
Alejandro Sztrajman

Director: Darío M. Mitnik

Métodos numéricos en física atómica

Algoritmos de evolución temporal de la ecuación de Schrödinger

**Tesis de Licenciatura en Ciencias Físicas
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires**

Marzo 2011

We are all atheists about most of the
gods that humanity has ever believed in.
Some of us just go one god further.

Richard Dawkins

Resumen

El desarrollo de métodos no perturbativos, a fines del siglo pasado, supuso un cambio de paradigma en el cual el cálculo de procesos atómicos ha reemplazado a las mediciones experimentales como referencia. El más exitoso de estos métodos, conocido como TDCC (*Time-Dependent Close-Coupling*), está basado en la integración temporal de la ecuación de Schrödinger.

En este trabajo vamos a presentar un análisis de varios algoritmos vinculados con el tratamiento numérico de la ecuación de Schrödinger dependiente del tiempo para un potencial arbitrario. Mostraremos implementaciones concretas de varios esquemas de integración temporal de la ecuación: principalmente el desarrollo explícito en serie del operador de evolución temporal, y el método pseudoespectral *Split-Operator*, imponiendo distintas condiciones de borde al potencial.

Adicionalmente vamos a analizar dos problemas que se manifiestan respectivamente en la evolución temporal de la función de onda en altas energías, y en la proyección en la base de autoestados del Hamiltoniano en bajas energías. Los mismos no son intrínsecos a la ecuación de Schrödinger, y surgen de forma general en la resolución de ecuaciones diferenciales en derivadas parciales. Plantearemos aquí posibles esquemas de cálculo orientados a sortear dichos problemas, y mostraremos su importancia en un caso de aplicación: el cálculo de emisión electrónica en un metal por incidencia rasante de un pulso laser ultracorto.

Índice general

1	Introducción	4
2	Métodos numéricos	8
2.1	Diagonalización de \hat{H}	8
2.2	Diferenciación Numérica	9
2.3	Ecuación de Schrödinger y esquema FTCS	10
2.4	Desarrollo espacial de orden arbitrario	11
2.5	Operador de evolución temporal	13
2.6	Split-Operator	14
2.7	Condiciones de borde	16
2.7.1	Condiciones de borde transparentes	16
2.7.2	Condiciones de borde periódicas	18
2.8	Resumen de métodos	19
3	Análisis y resultados de las implementaciones	20
3.1	Benchmarks	20
3.1.1	Paquete Gaussiano libre	21
3.1.2	Paquete Gaussiano confinado	23
3.1.3	Autoestados del pozo infinito	25
3.2	Animaciones de propagaciones libres con distintas condiciones de borde	27
3.2.1	Bordes rígidos (Método explícito)	27
3.2.2	Condiciones transparentes (Método explícito con TBC)	28
3.2.3	Condiciones periódicas (Split-Operator)	29
3.3	Comparación del tiempo de ejecución de los métodos	30

4	Problemas numéricos	31
4.1	Propagación de paquetes de alta energía	31
4.1.1	Síntomas	31
4.1.2	Explicación y solución	33
4.2	Proyección de paquetes de baja energía	36
4.2.1	Explicación	37
5	Generación de bases arbitrarias	39
5.1	Generación de estados no ligados (dispersión)	40
5.2	Resultados de la proyección en la base generada	41
6	Aplicación práctica: emisión electrónica en una superficie metálica por incidencia rasante de un pulso laser ultracorto	44
6.0.1	Resultados	45
7	Conclusiones	48
A	Unidades atómicas	50
B	Método TDCC	51
C	Notación de Wigner	53
D	Notación de Landau	54
E	Esquema de convergencia Steepest-Descent	55
F	Método de integración de Runge-Kutta	56
G	Código Fuente y detalles de implementación	57
G.1	Split Operator (C++)	57
G.2	Esquema FTCS con TBC (C++)	59
G.3	Generación de estados continuos (Fortran 77)	61
	Bibliografía	64
	Publicaciones derivadas de este trabajo	67
	Agradecimientos	68

Hacia fines del siglo XX, los cálculos más precisos de procesos colisionales estaban basados en métodos perturbativos, limitados por su tratamiento incompleto de las interacciones coulombianas de largo alcance en problemas de tres cuerpos. Desde entonces, se ha desarrollado una serie de métodos no perturbativos que proveen una precisión de referencia en el cálculo de procesos [1]: acoplamiento cercano convergente (CCC - *Convergent Close-Coupling*), acoplamiento cercano en coordenadas hipersféricas (*Hyperspherical Close-Coupling*), matriz R con pseudoestados (RMPS - *R Matrix with PseudoStates*), acoplamiento cercano dependiente del tiempo (TDCC - *Time-Dependent Close-Coupling*), y método de escaleo exterior complejo (ECS - *Exterior Complex Scaling*).

