

Pràctiques en empresa: Packet vs Cat

Autor: Adrià Bravo Vidal

Tutors: Bruno Julià Díaz, Munts Guilleumas Morell, Carles Calero

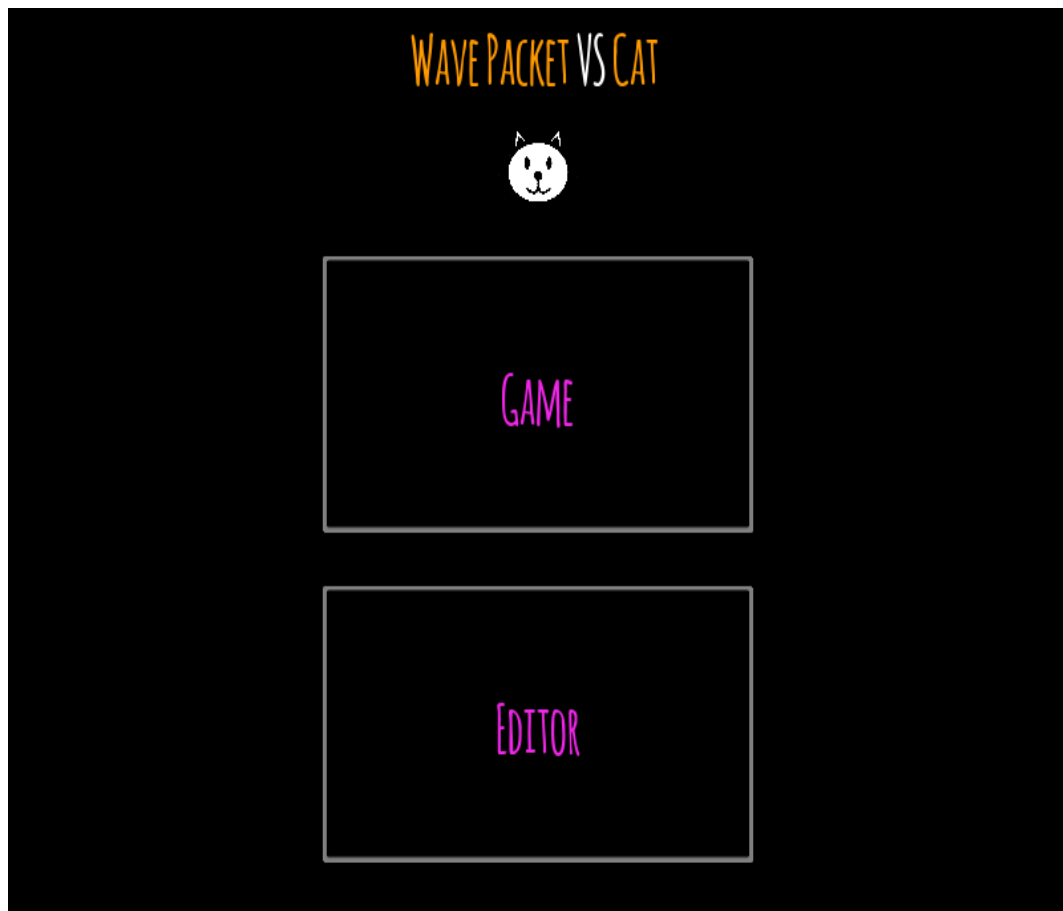


Figure 1: Portada del programa **Packet vs Cat**

Contents

1	Introducció	2
1.1	QuantumLab-UB	3
2	Metodologia: Funcionament de les pràctiques	3
2.1	L'ús del GitHub	4
2.2	Documentació	5
3	Resultat final: Packet vs Cat	5
3.1	Plantejament inicial	5
3.2	Resultat final: Packet vs Cat	6
3.2.1	Aspectes Generals	6
3.2.2	Editor	7
3.2.3	Videojoc	9
4	Desenvolupament	12
4.1	Plantejament físic	12
4.1.1	Condicions de contorn	14
4.1.2	Unitats	14
4.2	Problema computacional	14
4.2.1	Mètode Crank-Nicolson	15
4.2.2	Elecció d'un discretitzat i paràmetres del paquet	17
4.2.3	Discretitzat i moment inicial del paquet	17
4.2.4	Mida de la caixa i σ del paquet	19
4.2.5	Convergència del mètode	19
4.3	Problema gràfic/interactiu	21
4.3.1	Kivy	22
4.3.2	Connectant Kivy amb Matplotlib i Python	22
4.4	Realització del programa	23
4.4.1	Acceleració del mètode Crank-Nicolson	23
4.5	Realització del editor	24
4.5.1	Intuïtització del programa	25
4.6	Realització del videjoc	25
4.6.1	Elecció d'un personatge i el seu objectiu	25
5	Aspecte divulgatiu	26
6	Valoració de les pràctiques	26

1 Introducció

L'informe següent preten ser la descripció del treball realitzat durant el semestre de primavera del curs 2019-20 associat amb el Departament de Física Quàntica i Astrofísica de la Universitat de Barcelona a través de l'assignatura Pràctiques

d'Empresa. El projecte ha estat supervisat i guiat per la Dra. Montserrat Guilleumas, el Dr. Bruno Juliá Díaz i el Dr. Carles Calero.

Cap a finals del primer semestre de tercer el professor de l'assignatura de Física Computacional, el Dr. Bruno Juliá Díaz, ens va oferir la possibilitat de realitzar unes pràctiques d'empresa relacionades amb el desenvolupament de programes interactius (videojocs o simulacions) que tenien com objectiu la divulgació de la física quàntica i mecànica estadística. Per un altre banda, el rigor físic implícit a l'hora de desenvolupar els programes els convertia en útils per complementar explicacions d'assignatures de introducció en aquestes matèries.

La proposta em va interessar des del primer moment. Primerament, perquè m'encanta programar i d'aquesta manera podia seguir fent-ho, tot millorant les meves habilitats. Segon, perquè aquest projecte deixava un gran espai pel desenvolupament propi, un espai propi on deixar que la física que havia après durant el grau i la meua pròpia creativitat es barrejessin. I, per últim, perquè sempre he tingut un gran interès per divulgar el coneixement i aquest treball em semblava una gran manera de fer-ho.

1.1 QuantumLab-UB

El projecte en el que s'emmarquen aquestes pràctiques d'empresa és el QuantumLab-UB [1]. El conjunt ha estat realitzat per estudiants de física de la UB i els seus objectius principals són:

- Desenvolupar aplicacions per popularitzar aspectes de la mecànica quàntica a un públic ampli.
- Resoldre problemes quàntics realistes. Per tant els programes han de incloure resolució numèrica de l'equació de Schrödinger en diferents passos.
- Compartir el codi realitzat a través de la plataforma GitHub.

Cada un dels estudiants aplicants per aquestes pràctiques ha de desenvolupar individualment el seu propi programa o continuar amb un ja realitzat anteriorment. En cap cas dues o més persones treballen simultàniament en un únic projecte. Dos o tres estudiants realitzen cada trimestre aquestes pràctiques, establint reunions setmanals conjuntes amb els tutors per tal de desenvolupar el projecte amb feedback i ajuda constants.

En el meu cas, vaig desenvolupar aquestes pràctiques simultàniament amb els estudiants Jofre Vallés i Júlia Cabrera.

2 Metodologia: Funcionament de les pràctiques

Les pràctiques han consistit en treballar setmanalment i de manera autònoma de l'ordre de vint hores, tot avançant en el projecte proposat resolent el problema físic o el computacional, preparant la part gràfica interactiva ... És important

destacar el format setmanal, ja que aquesta ha estat la freqüència amb la que els tres estudiants i els tres doctors ens hem reunit.

Les reunions han tingut diferents papers al llarg del transcurs de les pràctiques:

Primerament, han estat una mena de guia dels tutors cap als alumnes, tot ajudant-nos a donar els primers passos: escollir el tema sobre el que farem l'aplicació, discutir la viabilitat de les nostres idees inicials, etc.

Un cop cada alumne ha tingut clar com procedir i que fer més o menys, **les reunions han consistit en la presentació ordenada del treball realitzat per cada alumne durant la setmana**. Aquest procés ha estat l'eix al voltant del que han girat les pràctiques. Aquesta presentació s'havia de preparar, ja que abans de la reunió cada alumne havia d'enviar un document en el que explicava què havia fet durant la setmana.

