditat gran part del codi d'aquesta secció, no vaig poder arribar al nivell de "fluïdesa" desitjat i la simulació funcionava només quan hi havia un nombre de partícules no molt elevat (però sí acceptable) però amb una temperatura significativament baixa, fent que la velocitat de les partícules no fos prou elevat com per donar l'element de joc que es buscava.

Així, finalment, vaig decidir abandonar aquesta secció i, tot i que no és una secció accessible per l'usuari, el codi sí que es troba en el programa i el càlcul en temps real funciona correctament.

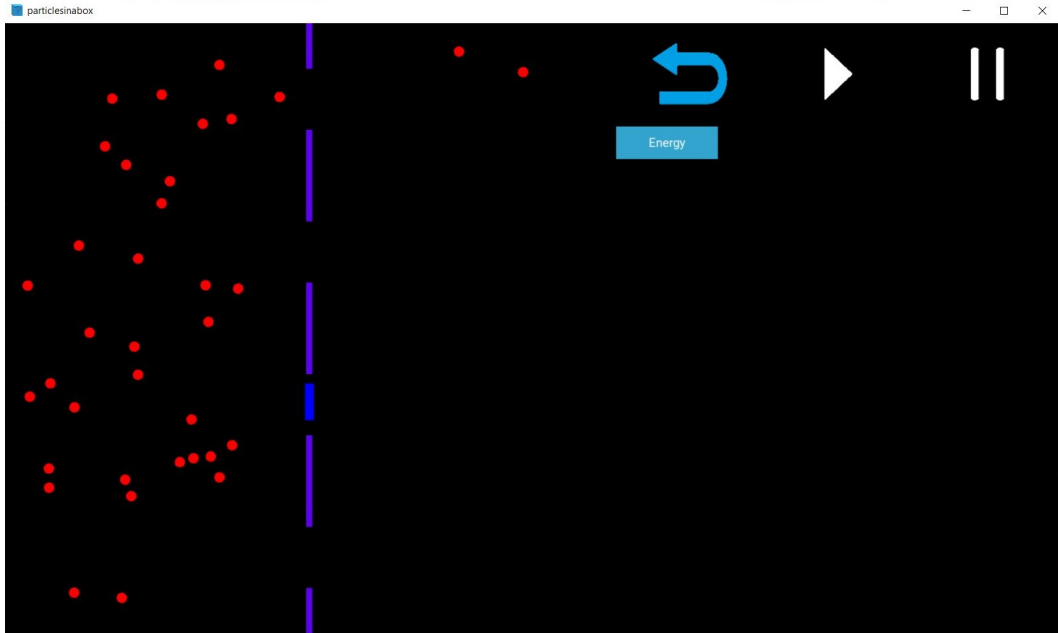


Figura 2.4: Imatge de la secció *Game*.

2.2 Equacions

Considerem cada partícula del nostre sistema independentment, un sistema en dues dimensions, interactuant amb la resta de partícules només a través d'un potencial V que dependrà únicament de la distància entre cada partícula considerada. Així doncs, cada partícula afectarà amb un potencial $V(\vec{r}_i, \vec{r}_j)$ sobre les altres $N - 1$ partícules.

El lagrangiana pel conjunt d' N partícules serà, doncs:

$$L = \sum_{i=1}^N \frac{1}{2} m_i \dot{x}_i^2 + \sum_{i=1}^N \frac{1}{2} m_i \dot{y}_i^2 - \sum_{i < j}^N V(\vec{r}_i, \vec{r}_j) \quad (2.1)$$

On

$$\dot{x}_i = \frac{\partial x_i}{\partial t}, \quad \dot{y}_i = \frac{\partial y_i}{\partial t} \quad (2.2)$$

Òbviamment escrivim el lagrangiana només amb les coordenades espacials x, y perquè estem tractant un problema en un camp amb dues dimensions.

Podem derivar les equacions del moviment de les partícules amb les equacions d'Euler-Lagrange:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = 0 \quad (2.3)$$

Sent q les coordenades generalitzades del sistema.

I obtenim:

$$\begin{aligned} \ddot{x}_i &= -\frac{1}{m_i} \sum_{j \neq i}^N \frac{\partial V(\vec{r}_i, \vec{r}_j)}{\partial x} \\ \ddot{y}_i &= -\frac{1}{m_i} \sum_{j \neq i}^N \frac{\partial V(\vec{r}_i, \vec{r}_j)}{\partial y} \end{aligned} \quad (2.4)$$

Que podem agrupar definint $\vec{r}_i = (x_i, y_i)$ i $\vec{f}_{ij} = -\vec{\nabla} V(\vec{r}_i, \vec{r}_j)$, on \vec{f}_i és la força que actua sobre la partícula i degut a la resta de partícules,

$$\ddot{\vec{r}}_i = \frac{1}{m_i} \sum_{j \neq i}^N \vec{f}_{ij} = \frac{1}{m_i} \vec{f}_i \quad (2.5)$$

2.2.1 Potencial $V(\vec{r}_i, \vec{r}_j)$

Aquest potencial $V(\vec{r}_i, \vec{r}_j)$ és únicament el potencial que descriu les interaccions entre les partícules, ja que la resta d'interaccions (amb els límits del sistema o, quan n'hi hagi, amb les barreres) seran xocs elàstics que vindran descrits per condicions de contorn i no amb un potencial.

Per descriure les interaccions entre partícules inicialment s'ha fet servir el potencial de Lennard-Jones, que pren la forma,

$$V(\vec{r}_i, \vec{r}_j) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^6 \right], \quad r_{ij} \equiv |\vec{r}_i - \vec{r}_j| \quad (2.6)$$

On σ es pot entendre com el radi d'esfera dura i ϵ com la profunditat del potencial de les partícules (inicialment tractem el cas en què les dues partícules tenen aquests dos paràmetres iguals).

Es tracta d'un potencial molt repulsiu a curtes distàncies $r_{ij} < \sigma$ i lleugerament atractiu per distàncies $r_{ij} > \sigma$, que s'ajusta força bé per descriure les propietats dels gasos, fent-lo adient pel nostre problema.

Com a valors experimentals per ϵ i σ hem fet servir els paràmetres del gas d'Argó [3]:

$$\begin{aligned} \epsilon &= 0.0103 \text{ eV}, & \sigma &= 3.405 \text{ \AA}, \\ M \simeq 0.04 \text{ kg/mol} &\implies m \simeq 6.6422 \cdot 10^{-26} \text{ kg} \end{aligned} \quad (2.7)$$

Un mètode molt utilitzat a l'hora de fer càlculs i que aquí també s'ha utilitzat és treballar en unitats reduïdes, això ens permet obtenir resultats propers a la unitat, simplificar les equacions i obtenir una certa universalitat en els resultats, ja que podem transportar els nostres resultats a qualsevol altre sistema utilitzant la corresponent ϵ , σ i M .

Així doncs aquestes unitats reduïdes seran:

$$\begin{aligned} E^* &= \epsilon, & r^* &= \sigma, & m^* &= m, \\ t^* &= \sqrt{\frac{m\sigma^2}{\epsilon}} \simeq 2.16 \cdot 10^{-12} \text{ s}, & T^* &= \frac{\epsilon}{k_B} \simeq 119.5 \text{ K} \end{aligned} \quad (2.8)$$

Sent E^* : la unitat d'energia, r^* : la unitat de longitud, m^* : la unitat de massa, t^* : la unitat de temps i T^* : la unitat de temperatura.