Estos métodos¹, a diferencia de los perturbativos, no involucran la utilización de aproximaciones descontroladas y son capaces —en principio— de reducir arbitrariamente los errores numéricos generados. Las limitaciones en la precisión de cálculo quedan establecidas, en este caso, por un aumento equivalente en el costo computacional involucrado. Por este motivo, el problema más simple de ruptura de ligadura —la ionización de un átomo de hidrógeno por impacto electrónico—, problema que se remonta a los primeros años de la mecánica cuántica, debió esperar hasta 1999 para ser resuelto numéricamente de forma exacta [2]. El cálculo involucrado implicó la inversión de un sistema lineal de dimensión $N = 10^6$ y requirió un poder de cómputo de 580 gigaFLOPS² en un sistema SGI/Cray T3E-900 de 640 procesadores de 450MHz. A partir de este punto, y gracias al continuo avance en el poder de cómputo³, se ha gestado un cambio de paradigma en la física atómica, en el cual el cálculo de procesos ha reemplazado a las mediciones experimentales como referencia [5].

Entre los métodos no perturbativos mencionados, el más exitoso en la actualidad es el TDCC. Este método fue aplicado por primera vez para calcular la sección eficaz total de ionización por impacto electrónico en Hidrógeno, y de doble io-

1. También conocidos como métodos “exactos” o “puramente cuánticos”.

2. gigaFLOPS (FLoating point OPerations per Second) equivale a 10^9 operaciones de punto flotante.

3. 13 años después de la aparición del sistema T3E-900 de Cray, la supercomputadora actual más rápida [3] posee un poder de cómputo de 2,566 petaFLOPS = $2,566 \times 10^{15}$ FLOPS $\approx 2^{12} \times 580$ gigaFLOPS, un valor que supera las predicciones más optimistas de la ley de Moore [4].

nización por impacto fotónico en He [6], que constituyen problemas de ruptura coulombiana de tres cuerpos. En la última década, el método TDCC en una grilla numérica bidimensional ha sido utilizado para calcular secciones eficaces totales y diferenciales de estos mismos procesos, en gran variedad de sistemas atómicos. Recientemente ha sido extendido a grillas tridimensionales para calcular secciones eficaces totales de doble ionización por impacto electrónico en He y H^- , y de triple ionización por impacto fotónico de Li , que constituyen problemas de ruptura coulombiana de cuatro cuerpos. Adicionalmente, el mismo método aplicado a grillas numéricas tetradimensionales ha sido utilizado para calcular la ionización por impacto electrónico en H_2^+ y H_2 , y la doble ionización por impacto fotónico en H_2 , variantes del problema de ruptura coulombiana de tres cuerpos que involucran un campo nuclear no esférico.

El componente básico del método TDCC es la resolución de la ecuación de Schrödinger dependiente del tiempo. Centraremos nuestro estudio en la resolución numérica de esta ecuación, en el caso unidimensional, y para un potencial arbitrario $\hat{V}(x, t)$. En el sistema de unidades atómicas (ver apéndice A) esto es

$$i\frac{\partial\Psi}{\partial t}(x, t) = \hat{H}\Psi(x, t) = \left(-\frac{1}{2}\frac{\partial^2}{\partial x^2} + \hat{V}(x, t)\right)\Psi(x, t) \quad (1.1)$$

La extensión de esta ecuación a más dimensiones nos conduce a un esquema de resolución de las ecuaciones del método TDCC, de la misma forma funcional, cuyo caso bidimensional se encuentra detallado en el apéndice B, incluyendo una de sus aplicaciones principales: el cálculo de ionización por impacto electrónico en sistemas atómicos.