Un cop la presentació s'havia realitzat es discutien els resultats obtinguts i es debatia què estava bé, què no, i quin era el pròxim pas a seguir. Ha de quedar clar que **aquest procés no era unidireccional**: tant els tutors com els alumnes podien defensar les seves idees i punts de vista, i era l'alumne qui normalment tenia el dret a decidir en quin aspecte del treball voldria centrar-se la setmana següent. Per un altre banda, s'intentava que aquestes reunions fossin dinàmiques i participatives per part de tots els implicats, així els alumnes podien també fer suggeriments i opinar sobre el treball dels altres alumnes.

El treball realitzat ha estat doncs autònom, però supervisat, donant espai per poder crear el que tu volies crear però adaptant-te a les exigències dels professors i acceptant i valorant idees i suggerències. A més a més, després de cada reunió cada alumne havia de compartir el seus avanços en el codi per tal que els altres poguessin testejar el seu analitzar-lo o testejar-lo. Per fer-ho utilitzàvem una plataforma anomenada GitHub. Per últim, cal destacar que durant totes les pràctiques s'ha intentat mantenir una bona documentació sobre tota la feina realitzada.

2.1 L'ús del GitHub

Un cop finalitzada la reunió setmanal, s'havien de penjar els avenços en el codi en la plataforma GitHub. GitHub és un servei de hosting de repositoris Git i funciona amb un sistema de branques i subbranques. El projecte QuantumLab-UB disposa de la seva pròpia branca (anomenada repositori) [1] en la que es troben en forma de subbranques tots els projectes corresponents als diferents alumnes que han realitzat les pràctiques. **Un cop l'avenç setmanal en el codi s'havia realitzat, s'acoplava al repositori principal**, i passava a ser públic. Realitzar aquesta

acció i utilitzar aquesta plataforma ha estat molt profitós ja que:



- Disposa el codi a l'abast de tots els implicats en el projecte: qualsevol que vulgui testejar-lo o consultar-lo ho pot fer en pocs minuts.
- S'apren a treballar amb Github: el host de codi font més gran del món.

2.2 Documentació

Aquestes pràctiques són sinònim d'hores i hores de programació i plantjemanet de simulacions, comprovacions de convergència del mètode numèric, obtenció de resultats bons i dolents. En aquest context, resulta molt convenient prendre notes sobre tot el que es fa. És un aspecte molt destacat pels professors que no deixen de recomenar-ho durant totes les pràctiques. **Cal anotar no només tots els plantejaments i tots els resultats obtinguts, sinó també el que s'apren i d'on s'ha après.** Tot aquesta especie de diari és el que s'ha anomenat com a documentació i ha estat un altre aspecte clau de les pràctiques. A banda de fer les hores d'eina més efectives, realitzar una bona documentació sobre el treball realitzat ha estat útil per aprendre a escriure en Latex.

L^AT_EX

3 Resultat final: Packet vs Cat

En la següent secció, es vol plantejar la idea inicial des de la que es va partir i explicar quin ha estat el resultat final obtingut, és a dir, explicar en que consisteix l'aplicació/videojoc de divulgació de física quàntica desenvolupat.

3.1 Plantejament inicial

La primera idea fou **desenvolupar un videojoc que mostrés en pantalla un paquet gaussià de dos dimensions atrapat en una caixa amb 'parets' de potencial infinit.** Aquest paquet gaussià podia tenir unes característiques inicials com un determinat moment i dispersió espacial, i es podria solucionar la seva evolució temporal resolent l'equació de **Schrodinger computacionalment.** Per pantalla es podria observar la densitat de probabilitat en funció de l'espai i veure com aquesta evoluciona amb el temps. D'aquesta manera l'usuari podria observar el comportament espacial d'un paquet gaussià a partir del qual **es poden inferir moltes propietats físiques força interessants.**

No hem d'oblidar que aquest paquet podria ser una bona representació d'alguna partícula subatòmica tal com un electró. L'usuari podria observar com en el món de la física quàntica (el món real) no sabem on està exactament aquest electró, però si que podem saber on podria trobar-se.

Inicialment, per tal de mantenir entretingut a l'usuari menys acostumat a la física quàntica, es va plantejar que **es pogués controlar directament el moviment d'un personatge fictici que estigués també atrapat a la**

caixa i que disposes d'una certa 'vida'. El paquet gaussià podria fer-li mal i restar-li vida cada cop que fos tocat per aquest, per tant l'objectiu de l'usuari seria escapar del paquet gaussià.

3.2 Resultat final: Packet vs Cat

La idea inicial va anar patint moltes variacions al llarg del transcurs de les pràctiques, tot matissant-se i deformant-se. El mètode numèric, la seva convergència i temps de càlcul van suposar la major limitació a algunes idees inicials, i per un altre banda, durant el procés en van sorgir de noves. Es van implementar moltes i el conglomerat de tot plegat ha acabat sent l'aplicació final, anomenada **Packet vs Cat**. A la figura 1, podem observar la portada del programa.

L'aspecte físic principal tractat en aquest programa és el mateix que el del plantejament inicial: observar l'evolució d'un paquet gaussià en dues dimensions atrapat en una caixa amb parets de potencial infinit. Durant el procés de desenvolupament del videojoc plantejat a la idea principal es va veure que **es disposava de prou eines per realitzar un simulador interactiu prou interessant** i es va decidir afegir-lo a la versió final del programa. Per tant, el programa presenta, a banda del joc plantejat inicialment, un petit simulador anomenat editor.

3.2.1 Aspectes Generals

Els dos modes del programa ens deixen inicialment davant d'un paquet gaussià dins una caixa en dues dimensions estàtic, pintat de manera que els colors més brillants corresponen a densitats de probabilitat més altes i els més foscos, a densitats de probabilitat més baixes. Al requadre superior de la dreta trobem, en unitats reduïdes, informació diversa sobre el paquet, com la norma, la seva energia, el temps de computació o el moment inicial. Més abaix, trobem els comandaments, que variaran segons ens trobem en un mode o en l'altre. Veure figures

3.2.2 Editor

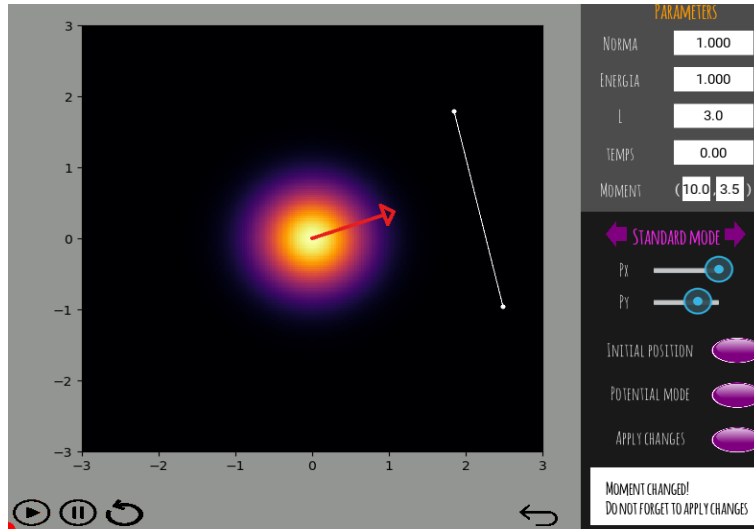


Figure 2: Mode editor configuració estandar de potencial. A més a més, s'ha utilitzat el mode 'potential mode' per tal d'obtenir una línia blanca al mig de la caixa que representa un potencial infinit.

L'editor és un mode del programa que **ens deixa variar alguns paràmetres inicials del paquet gaussià i del potencial interior de la caixa**. Ve donat en forma de dos submodes que deixen una configuració predeterminada de potencial: el submode standar i el submode slit. El primer no es més que una caixa buida amb el paquet inicialment centrat al mig, i el segon és una representació a petita escala d'un experiment d'una escaleta. Veure figura 2, 3. En cada un d'ells podem variar aspectes físics del paquet i del potencial. Es pot:

- Variar el **moment inicial** del paquet (abans que comenci l'evolució)
- Variar la **posició inicial** del paquet (abans que comenci l'evolució, i només en el submode standar)
- Seleccionar el **botó 'potential mode'**, que dona la possibilitat de dibuixar en qualsevol moment de l'evolució una línia blanca en pantalla. Aquesta línia representa un potencial infinit allà on sigui marcada. A la figura 4 podem veure un exemple de com actua aquest mode sobre el paquet.
- Variar la **mida del slit** (abans que comenci l'evolució, al submode 'slit')

Totes aquestes opcions deixen l'editor com un apartat del programa amb el que **es poden fer simulacions amb una bona quantitat de possibilitats**.

Divulгатivamente, pot ser utilitzat per explicar alguns aspectes generals de la mecànica quàntica i també per explicar-ne alguns més concrets sobre evolucions de paquets gaussians.