Per tant farem els càlculs amb aquestes unitats reduïdes i donarem el resultat amb aquestes unitats, però també s'ha afegit una opció per poder veure els resultats en unitats del Sistema Internacional, on simplement agafem els càlculs fets amb aquestes unitats reduïdes i hi apliquem els canvis d'unitats reduïdes a unitats del Sistema Internacional (seguint els valors de (2.8)).

Escrivint el potencial de Lennard-Jones en unitats reduïdes:

$$V(\vec{r}_i, \vec{r}_j) = 4 \left[\left(\frac{1}{r_{ij}} \right)^{12} - \left(\frac{1}{r_{ij}} \right)^6 \right] \quad (2.9)$$

I ara, podem obtenir $\vec{f}_{ij} = -\vec{\nabla}V(\vec{r}_i, \vec{r}_j)$:

$$\vec{f}_{ij} = \frac{48}{r_{ij}^2} \left[\left(\frac{1}{r_{ij}} \right)^{12} - \frac{1}{2} \left(\frac{1}{r_{ij}} \right)^6 \right] \vec{r}_{ij} \quad \text{amb} \quad \vec{r}_{ij} = (x_i - x_j, y_i - y_j) \quad (2.10)$$

És important entendre que les dues expressions anteriors pel potencial $V(\vec{r}_i, \vec{r}_j)$ i la força \vec{f}_{ij} , al ser un potencial (i conseqüentment una força) que actua per

parelles, només serà vàlid en el cas en què el radi d'esfera dura de la partícula i , σ_i sigui igual al de la segona, σ_j ; i igualment amb la profunditat del potencial ϵ_i i ϵ_j . En el cas en què les dues partícules no siguin iguals en algun d'aquests paràmetres, haurem de fer servir les regles de combinació.

Regles de Lorentz-Berthelot

Les regles de combinació pel potencial de Lennard-Jones s'anomenen regles de Lorentz-Berthelot [4], i ens permeten obtenir la σ i la ϵ d'interacció entre dues partícules amb algun d'aquests paràmetres diferents. Sent σ_i, ϵ_i els paràmetres de la partícula i ; σ_j, ϵ_j els paràmetres de la partícula j i $\sigma_{ij}, \epsilon_{ij}$ els paràmetres resultants d'interacció per les dues partícules:

$$\sigma_{ij} = \frac{\sigma_i + \sigma_j}{2} \quad \epsilon_{ij} = \sqrt{\epsilon_i \epsilon_j} \quad (2.11)$$

En la nostra simulació, totes les partícules tindran la mateixa ϵ , però sí que tindrem partícules de diferents mides, per tant farem ús d'aquestes regles per trobar la σ d'interacció.

Un exemple d'això és en la simulació pel moviment brownià, on hi ha un conjunt de partícules amb $R = 1$ (recordem que agafem com a unitats reduïdes els resultats experimentals del gas d'Argó) i una partícula gran amb $R = 5$, per exemple. Així, trobarem la σ d'interacció simplement fent $\sigma_{ij} = \frac{1+5}{2}$.

Considerarem que totes les partícules tindran la mateixa densitat, per tant si la unitat de massa $m = 1$ és per $R = 1$, la massa de la partícula gran serà $m_{gran} = \frac{R_{gran}^3}{R_1^3} = R_{gran}^3$

Potencial WCA

Com hem dit, en aquesta simulació inicialment es va començar utilitzant el potencial de Lennard-Jones, però es va acabar canviant per una variant d'aquest: el potencial WCA (Weeks-Chandler-Andersen) [6]. Es tracta de la part purament repulsiva del potencial de Lennard-Jones, negligint la part atractiva, és a dir, és el potencial de Lennard-Jones truncat a $r \geq 2^{1/6}\sigma$ (mínim del potencial L-J) i augmentat ϵ , de tal manera que el potencial queda continu. Així doncs, el potencial WCA té aquesta expressió:

$$u(r_{ij}) = \begin{cases} 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right] + \epsilon, & r \leq 2^{1/6}\sigma \\ 0, & r > 2^{1/6}\sigma \end{cases} \quad (2.12)$$

Es va utilitzar aquest potencial perquè així tenim un potencial purament repulsiu, i així només caldrà que calculem les forces entre partícules quan la distància entre elles sigui $r_{ij} \leq 2^{1/6}\sigma$, fent així prescindibles molts càlculs que abans havíem de fer.

Amb aquesta nova forma del potencial és interessant veure que la força \vec{f}_{ij} serà igual que pel potencial L-J però truncada per $r \geq 2^{1/6}\sigma$:

$$\vec{f}_{ij} = \begin{cases} \frac{48}{r_{ij}^2} \left[\left(\frac{1}{r_{ij}} \right)^{12} - \frac{1}{2} \left(\frac{1}{r_{ij}} \right)^6 \right] \vec{r}_{ij}, & r \leq 2^{1/6} \\ 0, & r > 2^{1/6} \end{cases} \quad \text{amb } \vec{r}_{ij} = (x_i - x_j, y_i - y_j) \quad (2.13)$$

Per tant, tenint ja l'expressió per la força \vec{f}_{ij} i amb l'equació del moviment (2.5) només necessitarem un mètode numèric per resoldre aquestes equacions.

2.3 Mètode numèric

El mètode que hem utilitzat per calcular les trajectòries de les partícules s'anomena l'algorisme de velocitat de Verlet (*Velocity-Verlet*), es tracta d'un algorisme similar a l'algorisme de Verlet però que arregla un dels seus problemes principals: en el *Velocity-Verlet* es fa un càlcul explícit de la velocitat, mentre que en el Verlet no. El fet de tenir la velocitat explícitament ens anirà molt bé a l'hora de posar condicions de contorn pels límits del sistema.

Podem obtenir l'algorisme de velocitat de Verlet de la següent manera: primer, desenvolupem en sèrie de Taylor la posició $x(t + \delta t)$ (escrivim x genèricament, el mateix s'aplica a les dues components x i y):

$$x(t + \delta t) = x(t) + \delta t \frac{dx(t)}{dt} + \frac{(\delta t)^2}{2} \frac{d^2x(t)}{dt^2} + O((\delta t)^3) \quad (2.14)$$

Que podem escriure més senzillament definint el pas de temps que utilitzarem en el càlcul: $h \equiv \delta t$ i també $\dot{x} = \frac{dx(t)}{dt}$, $\ddot{x} = \frac{d^2x(t)}{dt^2}$, ... Amb això podem reescriure-ho:

$$x(t+h) = x(t) + h\dot{x}(t) + \frac{h^2}{2}\ddot{x}(t) + O(h^3) \quad (2.15)$$

Anomenant la velocitat $v = \dot{x}(t)$ i també expandint-ho en sèries de potències:

$$v(t+h) = v(t) + h\dot{v}(t) + \frac{h^2}{2}\ddot{v}(t) + O(h^3) \quad (2.16)$$

Anomenem l'acceleració $a = \dot{v}(t) = \ddot{x}(t)$ i també l'expandim (en aquest cas expandint-la a ordre h^2 serà suficient, ja que introduïrem aquest resultat a $\frac{h^2}{2}\ddot{v}(t) = \frac{h^2}{2}\dot{a}(t)$, i només necessitem una aproximació d'ordre h^3 d'aquesta quantitat).