Nuestro objetivo al comenzar este trabajo fue implementar varios algoritmos vinculados al tratamiento numérico de la ecuación de Schrödinger 1.1, a fin de estudiar sus características: performance, ventajas, costo y limitaciones computacionales. La descripción de dichos algoritmos se halla en el capítulo 2, donde incluimos métodos apuntados a describir numéricamente los operadores diferenciales, y a integrar temporalmente la ecuación 1.1. En el capítulo 3 haremos un análisis de la validez y tiempo de ejecución de las implementaciones de los algoritmos, los códigos de cuyas rutinas principales se encuentran en el apéndice G. En el capítulo 4 estudiaremos dos problemas numéricos que pueden aparecer en la implementación de este tipo de algoritmos de evolución, y cuyo análisis y resolución dio lugar a una publicación [7] (ver **publicaciones derivadas de este trabajo**). El primer problema surge del *aliasing* de nuestra función de onda, impuesto por el paso de integración espacial, y se manifiesta visiblemente en paquetes de ondas de alta energía. Por el contrario, el segundo problema es más evidente en paquetes de ondas de baja energía, y está vinculado con la existencia de bordes rígidos en los límites de la grilla numérica. La primera manifestación de este problema se observó en los cálculos de sección eficaz de emisión electrónica realizados por Faraggi *et al.* [8], que reproducimos en la Figura 1.1, donde se comparan los resultados obtenidos por dos métodos distintos: IJV perturbativo (*Impulsive Jellium Volkov*) y TDSE no perturbativo (*Time-Dependent Schrödinger Equation*).

El fenómeno que se estudia en dicho trabajo es la emisión de electrones de valencia en una superficie metálica, por incidencia rasante de un pulso laser ultracorto. Aún cuando la interacción de la radiación con la materia ha sido

objeto de intensa investigación a través del tiempo, el creciente interés en el estudio de la incidencia rasante de pulsos electromagnéticos sobre superficies es motivado en gran medida por los últimos desarrollos en las aplicaciones laser. El surgimiento de pulsos cortos, con duraciones del orden de los femtosegundos y aún de los attosegundos, posibilita en estos días la irradiación sobre superficies metálicas, plásticas y biológicas, permitiendo la caracterización de las mismas [9]. Recientemente en un importante trabajo [10] se ha mostrado que la incidencia de pulsos laser de muy corta duración (del orden de los attosegundos) constituye un poderoso instrumento para investigar la dinámica electrónica en sistemas de materia condensada en tiempo real, debido a que numerosos procesos electrónicos tienen lugar en este intervalo temporal.

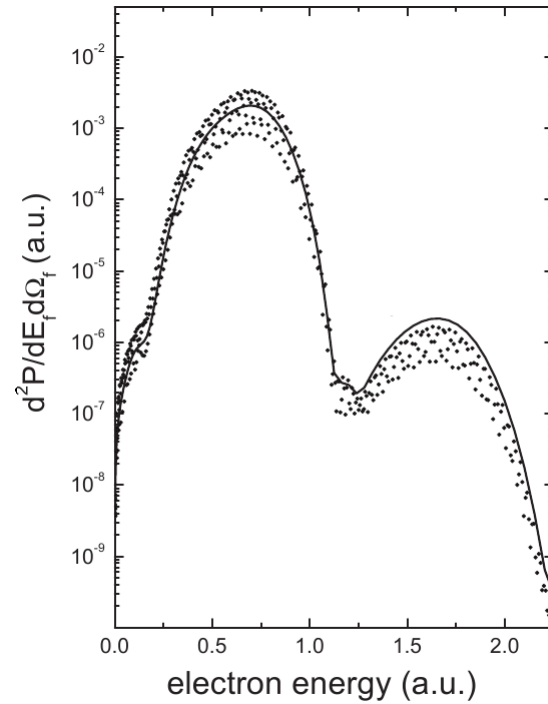


Figura 1.1: Sección eficaz diferencial de emisión electrónica por incidencia rasante de un pulso laser ultracorto (Faraggi *et al.* [8]). Línea sólida: cálculo perturbativo IJV. Puntos: cálculo no perturbativo TDSE.

Como se observa en la Figura 1.1, si bien los cálculos realizados con ambos métodos tienen un acuerdo general satisfactorio, los resultados del método no perturbativo presentan importantes oscilaciones que —mostraremos en el capítulo 4— no poseen un carácter físico y devienen de un problema numérico.