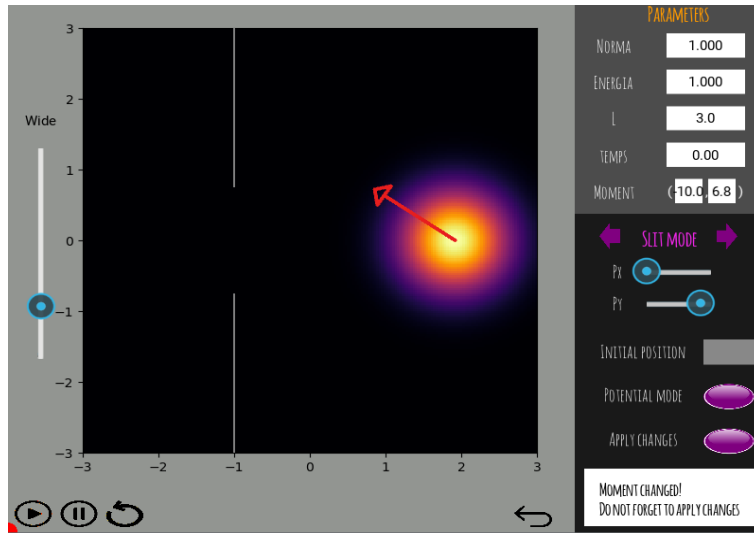


Figure 3: Mode editor configuració esletxa de potencial

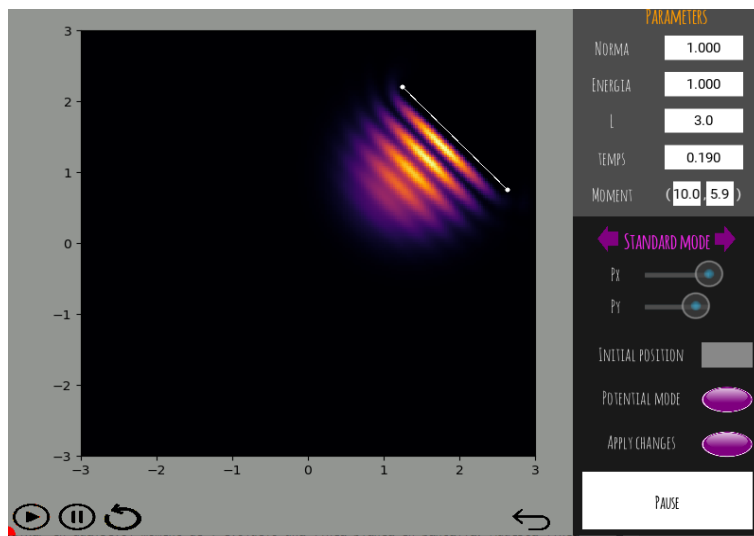


Figure 4: 'Potential mode' actuant.

3.2.3 Videojoc

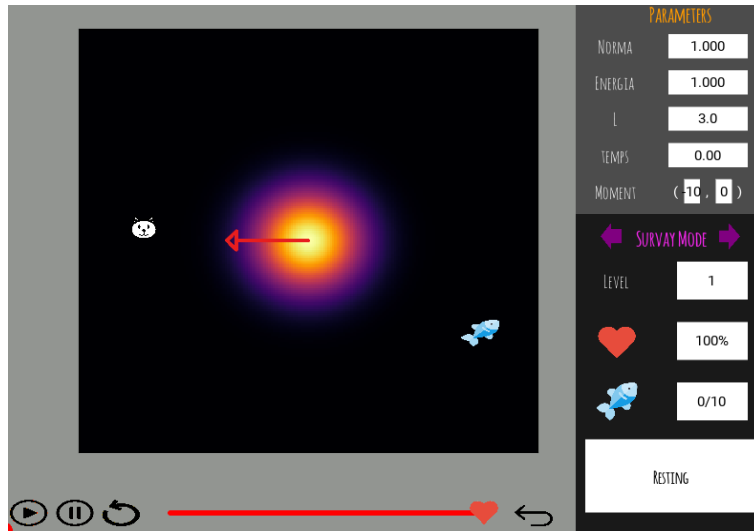


Figure 5: Mode 'game' en la pantalla inicial del mode survay.

El videojoc ha estat sempre l'objectiu principal del programa i segueix el plantejament inicial proposat anteriorment.

L'usuari **controla des del seu teclat o comandament un gat**, que podria ser el gat de Schrödinger. Pot moure'l horitzontalment i verticalment i només es pot moure dins una caixa en dues dimensions on també està confinat un paquet gaussià amb un cert moment inicial, com podem veure a la figura 5.

Aquest gat disposa d'una certa vida (barra inferior vermella, figura 5). Quan s'inicia el joc comença l'evolució del paquet i comença a moure's per la caixa. **L'objectiu de l'usuari és moure el gat de manera que no sigui tocat pel paquet gaussià.**

Recordem que l'aspecte del paquet gaussià que observem per pantalla és la densitat de probabilitat i aquesta es troba repartida per tota la caixa, com veurem més endavant. Per tant quan es parla de que el gat 'toca' el paquet és vol dir que el gat es troba en una zona d'alta densitat de probabilitat del paquet gaussià. A partir d'un cert umbral de densitat de probabilitat fixat, el gat nota el paquet i es fa mal, i quan més alta sigui aquesta densitat de probabilitat, més mal rebra. Quan reb mal, la seva barra de vida disminueix i si aquesta arriba ser zero, l'usuari perd la partida.

Per tant, l'objectiu del gat és evitar les zones d'alta densitat de probabilitat en pantalla.

El videojoc disposa de dos modes diferents:

Survay mode En aquest cas partim de la configuració inicial presentada en la figura 5. El paquet té un moment inicial gran horitzontal i quan comença el joc, s'activa la seva evolució i es comença a desplaçar. **El gat s'ha de moure tot evitant ser tocat pel paquet, i amb l'objectiu fixat de recol·lectar deu peixos** que van apareixent seqüencialment (veure figura 6). Cada cop que es menja un, recupera una mica de vida. Un cop se n'ha menjat deu (se'ls menja simplement situant-se a sobre seu) s'acaba el mode de joc i l'usuari guanya. **Si el gat perd tota la vida abans que això passi, l'usuari perd.**

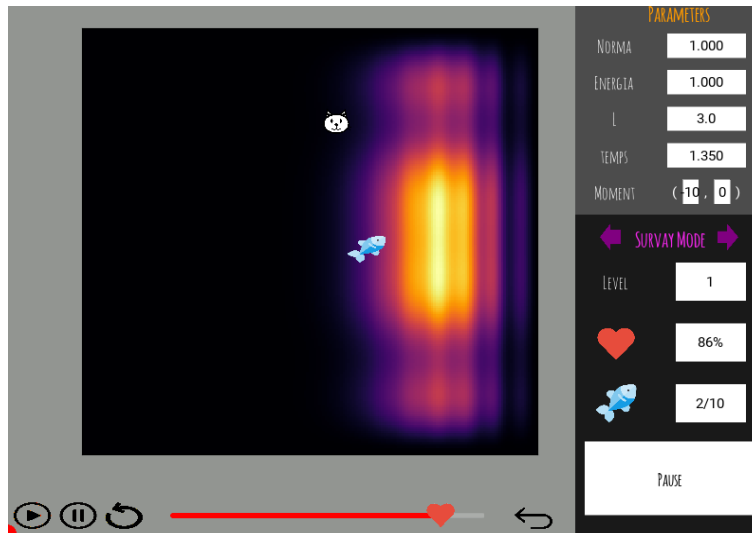
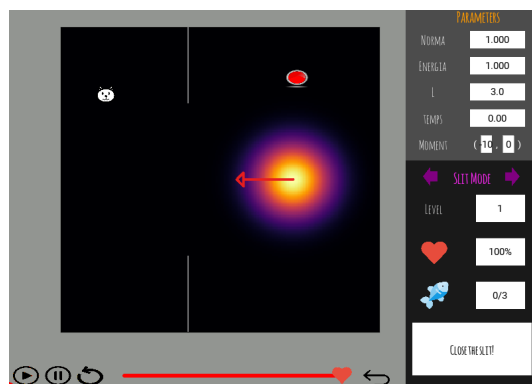
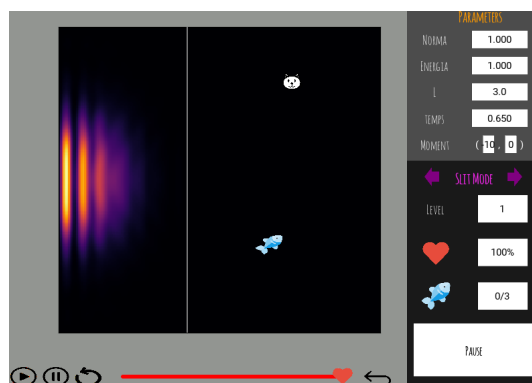


Figure 6: Submode 'survay'. El gat està escapant del paquet tot intentant recol·lectar alguns peixos per menjar. N'ha recol·lectat alguns com es veu al requadre dret inferior, però ha perdut una mica de vida per ser tocat pel paquet.