$$a(t+h) = a(t) + h\dot{a}(t) + O(h^2) \quad (2.17)$$

Com que volem el valor de $\frac{h^2}{2}\dot{a}(t)$ per introduir-ho a l'eq. (2.16), reordenem l'expressió anterior i multipliquem per $\frac{h^2}{2}$:

$$\frac{h^2}{2}\dot{a}(t) = \frac{h}{2}(a(t+h) - a(t)) + O(h^3) \quad (2.18)$$

Per tant, ja ho podem introduir a l'eq. (2.16) i ja tenim una expressió pel càlcul de la velocitat:

$$v(t+h) = v(t) + ha(t) + \frac{h}{2}(a(t+h) - a(t)) + O(h^3) \quad (2.19)$$

$$\implies \boxed{v(t+h) = v(t) + \frac{h}{2}(a(t+h) + a(t)) + O(h^3)} \quad (2.20)$$

I per l'expressió de la coordenada general x , agafem l'expressió de l'eq. (2.15):

$$\boxed{x(t+h) = x(t) + hv(t) + \frac{h^2}{2}a(t) + O(h^3)} \quad (2.21)$$

I finalment, en el nostre cas particular, podem substituir $a = -\frac{1}{m_i}\vec{f}_i$ (de l'eq. (2.5) i obtindrem les expressions per fer el càlcul numèric de la partícula i -èsima:

$$\begin{cases} \vec{x}_i(t+h) = (x_i(t+h), y_i(t+h)) = \vec{x}_i(t) + h\vec{v}_i(t) - \frac{h^2}{2}\frac{\vec{f}_i(t)}{m_i} + O(h^3) \\ \vec{v}_i(t+h) = \vec{v}_i(t) - \frac{h}{2}\left(\frac{\vec{f}_i}{m_i}(t+h) + \frac{\vec{f}_i}{m_i}(t)\right) + O(h^3) \end{cases} \quad (2.22)$$

I l'esquema bàsic d'aquest algorisme serà:

1. Calcular $\vec{x}(t+h) = \vec{x}(t) + h\vec{v}(t) - \frac{h^2}{2} \frac{\vec{f}(t)}{m}$.
2. Calcular $\vec{f}(t+h)$ a partir de la nova posició $\vec{x}(t+h)$.
3. Calcular $\vec{v}(t+h) = \vec{v}(t) - \frac{h}{2m} (\vec{f}(t+h) + \vec{f}(t))$.

On aquí no posem el subíndex i perquè ens estem referint a tot el conjunt de partícules. Per fer això, farem ús de les matrius que ens proporciona la llibreria *NumPy* del Python.

2.3.1 Càlcul matricial

Per referir-nos a totes les partícules del nostre sistema i fer càlculs amb cadascuna d'elles, utilitzarem les matrius del *NumPy*, molt més eficaces que utilitzar bucles repetidament.

Primer de tot, tindrem dues matrius $N \times 1$, una que guardarà el radi de cadascuna de les partícules i l'altra que en guardarà la seva massa. També generarem dues altres matrius $N \times 1$ que contindrà les posicions i les velocitats de cada partícula en cada instant.

$$R = \begin{pmatrix} R_1 \\ R_2 \\ \dots \\ R_N \end{pmatrix}, \quad m = \begin{pmatrix} m_1 \\ m_2 \\ \dots \\ m_N \end{pmatrix}, \quad r = \begin{pmatrix} \vec{r}_1 \\ \vec{r}_2 \\ \dots \\ \vec{r}_N \end{pmatrix} \text{ sent } \vec{r} = (x, y) \quad (2.23)$$

$$v = \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \dots \\ \vec{v}_N \end{pmatrix} \text{ sent } \vec{v} = (v_x, v_y) \quad (2.24)$$

Després, a partir d'aquesta matriu de posicions, generarem una matriu $N \times N$ que ens guardarà la diferència entre cada partícula:

$$d = \begin{pmatrix} \vec{r}_1 - \vec{r}_1 & \vec{r}_1 - \vec{r}_2 & \dots & \vec{r}_1 - \vec{r}_N \\ \vec{r}_2 - \vec{r}_1 & \vec{r}_2 - \vec{r}_2 & \dots & \vec{r}_2 - \vec{r}_N \\ \dots & \dots & \dots & \dots \\ \vec{r}_N - \vec{r}_1 & \vec{r}_N - \vec{r}_2 & \dots & \vec{r}_N - \vec{r}_N \end{pmatrix} \quad (2.25)$$

A partir d'aquesta matriu podrem fàcilment trobar la distància entre les parelles de partícules, simplement fent: $r^2 = (d[0])^2 + (d[1])^2$, sent $d[0]$ la

component x d'aquesta matriu d i $d[1]$ la component y .

Tenint aquestes matrius ja podrem fer el càlcul de les forces de manera matricial, optimitzant el rendiment del programa.

Una altra implementació que es va afegir per poder millorar el rendiment del programa és el fet de calcular la força entre partícules només per les M partícules més properes, evitant haver de fer el càlcul per totes les partícules a cada iteració del bucle. Per fer això, es va crear una matriu $N \times M$ que guarda el "nom" (la "id") de les M partícules més properes, per cada una de les N partícules:

$$\begin{pmatrix} \text{id}(1^{\text{a}} \text{ més propera a la partícula 1}) & \text{id}(2^{\text{a}} \text{ més propera a la partícula 1}) & \dots & \text{id}(M^{\text{a}} \text{ més propera a la partícula 1}) \\ \text{id}(1^{\text{a}} \text{ més propera a la partícula 2}) & \text{id}(2^{\text{a}} \text{ més propera a la partícula 2}) & \dots & \text{id}(M^{\text{a}} \text{ més propera a la partícula 2}) \\ \dots & \dots & \dots & \dots \\ \text{id}(1^{\text{a}} \text{ més propera a la partícula N}) & \text{id}(2^{\text{a}} \text{ més propera a la partícula N}) & \dots & \text{id}(M^{\text{a}} \text{ més propera a la partícula N}) \end{pmatrix}$$

I aquesta matriu es calcularà cada cert nombre de passos, estalviant-nos així una gran quantitat de càlculs que aporten poc al sistema, ja que a partir d'una certa distància el valor de la força de Lennard-Jones cau molt ràpidament.

En el programa, s'ha triat $M = \sqrt{N}$, ja que s'ha vist que és un valor que funciona bé per tot el rang de partícules possibles.

A continuació explicarem com calculem l'entropia. La resta de magnituds i la inicialització del sistema estan explicades en el document de l'Arnau Jurado [7], ja que el procediment que s'ha utilitzat per fer aquests càlculs és el que s'explica en el seu document.

2.4 Entropia

Per poder descriure el comportament de l'entropia en el sistema, s'ha optat per seguir el mètode descrit en l'article [5]. Es tracta d'una aplicació de l'entropia de Shannon, es discretitza el sistema en M bins (d'igual tamany), i per cada bin j , $j = 1, \dots, M$, s'avalua el nombre de partícules $n_{j,c}$ de l'espècie c , $c = 1, 2, \dots, C$. Amb això, la probabilitat que una partícula de l'espècie c estigui dintre del bin j es pot calcular dividint $n_{j,c}$ pel total de partícules del sistema. Si el nombre total de partícules varia en funció de l'espècie, també haurem d'incloure el factor de proporcionalitat P_c :

$$p_{j,c} = \frac{\frac{n_{j,c}}{P_c}}{\sum_{i=1}^M \sum_{c=1}^C \frac{n_{j,c}}{P}} \quad \text{sent} \quad P_c = \frac{n_c}{\sum_{c=1}^C n_c} \quad (2.26)$$

Amb aquestes probabilitats i amb la definició d'entropia de Shannon,

$$S = - \sum_{j=1}^M \sum_{c=1}^C p_{j,c} \ln p_{j,c} \quad (2.27)$$

Com que el nostre sistema de partícules no és gaire dens, per evitar que hi hagi masses *bins* sense cap partícula, s'ha calculat aquestes probabilitats cada 2 passos del mètode numèric. Així, es guarda quantes partícules han passat pel *bin* j durant aquests passos i després es fa el càlcul de l'entropia d'aquell interval.