Nos propusimos entonces, como objetivo adicional, eliminar estas oscilaciones en los cálculos evitando que la función de onda Ψ produzca rebotes al alcanzar los límites de la grilla numérica durante su evolución. Los problemas originados en el rebote de la función de onda son comunes en este tipo de cálculos basados en la integración temporal de la ecuación de Schrödinger 1.1 (TDSE y TDCC). La estrategia más sencilla para evitarlos consiste en extender el dominio espacial de manera que ésta no alcance el contorno en el tiempo total de integración. Sin embargo como ya vimos, la principal limitación de los cálculos no perturbativos está dada por los extraordinarios costos computacionales implicados, de modo

que elevar la dimensionalidad de la grilla numérica no resulta una solución óptima. Esto motivó el estudio de un método de implementación de condiciones de borde transparentes (TBC – *Transparent Boundary Conditions* [11]) en los bordes de la grilla numérica, detallado en el capítulo 2.

Sin embargo, en la implementación de TBC las oscilaciones de la Figura 1.1 mostraron ser independientes de los rebotes en los bordes de la grilla, e incluso de la evolución misma de la función de onda. Finalmente, un estudio detallado del problema probó, como veremos en el capítulo 4, que las oscilaciones se originan en la proyección de la función de onda en autoestados del Hamiltoniano obtenidos por diagonalización numérica, debido a la incorrecta imposición de condiciones de borde en estos. Dedicaremos el capítulo 5 a detallar un esquema sencillo de generación de autoestados del Hamiltoniano, de manera de sortear estos problemas de proyección.

Para finalizar, como ejemplo de aplicación práctica de los esquemas estudiados, en el capítulo 6 volveremos sobre los cálculos de emisión electrónica de Faraggi *et al.* [8] con un ejemplo sencillo que aísla los problemas de proyección, a fin de hacer explícita la importancia de su eliminación.

2.1 Diagonalización de \hat{H}

En el caso de Hamiltonianos independientes del tiempo, podemos representar a la función de onda ψ como combinación lineal de los autoestados de la energía a lo largo de la evolución temporal. Sean $\varphi_n(x)$ los autoestados de \hat{H} , es decir:

$$\hat{H}\varphi_n = E_n\varphi_n$$

Podemos representar cualquier función ψ como

$$\psi(x, t) = \sum_n a_n(t)\varphi_n$$

donde hemos considerado que los autoestados φ_n de la energía son constantes en el tiempo, y por tanto la variación temporal de $\psi(x, t)$ proviene de los coeficientes $a_n(t)$ del desarrollo. Si nos remitimos a la ecuación de Schrödinger 1.1, podemos encontrar una forma funcional para la dependencia temporal de dichos coeficientes:

$$\begin{aligned}\hat{H}\psi(x, t) &= i\frac{\partial\psi(x, t)}{\partial t} \\ \hat{H}\sum_n a_n(t)\varphi_n &= i\sum_n \dot{a}_n(t)\varphi_n \\ \sum_n a_n(t)E_n\varphi_n &= i\sum_n \dot{a}_n(t)\varphi_n \\ \sum_n [a_n(t)E_n - i\dot{a}_n(t)]\varphi_n &= 0\end{aligned}$$

Por ortogonalidad de los autoestados de energía φ_n , el término de la sumatoria debe ser nulo para cada n . Obtenemos así una ecuación diferencial de primer

orden en el tiempo para cada uno de los coeficientes $a_n(t)$. La solución es $a_n(t) = a_n(0)e^{-iE_n t}$, con lo cual podemos expresar la evolución temporal de la función $\psi(x, t)$ de forma explícita

$$\psi(x, t) = \sum_n e^{-iE_n t} a_n(0) \varphi_n(x) \quad (2.1)$$

donde los coeficientes $a_n(0)$ se calculan como proyección en la base de autoestados

$$a_n(0) = \langle \varphi_n | \psi \rangle_{t=0} = \int \varphi_n^*(x, 0) \psi(x, 0) dx \quad (2.2)$$

El esquema de evolución temporal resultante indica que si conocemos la proyección de $\psi(x, 0)$ en cada uno de los autoestados del operador Hamiltoniano, podemos calcular la función de onda para cualquier tiempo posterior, computando 2.1. En este cálculo no necesitamos propagar en pasos temporales pequeños, sino que evaluamos la función de onda en cualquier tiempo t arbitrario. Sin embargo, el costo computacional de este método debemos pagarlo al calcular los autoestados φ_n de \hat{H} , para luego realizar la proyección de $\psi(x, 0)$. La operación numérica implicada es la diagonalización de \hat{H} , que puede llegar a tener una complejidad computacional $O(N^3)$ [12]. Aún en el caso de contar con los recursos computacionales suficientes para realizar la diagonalización, debemos remarcar que este procedimiento es válido únicamente para Hamiltonianos independientes del tiempo, poco frecuentes en casos de aplicación reales.