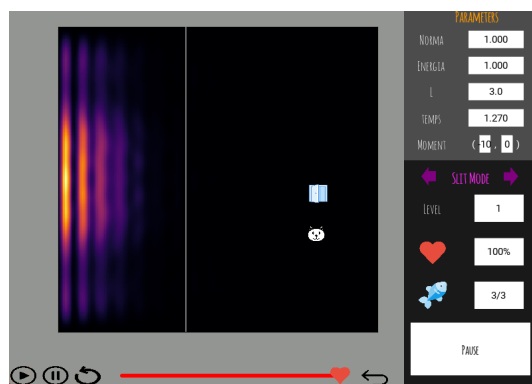
Slit mode : Aquest mode es mostrarà gràficament i s'explicarà seqüencialment com funciona ja que és força complexe. En aquest cas partim de la configuració 'slit' esmentada a la secció del editor. Inicialment, el gat es troba a una banda del slit i el paquet, a l'altre. Quan comença el joc, el paquet es dirigeix al slit i l'atravessa.



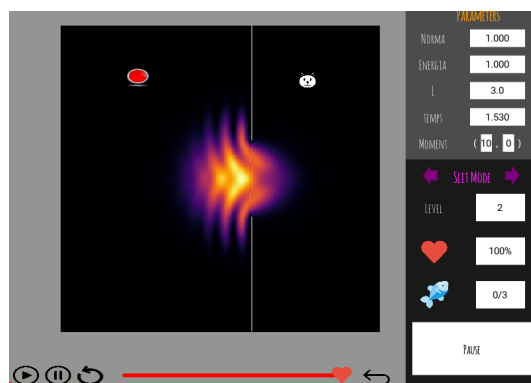
El gat, que no pot passar per sobre del potencial infinit del slit, ha de passar pel mateix forat que el paquet amb l'objectiu de pitjar un butó situat a l'altre banda. Quan ho fa, es tanca la slit i tota la densita de probabilitat que es troba en cada banda de la pantalla queda aïllada:



El gat llavors ha de recol·lectar tres peixos (que es troben a la seva banda) i un cop ho ha fet, li apareix una porta, que l'envia al pròxim nivell.



El pròxim nivell té el mateix plantejament del slit però aquest cop invertit espacialment i amb un forat més petit. El gat ha de fer el mateix que abans però ara la dificultat és major, ja que menys densitat de probabilitat pot passar a la banda on es troba un botó:



Ara es torna a repetir el mateix procediment fins arribar a la pròxima porta, que et porta a un nivell amb el slit més petit. Així es procedeix progressivament fins satisfer quatre nivells. Si l'usuari ho fa sense perdre tota la vida del gat, guanya la partida!

Nota: Tot el codi utilitzat per fer funcionar aquest programa així com els programes que s'han de instal·lar per utilitzar-lo es poden trobar a [1] i a [2].

4 Desenvolupament

Aquest apartat preten explicar els fonaments del programa i **com s'ha procedit per construir-lo**. Quin problema físic es va plantejar i com es va resoldre, com es va desenvolupar el mètode numèric i l'estudi de la seva convergència, i finalment, com es va construir la part gràfica i el propi videojoc.

Cal remarcar des d'una perspectiva personal que abans de les pràctiques no tenia pràcticament coneixement sobre Python, Numpy, Matplotlib o Kivy. Aquestes últimes són llibreries de Python utilitzades per realitzar la part computacional/estructural i la part gràfica del joc. Així, vaig haver d'aprendre segons vaig necessitar a programar en Python i a fer servir les llibreries descrites.

4.1 Plantejament físic

El problema físic que es vol resoldre en aquest problema és **la evolució d'un paquet gaussià amb un moment ben definit atrapat en una caixa 2D amb potencial infinit als cantons**. Un paquet gaussià satura les relacions d'incertesa i per aquesta raó, és una bona funció d'ona amb la que treballar. El

paquet gaussià, centrat a $(x, y) = (0, 0)$ amb el que s'ha treballat és el següent (següint [3]):

$$\psi(x, y, p_x, p_y, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{4\sigma^2}(x^2+y^2) + \frac{i}{\hbar}(p_x x + p_y y)} \quad (1)$$

Com podem observar, aquest paquet està caracteritzat per un moment i dispersió espacials inicials. El paquet gaussià està normalitzat a la unitat. Això vol dir que la densitat de probabilitat espacial de trobar una partícula que pugui estar representant és:

$$\rho(x, y) = |\psi(x, y)|^2 \quad (2)$$

I s'ha de satisfer:

$$1 = \iint_{R^2} \rho(x, y) dx dy \quad (3)$$

Aquest paquet gaussià **podria representar, per exemple, un electró atrapat en una caixa del tipus semblant al potencial descrit**. En aquestes escales subatòmiques, és impossible saber on es troba exactament l'electró i per això utilitzem la densitat de probabilitat de trobar-lo en un punt espacial per descriure la situació físicament.

Per un altre banda, **el potencial és infinit i es troba distribuït en el perímetre d'un quadrat de costat $2L$** , tot envoltant el paquet gaussià. Analíticament el podem descriure així:

$$V(x, y) = \begin{cases} \infty & \text{if } |x| > L \text{ or } |y| > L \\ V_i(x, y) & \text{if } |x| < L \text{ and } |y| < L \end{cases} \quad (4)$$

On $V_i(x, y)$ fa referència al potencial interior de la caixa. En el nostre cas i com a configuració inicial tindrem, normalment, $V_i(x, y) = 0$. Podem observar que tal i com s'ha definit el paquet a (1) és no nul a tot l'espai en dues dimensions, però quan imposem que estigui en l'interior d'una caixa com la definida en el potencial (4) passa a ser nul fora d'aquesta.

Per trobar l'evolució del paquet en el potencial plantejat **s'ha de resoldre la equació de Schrödinger en dues dimensions dependent del temps:**

$$i\hbar \frac{\partial \psi}{\partial t} = H\psi \quad (5)$$

On ψ fa referència al paquet gaussià plantejat i H fa referència al següent Hamiltonià expressat en l'espai de posicions:

$$H = -\frac{\hbar^2}{2m} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) + V(x, y) \quad (6)$$

L'utilització d'un **paquet gaussià** com aquest presenta alguns beneficis remarcables ja que **es coneixen algunes dades analítiques sobre ell**. A més, si ens centrem en la part espacial del paquet, ens adonem que correspon a un autoestat d'un potencial del tipus harmònic.

4.1.1 Condicions de contorn

Per resoldre el problema computacional, caldra imposar unes condicions de contorn al paquet en lloc de imposar explícitament un potencial infinit als cantons de la caixa (computacionalment és molt més difícil aquesta última opció). Aquestes condicions de contorn són intuïtivament fàcils. Ja que el paquet no pot atravesar un potencial infinit, s'haurà de satisfer:

$$\rho(x, y) = 0 \quad \text{if} \quad x, y = L \quad (7)$$

4.1.2 Unitats

Per treballar de manera còmoda en aquest treball s'han utilitzat unitats reduïdes, que corresponen a fer:

$$\hbar = 1, \quad m = 1 \quad (8)$$

En aquest context, tot passa a expressar-se en unitats reduïdes: des de la unitat de temps fins l'unitat d'espai. Presenta diversos beneficis treballar en aquestes unitats, ja que són les naturals del sistema en el que treballem i es poden expressar quantitats molt engorronides (com 37 Amstrongs) amb números com $L=3$ o $L=4$, que ja sabem que estem expressant en unitats reduïdes.

4.2 Problema computacional

Un cop plantejat el problema físic, s'ha de resoldre computacionalment l'evolució del paquet gaussià. Per resoldre computacionalment aquest problema primerament **s'ha de discretitzar l'espai, dividint la caixa de $2L \times 2L$ en una matriu de N^2 elements** on cada un representa una porció de la caixa d'àrea $dx dy$. L'indexat de la matriu dona idea de la ubicació de la porció d'àrea dins la caixa. Al discretitzar de l'espai se l'anomena malla.