Per tant, per implementar aquest càlcul de l'entropia s'ha afegit un comptador que guarda a cada pas en quin *bin* està cada partícula.

3. Desenvolupament del projecte

3.1 Metodologia

Un cop vam tenir clara la idea original del nostre projecte, cada alumne hem seguit un treball autònom intercalat amb una reunió setmanal amb tots els alumnes i professors (les reunions van ser inicialment presencials, fins que a causa del confinament les vam seguir fent des de llavors *online*), on explicàvem el progrés de l'última setmana, les dificultats amb què ens hem trobat i discutíem el camí a seguir en els dies vinents.

A més, aquestes reunions servien com a eina per donar i rebre *feedback* amb els altres companys, permetent obtenir un diferent punt de vista i millorar el treball. També durant la setmana manteníem comunicació via *email*.

Per documentar el nostre progrés, cada setmana fèiem un recull del progrés de la setmana i, abans de la reunió, pujàvem el codi al *Github*, en el repositori del ClassicalLabUb [2], en el meu cas. El *Github* ha estat una eina molt útil en el nostre treball, permetent tenir molt accessible l'historial d'actualitzacions del codi a més de fer-lo públic pels companys i professors.

3.2 Idea original

El principal objectiu d'aquest projecte era crear un programa que servís per divulgar la mecànica estadística i mostrar-ne els efectes més rellevants. Veient els anteriors programes desenvolupats per exalumnes d'aquestes pràctiques, vaig decidir partir del programa fet per l'Arnau Jurado Romero (Figura 3.1), en què l'usuari pot col·locar partícules amb una determinada velocitat i veure'n l'evolució al llarg del temps, podent observar com varia l'energia del sistema i el moment de les partícules.

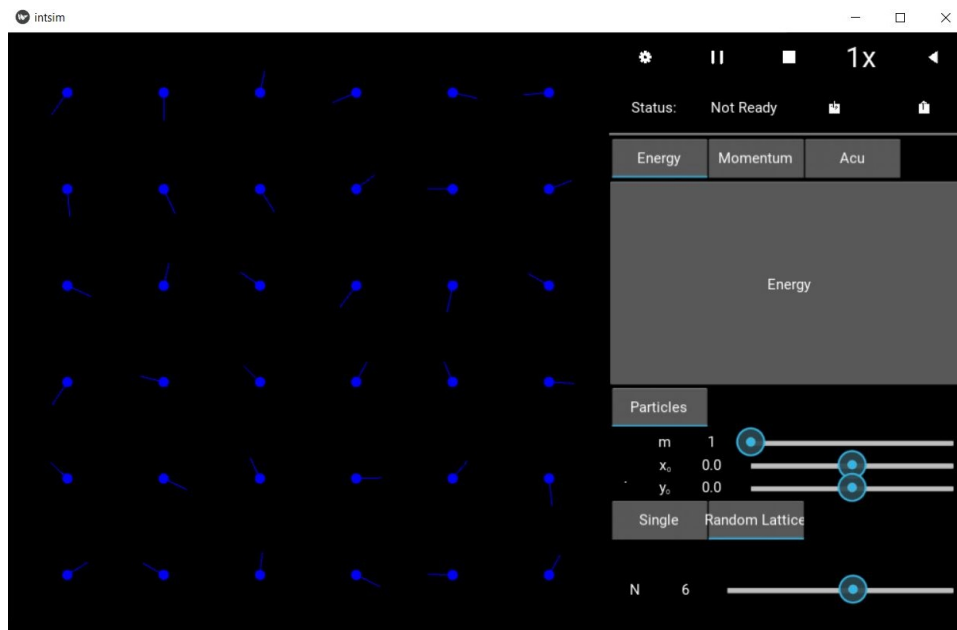


Figura 3.1: Programa *ClassicalLabUB* desenvolupat per l'Arnau Jurado.

Així doncs, vaig decidir continuar aquest programa per donar-li un enfocament des de la mecànica estadística, tenint els elements fonamentals ja desenvolupats (com la interfície, la resolució de les equacions de la interacció entre partícules i el càlcul de l'energia i moment) i podent-me centrar a implementar nous elements al programa o bé millorar alguns d'aquests elements, fent així un programa més complet i acabat.

3.3 Rendiment

Des del començament, una de les meves intencions era poder millorar el rendiment, ja que les simulacions tardaven un temps de càlcul força significatiu. Al llarg d'aquestes pràctiques s'ha anat implementant elements per tal de millorar-ne el rendiment, en aquesta secció els explico breument:

3.3.1 *Velocity-Verlet*

El programa inicialment utilitzava el mètode de Verlet per resoldre les equacions de manera numèrica. El problema de fer-ho amb aquest mètode és que no et permet fer un càlcul explícit de la velocitat, fent així impossible posar

condicions de contorn pels límits del sistema, fent necessari la utilització d'un potencial del tipus Woods-Saxon (per més informació anar al dossier de pràctiques de l'Arnau Jurado, [7]).

El problema d'imposar els límits del sistema d'aquesta manera és que, a part de crear possibles problemes quan les partícules van prou de pressa (havent-hi la possibilitat de què puguin travessar el potencial), hem d'estar calculant la força de les 4 barreres per a cada partícula a cada instant de temps, fent així molts càlculs que ens podríem estalviar amb unes condicions de contorn del tipus: $v_x = -v_x$ quan $x > L$

Així, com que el mètode de Velocity-Verlet ens permet fer un càlcul explícit de la velocitat (mètode està explicat en detall a la subsecció 2.3), podem imposar aquestes condicions de contorn i ja no necessitem aquest potencial extern, fent que ara l'únic potencial present en el sistema és el que creen les mateixes partícules.

Aquest mètode va reduir aproximadament en un 20% el temps de càlcul de les simulacions.

3.3.2 *Close particle list*

Una de les causes principals que fan que aquestes simulacions siguin tan lentes és degut al fet que per cada partícula, s'ha de calcular la força amb cada una de la resta de partícules, fent que a l'augmentar el nombre de partícules el nombre de càlcul de forces també creixi molt.

El mètode que es va aplicar per reduir el nombre de càlculs per partícula és crear una llista de les M partícules més properes per a cada una de les N partícules (vegeu subsecció 2.3.1) de tal manera que a l'hora de mirar les forces que actuen a la partícula i -èsima, només mirarem la força creada per aquestes M partícules més properes a i . Cada cert nombre de passos de temps s'actualitza aquesta llista.

Aquest mètode va millorar molt significativament el rendiment de les simulacions, reduint el temps de càlcul fins a més de la meitat. També és important el fet que al tenir un potencial que es fa zero per $r > r_c$ (potencial WCA, explicat a la subsecció 2.2.1), fa que calcular només la força per les partícules més properes sigui especialment vàlid, fent que aquest mètode no introdueixi error a la conservació de l'energia.