2.2 Diferenciación Numérica

La derivada de primer orden para una función $f(x)$ en un punto x está definida a partir del límite

$$f'(x) = \lim_{\Delta x \rightarrow 0} \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

si existe. Si ahora dividimos el espacio en puntos discretos equiespaciados $x_{i+1} - x_i = \Delta x$, obtenemos la expresión más sencilla para la derivada discreta de primer orden [13]

$$f'_i = \frac{f_{i+1} - f_i}{\Delta x} + O(\Delta x) \quad (2.3)$$

donde hemos abreviado $f(x_i) = f_i$, e indicado el comportamiento del error a través de la notación de Landau (ver apéndice D). La discretización 2.3 es conocida comúnmente como derivada adelantada¹ de primer orden, o bien fórmula de 2-puntos de la derivada, debido a la cantidad de evaluaciones que se realizan sobre la función f .

1. El nombre proviene de su carácter análogo a la derivada por derecha.

El orden de aproximación de la derivada puede mejorarse si expandimos f_{i+1} y f_{i-1} en Taylor de segundo orden y tomamos la diferencia

$$\begin{aligned} f_{i+1} &= f_i + f'_i \Delta x + f''_i \frac{\Delta x^2}{2} + O(\Delta x^3) \\ f_{i-1} &= f_i - f'_i \Delta x + f''_i \frac{\Delta x^2}{2} + O(\Delta x^3) \\ f_{i+1} - f_{i-1} &= 2f'_i \Delta x + O(\Delta x^3) \\ f'_i &= \frac{f_{i+1} - f_{i-1}}{2\Delta x} + O(\Delta x^2) \end{aligned} \quad (2.4)$$

La fórmula 2.4 es conocida como derivada central de primer orden, y presenta una mejora en el orden de convergencia, respecto de la expresión 2.3. Desarrollando f_{i+1} y f_{i-1} hasta tercer orden, y tomando la suma, obtendremos una expresión análoga para la derivada segunda

$$\begin{aligned} f_{i+1} &= f_i + f'_i \Delta x + f''_i \frac{\Delta x^2}{2} + f'''_i \frac{\Delta x^3}{6} + O(\Delta x^4) \\ f_{i-1} &= f_i - f'_i \Delta x + f''_i \frac{\Delta x^2}{2} - f'''_i \frac{\Delta x^3}{6} + O(\Delta x^4) \\ f_{i+1} - f_{i-1} &= 2f'_i \Delta x + O(\Delta x^4) \\ f''_i &= \frac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} + O(\Delta x^2) \end{aligned} \quad (2.5)$$

conocida como derivada central de segundo orden, o bien fórmula de 3-puntos de la derivada segunda. Más adelante veremos que la discretización de derivadas puede generalizarse a fórmulas de más puntos, con mayor orden de aproximación.

2.3 Ecuación de Schrödinger y esquema FTCS

Estamos en condiciones de plantear un esquema de discretización de la ecuación de Schrödinger 1.1. El esquema más sencillo, conocido como FTCS², consiste en tomar la derivada progresiva 2.3 en el tiempo, y la derivada central 2.5 en el espacio.

$$\frac{\partial \psi}{\partial t}(x, t) = \frac{\psi(x, t + \Delta t) - \psi(x, t)}{\Delta t} + O(\Delta t) \quad (2.6)$$

$$\frac{\partial^2 \psi}{\partial x^2}(x, t) = \frac{\psi(x + \Delta x, t) - 2\psi(x, t) + \psi(x - \Delta x, t)}{\Delta x^2} + O(\Delta x^2) \quad (2.7)$$

Introduciendo estas discretizaciones en la ecuación 1.1, obtenemos

$$\hat{H}\psi = -\frac{1}{2} \frac{\psi_{j+1}^n - 2\psi_j^n + \psi_{j-1}^n}{\Delta x^2} + \hat{V}_j \psi_j^n = i \frac{\psi_j^{n+1} - \psi_j^n}{\Delta t}$$

2. *Forward-Time Centered-Space*

donde hemos utilizado la notación $\psi_j^n = \psi(x_j, t_n) = \psi(j \cdot \Delta x, n \cdot \Delta t)$. Si expresamos el operador Hamiltoniano en forma matricial, veremos que en este esquema de discretización, la derivada segunda (correspondiente a la parte cinética de \hat{H}) está representada por una matriz tridiagonal, mientras que el potencial \hat{V} corresponde a una matriz diagonal.