Si volem dividir la nostra caixa $L \times L$ en una matriu de N^2 elements haurem de partir l'espai en porcions dx, dy tals que:

$$dx = dy = \frac{2L}{N} \quad (9)$$

Un cop discretitzat l'espai, **assignem a cada porció dx^2 de la malla un valor de la funció d'ona donat per (1)**. Ara podem representar aquesta funció d'ona com una matriu i en lloc de parlar del seu valor en x, y ens podem referir al seu valor en l'element de matriu i, j :

$$\psi(x_i, y_j) = \psi_{i,j} \quad x_i = -L + dx * i \quad y_j = -L + dy * j \quad (10)$$

Un cop tenim la funció d'ona discretitzada dins la caixa s'ha de procedir a resoldre la seva evolució temporal mitjançant l'equació de Schrödinger (5).

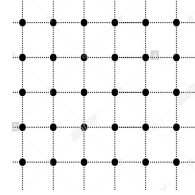


Figure 7: Representació d'una malla en dues dimensions

Com podem veure a (6) aquesta equació diferencial involucra tant derivades espacials com temporals i és força complexa. **Per resoldre-la computacionalment s'hauran d'utilitzar mètodes de diferències finites, en els que s'aproxima la derivada amb difències finites.**

Ja que estarem tractant amb una evolució temporal també el temps ha de ser discretitzat. Per referir-nos a cada petit interval de temps en el que es resol l'equació utilitzarem dt . Si volem resoldre l'evolució d'un temps A a un temps B tindrem N_t passos temporals:

$$N_t = \frac{t_B - t_A}{dt} \quad (11)$$

Per un altre banda, per referir-nos al temps t en el que es troba la funció de una utilitzarem la següent notació:

$$\psi_{i,j}(t_k) = \psi_{i,j}^k \quad t_k = t_A + dt * k \quad (12)$$

4.2.1 Mètode Crank-Nicolson

El mètode en diferències finites escollit per solucionar el problema computacional va ser el mètode Crank Nicolson (CK). Aquest mètode és utilitzat sovint per resoldre numèricament equacions amb derivades parcials. Es tracta d'un **mètode de segon ordre en el temps, implícit i numèricament estable**, sempre que s'agafi un discretitzat temporal i espacial adequat. Donada la seva efectivitat, no es va dubtar molt a l'hora d'escollir-lo.

Per explicar-lo i entendre-lo, farem una breu explicació de com s'obté l'algorisme en una dimensió i després ho generalitzarem a dues. Ens seran útils les següents expressions que discretitzen la primera i la segona derivada:

$$f'(x_k) = \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k} \quad (13)$$

$$f''(x_k) = \frac{f(x_{k+1}) - f(x_k) + f(x_{k-1}))}{x_{k+1} - x_{k-1}} \quad (14)$$

Crank-Nicolson en una dimensió Abans de començar amb el CK en dues dimensions es va dominar el CK en una dimensió, ja que el primer no deixa de ser una generalització del segon. Discretitzem (5) utilitzant (13) des de t fins a $t + \frac{\Delta t}{2}$. Per un altre banda, fem el mateix però retrocedint des de $t + \Delta t$ fins a $t + \frac{\Delta t}{2}$. Si ara aïllem $\psi(x, t + \frac{\Delta t}{2})$ de cada expressió i igualem arribem (passem a la notació introduïda anteriorment):

$$(1 + i \frac{\Delta t}{2}) H \psi_i^{k+1} = (1 - i \frac{\Delta t}{2}) H \psi_i^k \quad (15)$$

Continuem aplicant l'hamiltonia (6) en una dimensió a l'expressió anterior. S'observa que aquest H presenta una derivada espacial de segon ordre. Si desenvolupem (15) juntament amb (14) obtenim:

$$\begin{aligned}
& \psi_i^{k+1} \left[1 + \frac{i\Delta t}{2\hbar} \left(V_i + \frac{\hbar^2}{m\Delta x^2} \right) \right] - \frac{\hbar\Delta t}{4m\Delta x^2} (\psi_{i+1}^{k+1} + \psi_{i-1}^{k+1}) \\
& = \psi_i^k \left[1 - \frac{i\Delta t}{2\hbar} \left(V_i + \frac{\hbar^2}{m\Delta x^2} \right) \right] + \frac{\hbar\Delta t}{4m\Delta x^2} (\psi_{i+1}^k + \psi_{i-1}^k)
\end{aligned} \tag{16}$$

Recordem que estem treballant en unitats reduïdes i per tant, en aquesta expressió caldria fer $\hbar = 1, m = 1$. Conegut un vector ψ^k de N dimensions (recordem (9)), ens adonem que el problema es tracta en resoldre un sistema de N incognites (el vector ψ^{k+1}) i N equacions donades per (16). Per tant, hem de resoldre:

$$A\psi^{k+1} = B\psi^k \tag{17}$$

Aquest problema que en principi pot semblar molt costós computacionalment és **simplicifia enormement gràcies a que la matriu A és tridiagonal i existeix un algorisme que resol ràpidament aquest sistema en només N operacions.**

Es van realitzar diverses proves de CK en una dimensió utilitzant aquest algorisme i partint de un ψ^k conegut, com per exemple, un paquet gaussià en una dimensió amb un cert moment inicial. El **mètode CK conserva casi exactament la norma** i per tant és fàcil saber si està funcionant bé: mitjançant la conservació de la norma. Un cop conegut el CK en una dimensió, es va passar al problema en dues.

Crank-Nicolson en dues dimensions La generalització directe del mètode Crank-Nicolson en dues dimensions és certament força costosa computacionalment. Per aquesta raó en aquest cas es complementa el CK amb un altre mètode anomenat **'alternate direction implicit method' (ADI)** [4]. En aquest cas, tenim que ψ té dues dimensions i l'Hamiltoniana pren derivades tant en x i com en y . Tenint això en ment, un pas de ADI que vagui de t_k a t_{k+1} es pot dividir en dos subpassos:

- El primer pas pren la **derivada de X implícitament** en el temps $k + \frac{1}{2}$ i la **de Y explícitament** en el temps k . A partir d'aquí es va des de ψ^k a $\psi^{k+\frac{1}{2}}$.
- El segon pas pren la **derivada de Y implícitament** en el temps $k+1$ i la **derivada de X explícitament** en el temps $k + \frac{1}{2}$. D'aquesta manera s'arriba des de el $\psi^{k+\frac{1}{2}}$ obtingut abans al ψ^{k+1} .

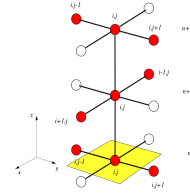


Figure 8: Representació gràfica del ADI

Desenvolupant com s'ha fet per obtenir (16), però tenint en compte que es pren una de les derivades espacials implícita i l'altre explícita, s'arriba a la següent expressió matricial, semblant a l'obtinguda anterior. Tenim que:

$$\begin{aligned} \psi_{i,j}^{k+\frac{1}{2}} [1 + i(\frac{\Delta t}{4\hbar} V_{i,j} + 2r)] - ir(\psi_{i+1,j}^{k+\frac{1}{2}} + \psi_{i-1,j}^{k+\frac{1}{2}}) = \\ \psi_{i,j}^k [1 - i(\frac{\Delta t}{4\hbar} V_{i,j} + 2r)] + ir(\psi_{i,j+1}^k + \psi_{i,j-1}^k) \end{aligned} \quad (18)$$

On $r = \frac{\Delta t \hbar}{4\Delta x^2 m}$. Un altre cop, tenim una equació del tipus:

$$A\psi^{k+\frac{1}{2}} = B\psi^k \quad (19)$$

Per obtenir un pas complet de ADI, és a dir, per obtenir ψ^{k+1} un cop obtingut $\psi^{k+\frac{1}{2}}$ s'han de canviar les matrius A i B ja que varia quin eix considerem explícit i quin implícit. Específicament, a partir de (18) s'ha de fer $k = k + \frac{1}{2}$ i permutar i,j. **La matriu A torna a ser tridiagonal i la matriu B ψ^k coneguda si coneixem en l'instat k el vector d'ona.** Un altre cop tenim doncs que aquest sistema d'equacions es pot solucionar amb l'algorisme ràpid tridiagonal, però en comparació amb el cas unidimensional:

- Un pas de k a k+1 involucra dos subpassos.
- Cada subpas involucra solucionar un sistema de NxN incògnites i NxN equacions amb molts càlculs involucrats.