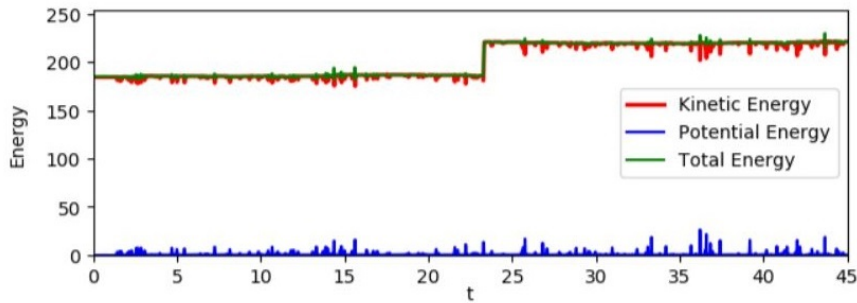


Figura 3.2: Exemple de la discontinuïtat de l'energia produït per un nombre M de partícules massa inferior (aquí $M=2$).

Es podria donar un error de conservació de l'energia si la llista de partícules properes és o massa petita o bé no s'actualitza prou freqüentment, fent que al moment que s'actualitza de nou, hi hagi alguna partícula més propera amb $r < r_c$, això dóna lloc a una discontinuïtat a l'energia que és molt fàcil d'identificar (Figura 3.2), per tant calibrant bé el nombre de partícules M a la llista i el temps d'actualització podem aconseguir no tenir cap problema d'aquest tipus.

3.3.3 Numba

El Numba [8] és una llibreria de codi obert de Python que permet accelerar molt codi que involucri iteracions i vectors i funcions de la llibreria NumPy. Com que aquest és el cas del nostre programa, ens serà molt útil.

El Numba funciona de tal manera que s'ha de definir certes funcions del nostre codi on actuarà, aquestes funcions han de ser independents: no poden cridar altres funcions on no actuï el Numba i tota la informació que reben i emeten es passa com a arguments.

Això va fer que s'hagués de fer un canvi radical de l'estructura de les funcions del programa: es va modificar de tal manera que les funcions on actua el Numba només fan un pas de temps, deixant la iteració per cada pas de temps a una funció exterior on no actua el Numba i que és l'encarregada de cridar la resta de funcions.

En total es va aplicar al Numba a les següents funcions:

- `dLJverlet()` i `LJverlet()`, funcions que a partir de les posicions de les partícules retornen les forces i l'energia, respectivament, produïda pel potencial WCA.
- `close_particles_list()`, funció que crea la llista de les M partícules més properes per cada partícula.
- `fv()`, funció encarregada d'iterar entre les partícules i calcular la força i l'energia entre les partícules que calgui.
- `vel_verlet()`, és la funció responsable d'aplicar les condicions de contorn.

El Numba va millorar notablement el temps d'execució, reduïnt gairebé a la meitat el temps de càlcul. És important veure que a conseqüència de la implementació del Numba, la primera vegada (des que s'ha obert el programa) que es fa una simulació, es necessita uns segons perquè el Numba es "prepari" abans no es posa a calcular, això només ho necessita fer quan es fa el primer càlcul, i quan es fa una nova simulació ja actua normalment.

3.4 Accessibilitat i estètica

A part de les millores de rendiment que s'acaben d'exposar, també s'ha invertit temps a millorar el programa pel que fa a l'accessibilitat i l'estètica.

La implementació d'un menú principal que et permet triar si anar a una pàgina on crear les teves pròpies simulacions (*Simulation*, veure a la figura 2.1) o una pàgina on simplement les pots observar (*Demo*, veure a la figura 2.1), fa que el programa sigui molt més accessible a usuaris nous. En particular, s'ha pensat per què professors que ho trobin pertinent puguin mostrar alguns resultats d'aquestes simulacions als alumnes. És per això que s'ha creat un apèndix que conté la guia bàsica d'usuari, que recull les diferents funcionalitats del programa de manera breu.

S'ha millorat l'estètica del programa fent servir un conjunt de colors blau, gris, negre i blanc pels botons i les diferents parts del menú que donen un aspecte força elegant. També ha millorat molt a l'aparença del programa la portada del menú principal, que es pot veure a la Figura 3.3. Aquestes dues imatges les ha dibuixat el meu germà Pau Vallès Muns, a qui l'hi agraeixo el favor d'haver-les dissenyat, milloren significativament l'aparença del programa i queden molt professionals.

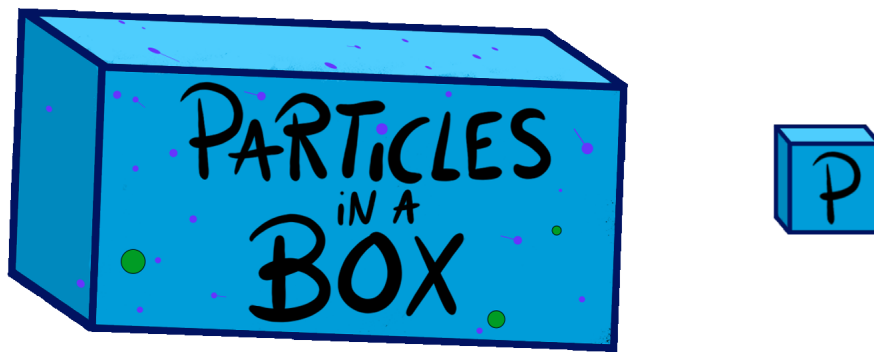


Figura 3.3: Les dues imatges que il·lustren el programa, la primera és la portada que es troba en el menú principal i la segona és la icona que apareix per obrir el programa i en la barra de tasques de l'ordinador. Les dues les ha dissenyat en Pau Vallès Muns.

4. Valoració i conclusions

Per concloure aquest document m'agradaria acabar donant una valoració personal, voldria agrair als tutors d'aquestes pràctiques per haver-nos donat les eines i coneixements per dur a terme aquest treball, han estat unes pràctiques molt interessants amb molt bona comunicació entre els professors i nosaltres, els alumnes, fent així un entorn de treball molt agradable. He pogut desenvolupar plenament els coneixements de la mecànica estadística des d'un punt de vista computacional i he gaudit molt creant un programa que pot servir per divulgar aquests elements de la física a gent que en tingui interès. He obtingut molt coneixement que estic segur que aplicaré en el meu futur professional i m'emporto una experiència molt positiva d'aquestes pràctiques.