$$\hat{H}_{jk} = -\frac{1}{2} \frac{\delta_{j+1,k} - 2\delta_{j,k} + \delta_{j-1,k}}{\Delta x^2} + \hat{V}_j \delta_{jk} \quad (2.8)$$

$$\hat{H} = \frac{-1}{2\Delta x^2} \begin{pmatrix} -2 & 1 & & & 0 \\ 1 & -2 & 1 & & \\ & 1 & -2 & 1 & \\ & & \ddots & \ddots & \ddots \\ & & & 1 & -2 & 1 \\ 0 & & & & 1 & -2 \end{pmatrix} + \begin{pmatrix} V_1 & & & & 0 \\ & V_2 & & & \\ & & V_3 & & \\ & & & \ddots & \\ & & & & V_{N-1} \\ 0 & & & & & V_N \end{pmatrix}$$

De esta forma, podemos reducir el esquema de integración a una expresión compacta.

$$\begin{aligned} i \frac{\psi_j^{n+1} - \psi_j^n}{\Delta t} &= \hat{H}_{jk} \psi_k^n \\ \psi_j^{n+1} &= \left(I_{jk} - i\Delta t \hat{H}_{jk} \right) \psi_k^n \\ \psi^{n+1} &= \left(I - i\Delta t \hat{H} \right) \psi^n \end{aligned} \quad (2.9)$$

Conociendo la función de onda para todo punto espacial x_j en un tiempo t_n , la ecuación 2.9 nos permite calcular la función para un tiempo posterior $t_n + \Delta t$. La operación implicada en cada paso temporal de integración es la aplicación de una matriz tridiagonal sobre ψ , de costo computacional $O(N)$. El orden de aproximación del algoritmo está dado por la discretización de las derivadas 2.6 y 2.7, es lineal en Δt y cuadrático en Δx . Tomando desarrollos de mayor orden en las derivadas, podremos reducir los errores de cálculo y mejorar la estabilidad del método.

2.4 Desarrollo espacial de orden arbitrario

La ecuación 2.7, que expresa nuestra discretización de la derivada segunda espacial de ψ , implica la evaluación de la propia función ψ en tres puntos distintos, dispuestos simétricamente respecto del punto de evaluación de la derivada. Vamos a ver que podemos generalizar la discretización de la derivada a fórmulas de $2r + 1$ puntos, de las cuales la ecuación 2.7 constituye el caso más sencillo. La forma general de dicho desarrollo será [14]:

$$\psi^{(2)}(x) = \frac{1}{\Delta x^2} \sum_{k=-r}^r c_k^{(r)} \psi(x + k\Delta x) + O(\Delta x^{2r}) \quad (2.10)$$

Nuestro objetivo será encontrar los coeficientes $c_k^{(r)}$. Para ello partimos de los siguientes desarrollos:

$$\psi(x + k\Delta x) = \psi(x) + \psi'(x)k\Delta x + \dots = \sum_{n=0}^{2r+1} \frac{\psi^{(n)}(x)(k\Delta x)^n}{n!} + O(\Delta x^{2r+2})$$

$$\psi(x - k\Delta x) = \psi(x) - \psi'(x)k\Delta x + \dots = \sum_{n=0}^{2r+1} \frac{\psi^{(n)}(x)(k\Delta x)^n(-1)^n}{n!} + O(\Delta x^{2r+2})$$

Sumando ambas expresiones, vemos que los términos impares de las sumatorias se anulan

$$\psi(x + k\Delta x) + \psi(x - k\Delta x) = 2 \sum_{n=0}^r \frac{\psi^{(2n)}(x)(k\Delta x)^{2n}}{(2n)!} + O(\Delta x^{2r+2})$$

$$\psi(x + k\Delta x) + \psi(x - k\Delta x) - 2\psi(x) = 2 \sum_{n=1}^r \frac{\psi^{(2n)}(x)(k\Delta x)^{2n}}{(2n)!} + O(\Delta x^{2r+2})$$

Si sobre esta expresión tomamos $k = 1, 2, \dots, r$ obtenemos r ecuaciones independientes para nuestras r incógnitas constituidas por los $\psi^{(2n)}(x)$.