Per tant, tot i que es pot demostrar que aquest mètode es **molt estable per un gran rang de valors de r i conserva la norma de la funció de ona molt bé**, es pot veure que pel gran nombre de càlculs involucrats **no serà un mètode ràpid.**

Aquest ha estat, sense dubte, la batalla permanent d'aquestes pràctiques. **El programa havia de funcionar amb càlculs en directe** i això implicava, si es volia obtenir alguna cosa fluïda en pantalla, aconseguir que el mètode CK en dues dimensions fos ràpid. Més baix trobarem com es va solucionar parcialment la lentitud del mètode.

4.2.2 Elecció d'un discretitzat i paràmetres del paquet

Un cop desenvolupat el mètode hi havia un món de possibilitats per davant. Des del discretitzat espacial i temporal escollit fins a la σ del paquet, passant per quin rang de moments inicials podia tenir el paquet o la mida de la caixa. La búsqueda es va fer tenint en compte que es buscava un equilibri òptim entre temps de computació petit i mostrar una evolució física realista, a més a més de buscar un aspecte visual atractiu i interessant.

4.2.3 Discretitzat i moment inicial del paquet

Quan passem del continu real a un espai que està discretitzat finitament és perd informació, ja que representem els infinits punts del espai continu que hi ha dins la caixa amb només NxN. **Quan més gran sigui el discretitzat (quan més**

gran sigui dx, dy i dt) més informació és perd però per un altre banda, quan més petit és el discretitzat més gran és el temps de computació, ja que s'ha de resoldre un sistema amb més equacions i incògnites.

En el nostre cas, es va fer un estudi de la energia cinètica del paquet calculada computacionalment per a diferents moments inicials i diferents discretitzats espacials. La energia cinètica del paquet gaussià també es coneix analíticament i per tant, és fàcil realitzar una comparació, com es veu a la figura 9.

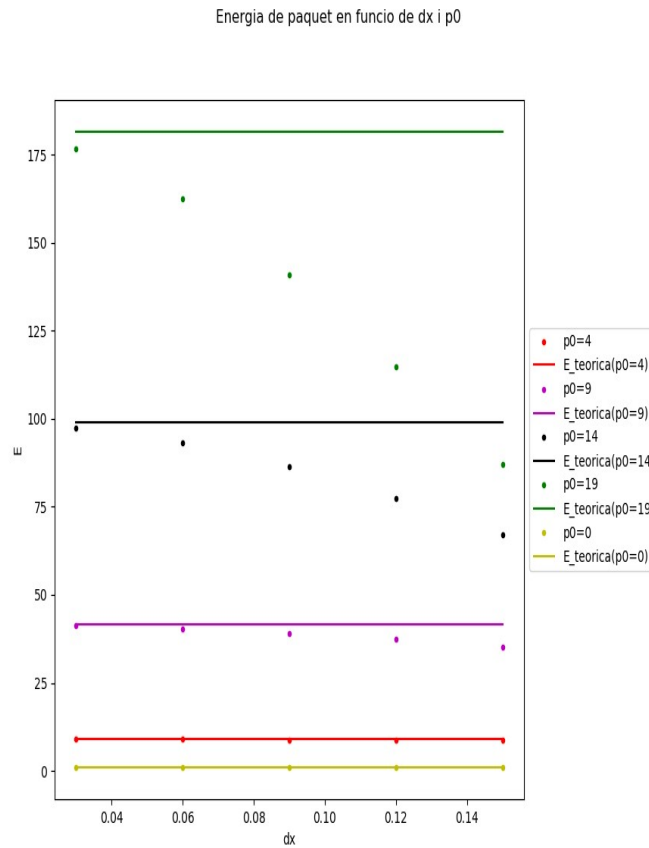


Figure 9: Energia teòrica i computacional d'un paquet gaussià amb un determinat moment inicial

Aquests resultats mostren que **quant més alt és el moment, més petit ha de ser el discretitzat espacial per descriure la seva evolució física de manera correcte**. Per tant, ja que es volia obtenir un rang de moments prou ampli en les simulacions i en el videojoc, es va acabar escollint el següent discretitzat i moments inicials:

$$\Delta x, \Delta y = 0.03, \quad \Delta t = 0.02, \quad 0 < p_i < 10 \quad i = x, y \quad (20)$$

4.2.4 Mida de la caixa i σ del paquet

Un cop escollit el discretitzat, s'havia d'escollir la mida de la caixa. Tenint en compte que estan directament relacionades per (9) s'havia d'escollir **una mida que deixes prou espai al paquet per evolucionar de manera vistosa gràficament però no tan gran com per proporcionar matrius massa grans i costoses computacionalment**. Finalment es va optar per $L = 3$. El criteri per escollir σ va tornar a ser el d'obtenir resultats vistosos quan el paquet evolucionés: es volia que es mantingues compacte el màxim temps possible, tot i que hagués de rebotar contra la parets. Per aquesta raó, és va escollir $\sigma = 0.25$.

Nota Python funciona conjuntament amb diverses llibreries especialitzades en un tema específic. En aquest cas, es va fer servir la llibreria Numpy pels aspectes matemàtics de matrius i operacions i Matplotlib per representar els resultats.

4.2.5 Convergència del mètode

Els detalls tècnics de com es va realitzar computacionalment el mètode es poden trobar al codi del joc, a [2]. Un cop el mètode va estar llest, s'havia de testejar si era correcte i la seva eficiència (temps de càlcul).

Per testejar la fiabilitat del mètode es va procedir de diverses maneres. La més evident és **la comprovació de la conservació de la norma i de la energia**. No obstant, s'havia d'anar més enllà.

Com s'ha dit ja varies vegades, el paquet gaussià presenta diverses propietats analítiques que resulten molt útils quan es vol testejar qualsevol mètode numèric relacionat amb la física quàntica. En el nostre cas, la convergència es va verificar amb **l'evolució del valor esperat de la posició (utilitzant teorema de Ethern), amb l'evolució de la dispersió en x (veure figura 4.2.5) i calculant l'energia cinètica i potencial del paquet sotmes a un potencial harmònic**

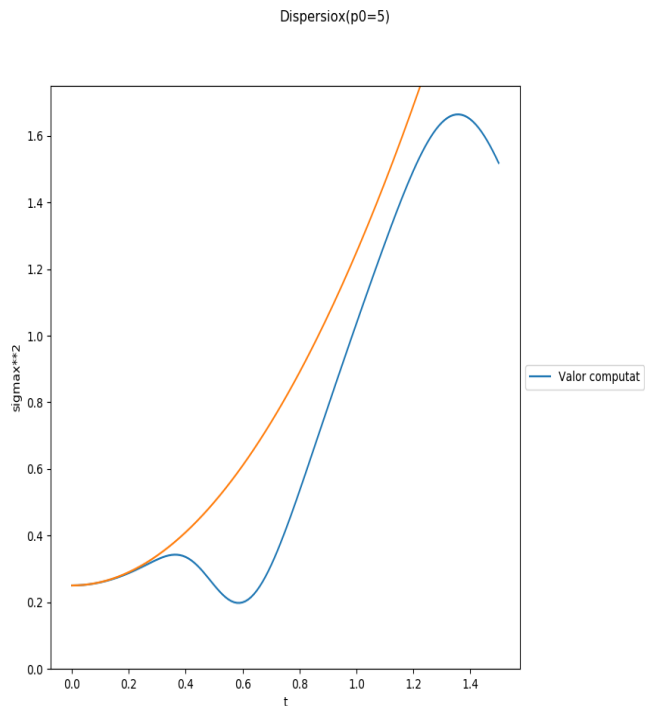


Figure 10: Evolució de la dispersió espacial x calculada computacionalment pel paquet en la caixa i analíticament per un paquet sotmés a un potencial nul . Es pot veure que inicialment coincideixen.

En la documentació,[2], es pot trobar de manera detallada com es va procedir en cada un dels casos mencionats, no obstant, a tall d'exemple, n'explicarem un. Cal remarcar que realitzar aquests tests va ser enormement útil per detectar errors en el codi i per conèixer a fons el CK.

Exemple: Teorema de Ethern Segons el Teorem d'Ethern, la evolució del valor esperat de les magnituds quàntiques d'un sistema haurien d'evolucionar segons evolucionaran les clàssiques. En el nostre cas, això es tradueix en que el valor esperat en x del paquet hauria de tenir el mateix valor que una partícula de la mateixa massa dins la mateixa caixa amb la mateixa velocitat. Per tant:

$$v = \frac{p}{m}, \quad \langle X \rangle = vt \tag{21}$$

Tenint en compte que la partícula es troba tancada en una caixa 2D i per tant rebota contra les parets, l'evolució obtinguda analíticament i computacionalment comparen així de bé en la figura 4.2.5.