Referències

- [1] *QuantumLabUb*, <https://github.com/brunojuliana/quantumlabUB>
- [2] *ClassicalLabUb*, <https://github.com/brunojuliana/classicallabUB>
- [3] L.A Rowley, D Nicholson, N.G Parsonage, *Monte Carlo grand canonical ensemble calculation in a gas-liquid transition region for 12-6 Argon*, Journal of Computational Physics, Volume 17, Issue 4, 1975, ISSN 0021-9991, [https://doi.org/10.1016/0021-9991\(75\)90042-X](https://doi.org/10.1016/0021-9991(75)90042-X).
- [4] Lorentz, H.A. (1881), *Ueber die Anwendung des Satzes vom Virial in der kinetischen Theorie der Gase*. Ann. Phys., 248: 127-136. <https://doi.org/10.1002/andp.18812480110>
- [5] Camesasca, Marco; Kaufman, Miron; Zlockzower, Ica Manas. *Quantifying Fluid Mixing with The Shannon Entropy* (2006). Physics Faculty Publications. 238. <https://doi.org/10.1002/mats.200600037>
- [6] J. D. Weeks, D. Chandler, and H. C. Andersen, J. Chem. Phys. 54, 5237 (1971)
- [7] Dossier de pràctiques d'empresa de l'Arnau Jurado Romero. <http://www.ecm.ub.es/~bruno/works/PE-JuradoRomero.pdf>
- [8] *Numba*, <http://numba.pydata.org/>

Apèndix: Guia d'usuari

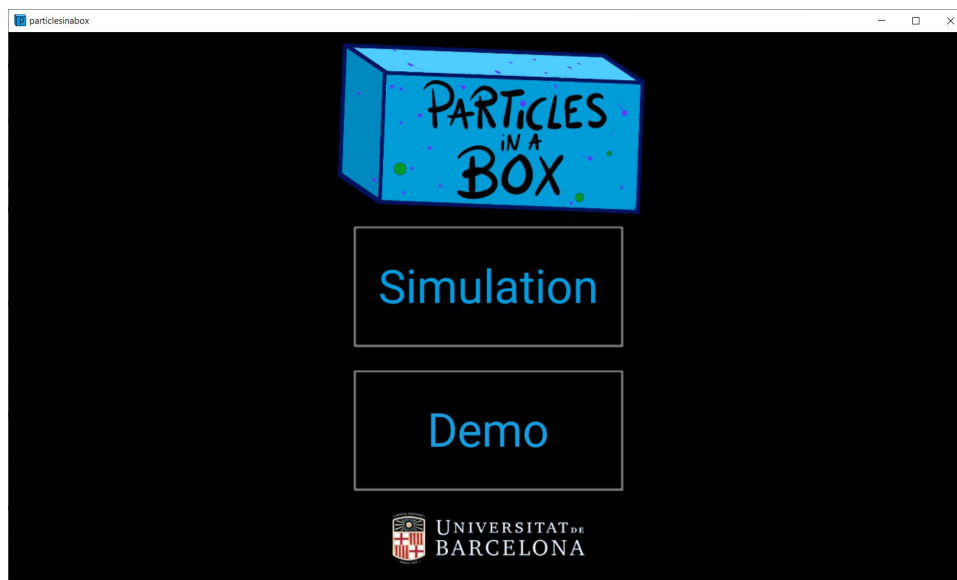


Figura A.1: Menú, pàgina per defecte en obrir el programa.

Aquesta guia d'usuari pretén explicar de manera clara els diferents elements del programa. Es divideix en dues parts, en una s'explica com fer servir la pàgina de *Demo* i en l'altre la de *Subsystems*.

1. Demo

Aquesta és la pàgina recomanada en fer servir el programa per primera vegada, o per usuaris que no es vulguin endinsar en personalitzar les simulacions. Aquesta pàgina permet visualitzar simulacions carregades a més de mostrar un petit resum que explica el que està passant.

Per poder fer servir correctament aquesta pàgina, s'ha de descarregar les simulacions que hem generat amb una breu descripció del que s'està mostrant, o bé l'usuari ha de guardar les simulacions que faci en la pàgina *Subsystems* (això s'explicarà en la secció de *Subsystems* amb més detall). Si el programa detecta que no hi ha cap simulació guardada, s'obrirà un *Pop-up* i avisarà que no es pot accedir a aquesta pàgina.

Com es pot veure a la figura A.2, la utilització d'aquesta pàgina és força intuïtiva: la pàgina es divideix en dues parts: la interacció de les partícules

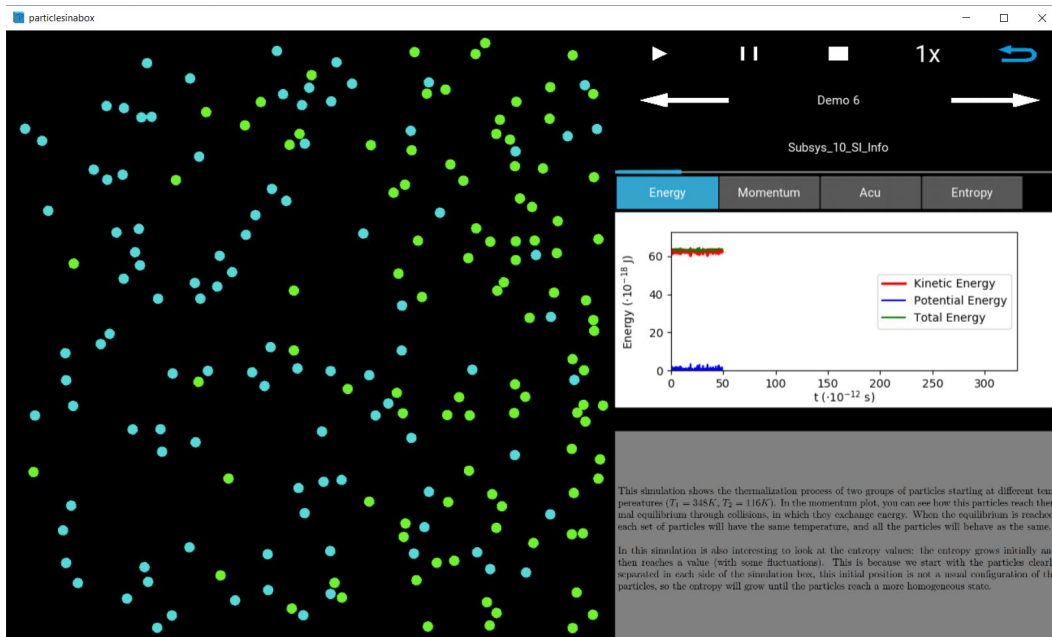


Figura A.2: Imatge de la pàgina *Demo* durant la reproducció d'una simulació on interactuen dues espècies de partícules amb temperatures inicials diferents.

a l'esquerra i el control i visualització de l'evolució de quantitats a la dreta.

A l'esquerra es mostra una simulació quadrada on interactuen partícules en moviment, les partícules xoquen les unes amb les altres i, depenent de les condicions de contorn seleccionades, poden xocar o bé travessar els límits de la simulació. Aquesta part esquerra és només per visualitzar i no s'hi pot interactuar amb el ratolí.

A la dreta es troba la part controlable per l'usuari. Els cinc botons de la part superior:

- *Play*, reproduueix la simulació en què ens trobem.
- *Pause* atura la simulació, permetent tornar a continuar la reproducció de la simulació clicant de nou *Play*.
- *Stop*, atura tota la simulació i la reinicialitza.
- Velocitat de reproducció, clicant et permet reproduir la simulació al ritme normal ($\times 1$), $\times 2$ (dues vegades més ràpid), $\times 5$ (cinc vegades més ràpid) o $\times 10$ (deu vegades més ràpid).

- El botó que et retorna al menú principal.

A sota d'aquests cinc botons es troben les dues fletxes que et permeten navegar a través de les simulacions carregades, mostrant el nom de la simulació entre les dues fletxes.

A baix d'aquestes, es troba la barra temporal que ens indica amb color blau en quin punt temporal ens trobem en la reproducció.