$$k = 1 \rightarrow \psi(x + \Delta x) + \psi(x - \Delta x) - 2\psi(x) = 2 \sum_{n=1}^r \frac{\psi^{(2n)}(x)(\Delta x)^{2n}}{(2n)!} + O(\Delta x^{2r+2})$$

$$k = 2 \rightarrow \psi(x + 2\Delta x) + \psi(x - 2\Delta x) - 2\psi(x) = 2 \sum_{n=1}^r \frac{\psi^{(2n)}(x)(2\Delta x)^{2n}}{(2n)!} + O(\Delta x^{2r+2})$$

\vdots

$$k = r \rightarrow \psi(x + r\Delta x) + \psi(x - r\Delta x) - 2\psi(x) = 2 \sum_{n=1}^r \frac{\psi^{(2n)}(x)(r\Delta x)^{2n}}{(2n)!} + O(\Delta x^{2r+2})$$

Escrito en forma matricial:

$$2 \begin{pmatrix} \frac{\Delta x^2}{2} & \frac{\Delta x^4}{4!} & \frac{\Delta x^6}{6!} & \dots & \frac{\Delta x^{2r}}{(2r)!} \\ \frac{(2\Delta x)^2}{2} & \frac{(2\Delta x)^4}{4!} & \frac{(2\Delta x)^6}{6!} & \dots & \frac{(2\Delta x)^{2r}}{(2r)!} \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{(r\Delta x)^2}{2} & \frac{(r\Delta x)^4}{4!} & \frac{(r\Delta x)^6}{6!} & \dots & \frac{(r\Delta x)^{2r}}{(2r)!} \end{pmatrix} \begin{pmatrix} \psi^{(2)}(x) \\ \psi^{(4)}(x) \\ \vdots \\ \psi^{(2r)}(x) \end{pmatrix} = \begin{pmatrix} \psi(x + \Delta x) + \psi(x - \Delta x) - 2\psi(x) \\ \psi(x + 2\Delta x) + \psi(x - 2\Delta x) - 2\psi(x) \\ \vdots \\ \psi(x + r\Delta x) + \psi(x - r\Delta x) - 2\psi(x) \end{pmatrix}$$

El problema puede expresarse de forma más sencilla como un sistema lineal de la forma $A_{kn}x_n = b_k$ con las siguientes identidades:

$$A_{kn} = 2 \frac{(k\Delta x)^{2n}}{(2n)!}$$

$$x_n = \psi^{(2n)}(x)$$

$$b_k = \psi(x + k\Delta x) + \psi(x - k\Delta x) - 2\psi(x)$$

Comparando con la expresión 2.10, la solución a nuestro problema de encontrar los coeficientes $c_k^{(r)}$ estará dada por $x = A^{-1}b$. Por inversión de la matriz A obtendremos la discretización de todas las derivadas pares de ψ en términos de $2r + 1$ evaluaciones de la función, y con un orden de aproximación $O(\Delta x^{2r})$.

La tabla 2.1 muestra los coeficientes $c_k^{(r)}$ hasta $r = 7$ (fórmula de 15 puntos) para la derivada segunda [14].

$r \backslash k$	0	1	2	3	4	5	6	7
1	-2	1	0	0	0	0	0	0
2	-5/2	4/3	-1/12	0	0	0	0	0
3	-49/18	3/2	-3/20	1/90	0	0	0	0
4	-205/72	8/5	-1/5	8/315	-1/560	0	0	0
5	-5269/1800	5/3	-5/21	5/126	-5/1008	1/3150	0	0
6	-5369/1800	12/7	-15/56	10/189	-1/112	2/1925	-1/16632	0
7	-266681/88200	7/4	-7/24	7/108	-7/528	7/3300	-7/30888	1/84084

Cuadro 2.1: Coeficientes $c_k^{(r)}$ correspondientes al desarrollo de la derivada segunda en términos de $2r + 1$ evaluaciones de la función. La discretización numérica resulta simétrica espacialmente alrededor del punto de evaluación de la derivada, es decir $c_k^{(r)} = c_{-k}^{(r)}$.

Al considerar el orden de aproximación de la derivada espacial debemos tener en cuenta que, si bien la precisión de la derivada aumenta con el numero de puntos de evaluacion, el orden de desarrollo máximo se acota en la medida en que nos acercamos a los extremos de la grilla numérica.