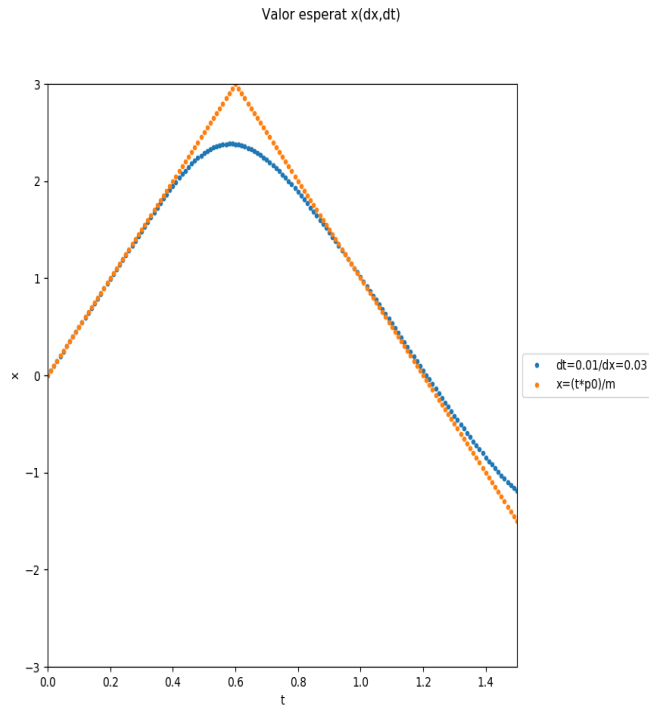


Figure 11: Evolució del valor esperat x calculat computacionalment pel paquet en la caixa i analíticament per una partícula clàssica. Es pot veure que coincideixen força bé.

4.3 Problema gràfic/interactiu

Un cop resolt el problema numèric i ja amb un bon control de la física quàntica i computacional implicada, tocava realitzar un programa interactiu, un programa que deixes a l'usuari:

- Observar en pantalla l'evolució de la densitat de probabilitat del paquet de manera clara i fluida.
- Variar condicions inicials físiques del sistema.
- Controlar per teclat el moviment d'un personatge fictici.

Per tant, **s'havia de combinar i relacionar tot el treball realitzat computacionalment amb Python i representat en Matplotlib amb alguna interfàç gràfica interactiva còmoda i vistosa per l'usuari.** Kivy va ser la llibreria de Python escollida per realitzar aquesta tasca.

4.3.1 Kivy

Kivy és una interfaç gràfica que **permet representar gràficament objectes i elements diversos, a banda de poder interactuar amb ells** [5]. Per fer-ho, Kivy connecta aquests objectes i elements a codi de Python mitjançant les anomenades classes: elements de programació que actuen com a funcions i contenidors de variables alhora. Aquesta llibreria s'aprofita doncs, de l'anomeada programació orientada a objectes.



Kivy funciona barrejant Python i arxius del tipus kivy on s'escriu amb el seu propi llenguatge. No s'especifica pas com es va programar en aquest llenguatge, sinó més bé què és va fer amb aquesta llibreria.

4.3.2 Connectant Kivy amb Matplotlib i Python

La idea inicial de representar la densitat de probabilitat d'un paquet gaussià en una caixa en 2D necessitava d'una bona representació vistosa e interessant. Durant tot el procediment anterior s'havia utilitzat per representar la densitat en funció del temps animacions generades per Matplotlib.

Donada la clarietat i la infinitat de possibilitats que oferia Matplotlib **es va decidir acoplar Matplotlib a Kivy** [7]. Kivy ofereix eines de visualització de funcions però es va creure que no eren tan bones com les de Matplotlib. Per tant, el que observem en la figura 2, per exemple, no deixa de ser un plot de Matplotlib acoplat a Kivy.

Aquesta llibreria de Python ofereix la funció `imshow` [6], que mostra una matriu numèrica de dos dimensions plasmada en un pla. Va ser aquesta la funció utilitzada per representar la densitat de probabilitat en pantalla.

Si aplem Matplotlib a Kivy per a que serveix doncs Kivy? **Kivy fa de mediador entre el usuari i el gruix del programa.** Per exemple, ofereix la possibilitat d'obrir una finestra en la que es mostra el plot de Matplotlib i altres elements interactius, com per exemple botons o slides. (figura 4.3.2). Aquests elements són control·lats pel propi usuari i permeten realitzar canvis numèrics en alguns aspectes de la simulació. Al pitjar algú butó podríem per exemple canviar la direcció i el mòdul del paquet, i això implica canviar la matriu ψ inicialment plantejada a Python. Per aquesta raó es diu que Kivy fa de mediador amb Python.

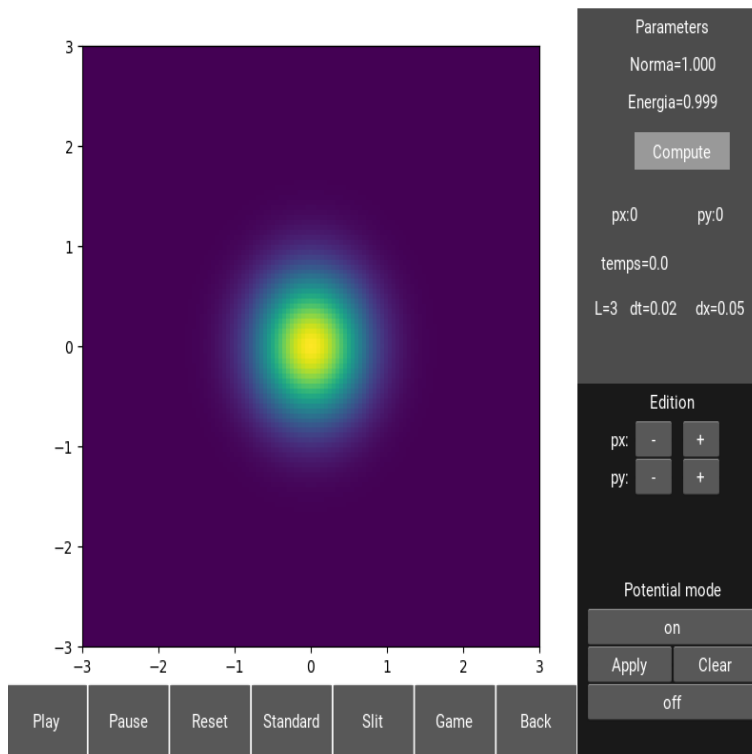


Figure 12: Un dels primers prototips del programa

4.4 Realització del programa

Les funcions del programa final han estat descrites al apartat Resultats Finals. No obstant, ara es vol fer èmfasis en com un cop dominades les eines necessàries per realitzar el programa (Kivy, Python i Matplotlib) es va procedir per realitzar el programa final.

Tot i que l'objectiu era crear un videojoc, per facilitar **es va posar el primer punt de mira en crear un apartat de simulacions**, per agafar solvència amb les llibreries. Un dels requisits necessaris per portar a terme tot el que es tenia en ment era mostrar en pantalla un càlcul en directe. És a dir, solucionar la evolució temporal del paquet a l'hora que es mostrava en pantalla.

Aquesta exigència portava certes complicacions i aviat es va veure que, **tal i com s'havia programat fins ara el mètode CK, l'animació en pantalla no era prou fluida.**

4.4.1 Accel·laració del mètode Crank-Nicolson

Un cop es va veure que el mètode CK donava un càlcul en directe massa lent per mostrar-lo en pantalla simultàniament de manera fluida, es va començar a

buscar una solució. Aquesta passa per tres punts diferents:

Depuració del codi La utilització de massa passes intermitjes en el codi del mètode numèric, utilització d'un tipus de indexació o un altre, fer servir més funcions de les que caldrien... Tot això causa la relentització del codi. Per tant, es va depurar el codi per millorar el temps de computació.

Utilització de Numba Numba és un compilador JIT de codi obert que tradueix codi de Python o Numpy en codi de màquina utilitzat [8]. Bàsicament, aquesta llibreria de Python actua directament sobre les funcions que tenen bucles i els accelera en bona mesura. **Es va aplicar aquesta llibreria amb grans resultats.**

Divisió d'un pas ADI Aquest punt de la solució no és una millora directa sobre el codi sinó **una millora sobre la seva representació**. Com s'ha explicat a l'apartat Crank-Nicolson en dues dimensions, un pas del mètode ADI implica dos subpassos.