Després, es troba els 4 botons que permeten visualitzar l'evolució de diferents quantitats:

- *Energy*, aquesta gràfica mostra l'evolució de l'energia total del sistema, l'energia cinètica total i l'energia potencial total. Es pot observar petites discrepàncies en la conservació de l'energia, aquestes disminueixen disminuint el pas de temps computació.
- *Momentum*, aquí es mostra la distribució de moments de les partícules (com es pot veure en la Figura A.3). També es mostra la temperatura corresponent al moment de les partícules i la corba de distribucions (distribució de Maxwell-Boltzmann) esperades per a la temperatura que es troben, la distribució de moments correspondrà a aquesta corba en l'equilibri.
- *Accu*, en aquesta gràfica, a partir d'un cert temps, es mostra el moment de les partícules acumulat al llarg del temps, es pot veure com també segueixen una distribució de moments de Maxwell-Boltzmann.
- Si la simulació és del tipus *Random Lattice* o *Subsystems* es mostrarà *Entropy*, on es pot veure l'evolució del càlcul de l'entropia. Qualitativament, podem entendre que l'entropia creixerà fins que s'arribi a l'equilibri, on després s'estabilitzarà.
- Si en canvi la simulació és del tipus *Brownian*, es mostrarà el desplaçament quadràtic mitjà, $\langle |x(t) - x(0)|^2 \rangle$, de la partícula gran *Browniana*. Aquí es pot veure com aquest desplaçament quadràtic mitjà hauria de ser proporcional a al temps ($\langle |x(t) - x(0)|^2 \rangle = 2Dt$).

2. Simulation

Aquesta és la pàgina on es poden dur a terme les simulacions personalitzades i amb els paràmetres que un vulgui. L'usuari pot preparar les simulacions

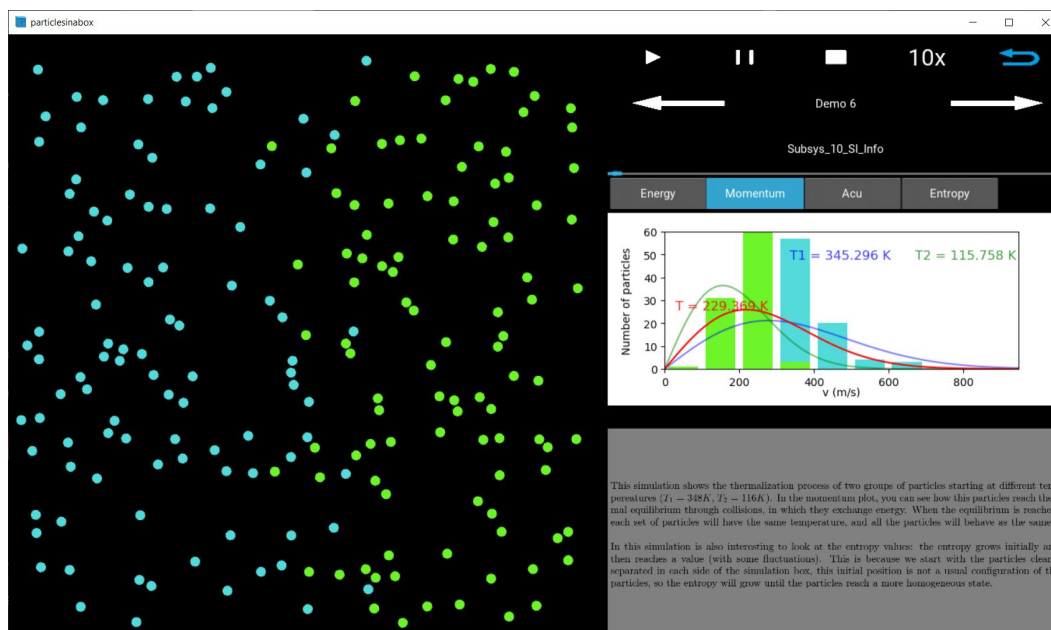


Figura A.3: Imatge durant la reproducció d'una simulació a la pàgina *Demo*, es pot veure a la dreta com a *Momentum* es mostra la distribució de moments i la temperatura de les partícules.

modificant paràmetres i navegant pels diferents menús. Després, les simulacions s'han de computar, i un cop acabat el càlcul, es poden observar de la manera que s'observaven a la pàgina de *Demo*. El fet que s'hagi d'esperar un temps de càlcul de les simulacions i el grau de complexitat que presenta aquesta pàgina, fa que sigui menys atractiva per l'usuari que simplement vol veure algunes simulacions sense haver d'esperar el temps de càlcul, però és molt interessant per poder observar casos particulars.

Com es pot veure en la Figura A.4, les gràfiques on es mostren els valors calculats són les mateixes que en el cas de la pàgina *Demo*. Hi ha un total de sis botons que permeten controlar el tipus de simulació, tres controlen les condicions de contorn i tres la distribució de partícules, anem a veure'ls:

Condicions de contorn

El botó *In a box* és el més intuïtiu, les partícules interaccionen elàsticament amb els límits de la simulació. Amb el botó *Free!* les partícules no interaccionen amb els límits, sinó que tenen condicions periòdiques de contorn amb ells, fent que al travessar un límit, la partícula surti pel límit oposat.

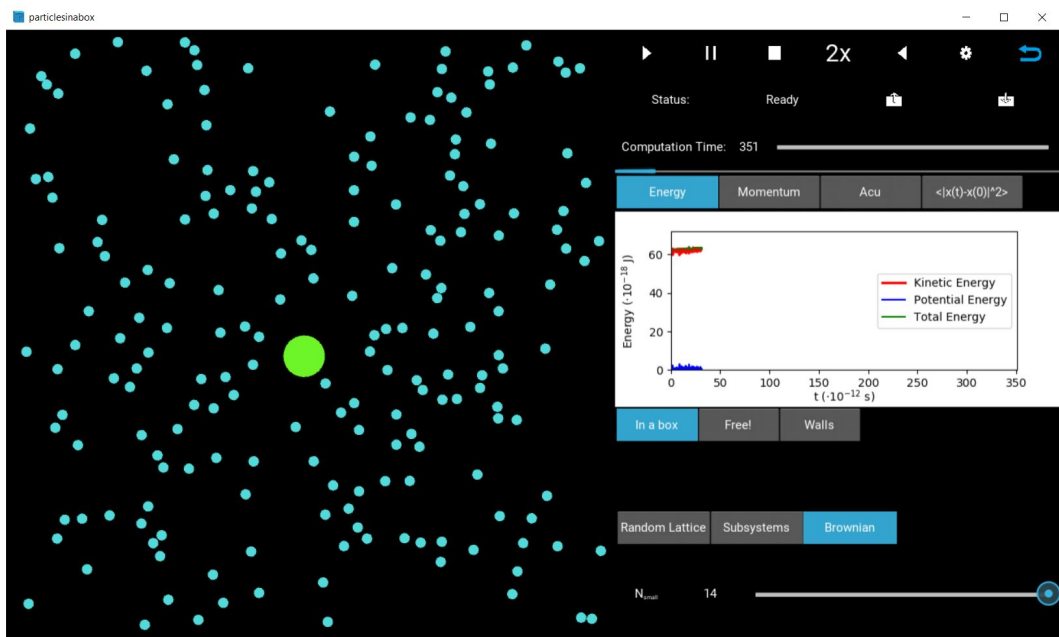


Figura A.4: Imatge de la pàgina *Simulation* durant la reproducció d'una simulació on interactuen diverses partícules amb una de més gran.

Finalment, el botó *Walls*, inclou una barrera elàstica amb un forat, amb la posició i el tamany del forat variables per l'usuari a través de dos *sliders*.

Distribució de les partícules

Amb el botó *Random Lattice*, es genera una graella de $N \times N$ partícules, sent N variable per l'usuari a través d'un *slider* que apareix a la part inferior. Amb el botó *Subsystems* es generen dues graelles de $N_1 \times N_1$ i $N_2 \times N_2$ partícules a temperatures diferents, sent N_1 i N_2 variable per l'usuari a través de dos *sliders*. Per últim, a *Brownian* (com el que es mostra en la Figura A.4), apareix una partícula més gran (i, per tant, amb més massa) que permet observar el moviment brownià que experimenta aquesta partícula a través de col·lisions amb la resta.

A part d'això, clicant el botó d'engranatge de la part superior a la dreta que es pot veure a la Figura A.4, s'obre un menú anomenat *Advanced Settings*. Aquest menú, que es pot veure en la Figura A.5, permet a l'usuari retocar molts paràmetres que afecten la simulació:

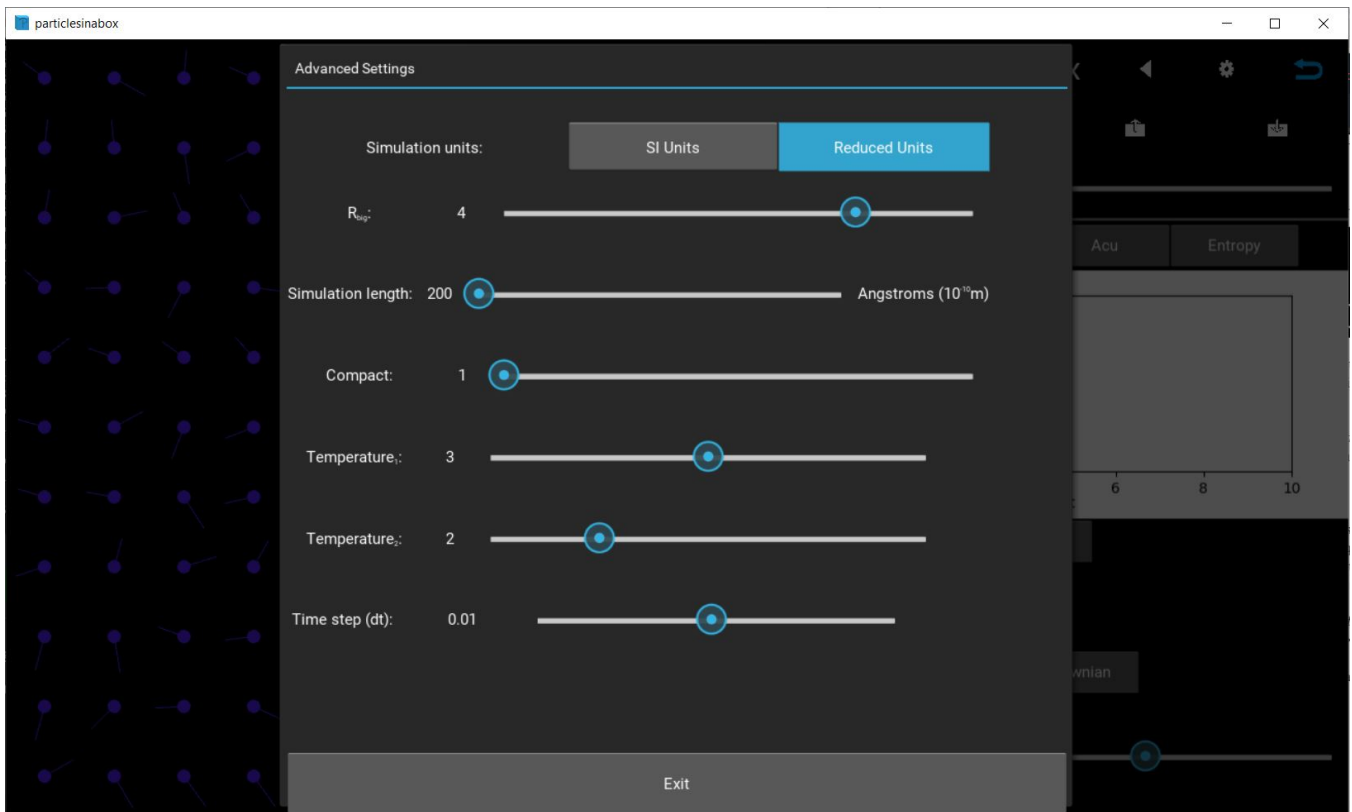


Figura A.5: Imatge del menú *Advanced Settings*, on es pot veure tots els paràmetres variables per l'usuari.

- *SI Units/Reduced Units*. Amb aquests dos botons es pot escollir si el resultat que es mostra en les gràfiques és mostrat en unitats del Sistema Internacional o bé unitats reduïdes. Resulta interessant mostrar les magnituds en unitats del Sistema Internacional per poder obtenir una idea de les quantitats que s'estan tractant. Les unitats reduïdes són útils per entendre el funcionament qualitatiu del sistema sense molestar-se amb les unitats.
- R_{Big} . Aquest *slider* permet modificar la mida de la partícula gran que apareix en el mode *Brownian*. El nombre indica quantes vegades és més gran el radi de la partícula gran en relació amb el radi de les partícules normals.
- *Simulation length*. Amb aquest *slider* es pot ajustar la longitud d'un costat del recipient que conté les partícules (en Àngstroms), això fa que a l'augmentar aquesta longitud, les partícules es vegin més petites, i

per tant, n'hi càpiguen més dintre de la simulació.

- *Compact*. Modificar aquest paràmetre fa que les partícules en el mode *Random Lattice* es compactin, que la seva posició inicial sigui més propera entre elles.
- *Temperature 1*. Modifica la temperatura inicial de les partícules que es mostren en *Random Lattice*, les partícules petites de *Brownian* i l'espècie de partícules 1 en el mode *Subsystems*.
- *Temperature 2*. Modifica la temperatura inicial de les partícules de la segona espècie del mode *Subsystems*.
- *Time step (dt)*. Modifica el pas de temps amb què es calculen les simulacions. Valors més petits garanteixen una millor conservació de l'energia, però a canvi d'augmentar substancialment el temps de computació. El valor que es dona per defecte és un bon balanç entre un temps no molt elevat de càlcul i una prou bona conservació de l'energia.

Una altra característica important del programa és el botó de *Load* (en la Figura A.4, l'últim botó de la segona fila de botons de la part superior), que et permet carregar les simulacions ja computades que s'han guardat en l'ordinador (com les que es mostren en la pàgina de *Demo*) i poder veure els paràmetres amb què s'ha calculat. El botó que té al costat esquerre (com es pot veure en la Figura A.4) és el botó de *Save*, que et permet guardar la simulació que s'ha computat per poder-la reproduir posteriorment sense esperar el temps de càlcul de l'ordinador. També és interessant que aquestes simulacions que es guardin, es podran veure directament en la pàgina de *Demo*.

Per últim, un cop s'està reproduint una simulació, es pot clicar el botó *Reverse* (el botó que es pot veure a la dreta del multiplicador del temps de reproducció, en la figura A.4), aquest botó atura la reproducció de la simulació i gira les velocitats (180°) de les partícules al moment en què s'ha clicat, fent que si després es clicca el botó de càlcul, et calcula l'evolució del sistema des d'aquell estat. La idea principal és que les partícules haurien de tornar al seu estat inicial, ordenades com a l'inici. Això sol ser així quan no hi ha massa partícules (massa interaccions) i el temps de la simulació no és molt elevat, si no és així, és probable que a causa del càlcul finitament precís de l'ordinador sorgeixin evolucions que no retornen a l'inici com era esperat.