La representació del càlcul en directe 'en pantalla es pot descriure així, i és manejat per la llibreria Kivy:

- En pantalla, s'observa la densitat de probabilitat del paquet en el moment k , ρ^k
- Es realitza una passa ADI a partir del ψ^k donat, obtenint ψ^{k+1} .
- Es calcula la densitat de probabilitat utilitzant 2
- Es dona ρ^{k+1} a Matplotlib, que el mostra en pantalla.
- Es torna a l'inici de l'algorisme.

Aquesta és la seqüència iterada una vegada i un altre per realitzar una animació. En aquest cas, el pas que enlenteix l'animació és la realització d'una passa ADI. Per tant, es va pensar que **una bona manera de fer-la més fluida seria realitzar la seqüència anterior per només un ADI, és a dir, partint de ρ^k anar a parar a $\rho^{k+\frac{1}{2}}$** . Aquest punt va millorar notablement la fluïdesa de l'animació.

4.5 Realització del editor

Un cop es va lograr un temps de computació decent i una animació fluida en pantalla de la evolució de la densitat de probabilitat del paquet, es va procedir a realitzar el mode editor, on es pot jugar amb un paquet gaussià tot variant les seves condicions inicials i on es pot variar la configuració inicial de potencial.

El que es pot fer amb aquest mode s'ha especificat ja a l'apartat Resultats finals. Tot hi així es ressaltaran alguns aspectes importants.

4.5.1 Intuïtització del programa

Durant la realització del mode editor es van començar a afegir moltes opcions variades i va arribar un punt en el que els tutors van resaltar una cosa certament obvia: Una persona externa que comencés a operar amb aquest programa no tindria ni idea de com funciona. No sabia quins botons podria pitjar i amb quin ordre.

Es va veure que moltes de les opcions no aportaven molt divulgativament i es va preferir centrar els esforços en depurar les opcions interactives més interessants com variar el moment, variar la posició inicial del paquet o dibuixar un potencial infinit en la caixa en 2D.

Tenir poques opcions però tan depurades van fer molt atractiu aquest mode. Es va afegir un petit requadre on es podien llegir els paràmetres més importants del paquet gaussià, com la seva energia o la seva norma. També es va afegir un petit requadre a l'esquerra inferior on es podia llegir l'ordre en el que s'havien d'aplicar els botons. Per exemple: Si varies el moment, al requadre apareix un missatge que diu: 'Recorda donar-li al butó d'aplicar els canvis!'. Tot això es pot observar en les figures 2, 3.

4.6 Realització del videojoc

La creació d'un videojoc amb tota l'artilleria reunida fins ara (ser capaços de variar paràmetres del paquet ràpidament i veure com evoluciona en pantalla, temps de computació ràpids, aspecte en pantalla intuïtiu...) va ser l'últim pas de la creació del programa Packet vs Cat. Per això és que el videojoc no deixa de ser la punta del icebearg de tot el treball realitzat.

Els detalls de programació de com es va fer tota aquesta part són molts i variats i es poden obtenir a [2].

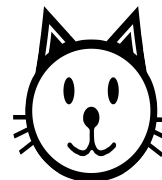
Seqüencialment, un cop teníem tot el marc de programació físic, computacional, i gràfic /interactiu preparat, **només s'havia d'afegir un personatge fictici capaç de moure's per la caixa 2D a voluntat de l'usuari.**

Es va connectar per tant el teclat (amb les tecles clàssiques WASD) amb el moviment d'un personatge confinat en la caixa 2D. Després, es van definir unes normes i uns objectius pel personatge en qüestió i es van posar en pràctica a dues situacions físiques interessants pel paquet tot generant dos modes de joc ben diferenciats. Aquests modes es poden observar al apartat Resultats Finals.

4.6.1 Elecció d'un personatge i el seu objectiu

El personatge havia de ser vistós i es va decidir que fos representat per la cara d'un gat, pot ser com a referència al mític Gat de Schrödinger.

Com s'ha dit, el paquet evolucionava movent-se i diluïnt-se per la caixa, i feia mal al gat si el 'tocava' (Veure apartat de Resultats Finals). Per a què el joc no fos simplement esquivar el paquet, es va otorgar l'objectiu al gat de menjar-se el seu apat preferit: un peix.



Es va pensar que aquesta aprença simpàtica del videojoc atrauria al públic menys relacionat amb la ciència, i que sense adonar-se'n, de sobte estarien aprenent i observant com es comporta un paquet gaussià! Aquest moviment del paquet és observat per l'usuari ja que d'esquivar-lo depen la seva supervivència.

5 Aspecte divulgatiu

Aquestes pràctiques han tingut l'objectiu de crear una eina per divulgar ciència. Abans que el covid arribés arrasant amb tot, els programes realitzats al Quantumlab-UB eren portats a salons d'ensenyament per gent jove. Allà podien interactuar amb els programes amb la presència dels seus creadors. Per tant, era una oportunitat única tant pels membres de les pràctiques, que podien explicar conceptes quàntics amb videojocs i simulacions intuïtives i vistoses, com pels joves estudiants inexperts en el món quàntic, que aprenien amb programes realitzant expressament per ells.

Per un altre banda, no se'ns ha de quedar curt l'abast d'aquest programa, ja que fàcilment podria conviure a un aula d'institut, on un professor de física podria començar a introduir la física quàntica a l'alumnat d'aquesta manera tan especial.

Aquest és llavors, l'últim pas que no s'ha pogut portar a terme de moment: utilitzar el programa per fer divulgació. No obstant, sempre hi ha un pic d'esperança i s'espera poder portar algun dia aquest programa al saló del ensenyament, a les aules d'un institut, o inclús arribar a gent més gran amb la il·lusió d'aprendre.

Ha quedat demostrada això sí la utilitat d'aquest programa per explicar conceptes físics i la seva capacitat d'il·lustrar-los amb el rigor característic d'aquesta ciència. El programa podria també acabar entre aules de la facultat, on algún professor de Mecànica Quàntica volgués, al introduir els paquets gaussians, mostrar com actuen visualment.

6 Valoració de les pràctiques

Aquestes pràctiques han estat enormement útils pel meu creixement com a físic. He après a aprendre a programar, a tractar la convergència d'un mètode numèric i intentar accelerar-lo, a representar visualment resultats i a dubtar d'ells. Per un altre banda, les reunions setmanals han estat emocionants: la possibilitat de tenir a tres investigadors escoltant els teus avenços setmanals en la realització d'un programa divulgatiu no deixa de ser curiós (i més, si a sobre després posen en dubte el que estas fent). Els tutors tenen una gran experiència en el món de la física i els seus consells i la seva supervisió han estat molt útils. El seu paper en les pràctiques és molt destacat i s'ha d'agrair el seu esforç i paciència amb nosaltres, els seus tutelats.

Un treball autònom tan extens com aquest serà probablement una situació comú en el meu futur com a físic i he après a tractar-lo de la manera adequada gràcies en aquestes pràctiques.

Com a punt negatiu, mencionaré que són unes pràctiques molt absorbents (imagino que com totes) i podrien equivaldre perfectament a dos o tres assignatures de sis crèdits. Tot hi així, donada la gran flexibilitat dels tutors, les pràctiques es poden allargar més allà de només un trimestre i, en el nostre cas, ho han fet pràcticament nou mesos, diluïnt-se en el temps i deixant pas a un ambient de treball no tant estressant.

El total d'hores de treball realitzat s'aproxima a unes 280 i tot hi així, han faltat més per acabar d'arrodonir el programa. Algunes coses que sempre he volut afegir es queden en el tinter i sempre quedarà el punt negatiu de no haver pogut presentar el programa en el saló de l'ensenyament, tot i que es probable que això es pugui fer al futur.

Tot i no haver fet tot el que volia fer amb el programa, estic molt satisfet del resultat final i és possible que quan tingui una mica de temps (en algun moment, no sé quan) em dediqui a acabar aquestes petites coses que no he pogut acabar, ja que realment tinc un gran apreci en aquest programa.

References

- [1] Github: Quantumlabs-UB,
<https://github.com/brunojuliana/quantumlabsUB>
- [2] Github: Quatlabs-UB, TennisQuàntic
<https://github.com/brunojuliana/quantumlabsUB/tree/master/TennisQuantic>
- [3] Luis Garrido. *Mecànica Quàntica*
- [4] Mètode ADI
https://en.wikipedia.org/wiki/Alternating_direction_implicit_method
- [5] Kivy
<https://kivy.org/#home>
- [6] Matplotlib: imshow
https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.imshow.html
- [7] Garden: Kivy i Matplotlib
<https://github.com/kivy-garden/garden.matplotlib>
- [8] Numba
<https://numba.pydata.org/>