2.5 Operador de evolución temporal

El esquema de evolución temporal de la ecuación 2.1 puede reescribirse en términos del operador unitario de evolución temporal [15].

$$\hat{U}(t, t_0) = e^{-i\hat{H}(t-t_0)} \quad (2.11)$$

En este caso, la solución de dicha ecuación se puede escribir como

$$\psi(t) = \hat{U}(t, 0)\psi(0) = e^{-i\hat{H}t}\psi(0) \quad (2.12)$$

Si queremos aplicar el operador de evolución temporal explícitamente, debemos desarrollarlo en serie en t:

$$e^{-i\hat{H}t} \simeq 1 - i\hat{H}t + \frac{(-i)^2}{2}\hat{H}^2t^2 + \dots = \sum_{n=0}^M \frac{(-i)^n}{n!}\hat{H}^n t^n \quad (2.13)$$

El orden M en el que cortamos el desarrollo es el orden de aproximación de la derivada temporal en la ecuación de Schrödinger. Consistentemente con el desarrollo FTCS, si elegimos $M = 1$ (primer orden en el desarrollo en t) obtenemos el esquema de evolución de la fórmula 2.9.

En este esquema de evolución, por cada paso de integración temporal debemos realizar M multiplicaciones matriciales $\hat{H}\psi$, de costo $O(N)$. El costo computacional resultante es $O(N.M)$. En el capítulo 3 veremos que, si queremos asegurar la estabilidad del método, el orden de desarrollo M debe aumentar con el tamaño de la grilla, es decir $M = M(N)$.

2.6 Split-Operator

El operador Hamiltoniano puede ser descompuesto en términos cinético y potencial, expresados por los operadores \hat{T} y \hat{V} . De esta forma, podemos expresar el operador de evolución temporal como

$$\hat{U}(dt, 0) = e^{-i(\hat{T}+\hat{V})dt} = e^{\lambda(\hat{T}+\hat{V})}$$

donde suponemos que $\lambda = -idt$ es pequeño.

Los operadores que componen \hat{H} poseen representaciones diagonales en espacios distintos, dado que no conmutan. El operador cinético \hat{T} es diagonal en el espacio de momentos, y el operador potencial $\hat{V}(x)$ es diagonal en el espacio de coordenadas. Si pudiéramos separar al operador de evolución temporal en dos operadores (uno vinculado a \hat{T} , y otro a $\hat{V}(x)$), podríamos transformar ψ convenientemente para que siempre actuara en operadores diagonales. De esta forma, ahorraríamos cierto tiempo de cómputo en las operaciones matriciales. El problema al separar \hat{U} aparece debido a que la no conmutatividad de los operadores \hat{T} y \hat{V} resulta en:

$$e^{\lambda(\hat{T}+\hat{V})} \neq e^{\lambda\hat{T}}e^{\lambda\hat{V}}$$

Lo que haremos a continuación será mostrar una separación del operador de evolución que es válida hasta segundo orden en λ .

$$\begin{aligned} e^{\lambda\hat{V}} &= 1 + \lambda\hat{V} + \frac{\lambda^2}{2}\hat{V}^2 + O(\lambda^3) \\ e^{\lambda\hat{T}/2} &= 1 + \lambda\frac{\hat{T}}{2} + \frac{\lambda^2}{8}\hat{T}^2 + O(\lambda^3) \\ e^{\lambda(\hat{T}+\hat{V})} &= 1 + \lambda(\hat{T} + \hat{V}) + \frac{\lambda^2}{2}(\hat{T} + \hat{V})^2 + O(\lambda^3) \end{aligned} \quad (2.14)$$

Si expandimos el desarrollo 2.14, obtenemos:

$$e^{\lambda(\hat{T}+\hat{V})} = 1 + \lambda(\hat{T} + \hat{V}) + \frac{\lambda^2}{2}(\hat{T}^2 + \hat{V}^2 + \hat{T}\hat{V} + \hat{V}\hat{T}) + O(\lambda^3) \quad (2.15)$$

Ahora veamos que

$$\begin{aligned}
e^{\lambda \hat{T}/2} e^{\lambda \hat{V}} e^{\lambda \hat{T}/2} &= \left(1 + \frac{\lambda \hat{T}}{2} + \frac{\lambda^2 \hat{T}^2}{8} + O(\lambda^3)\right) \left(1 + \lambda \hat{V} + \frac{\lambda^2 \hat{V}^2}{2} + O(\lambda^3)\right) \times \\
&\quad \left(1 + \frac{\lambda \hat{T}}{2} + \frac{\lambda^2 \hat{T}^2}{8} + O(\lambda^3)\right) \\
&= \left(1 + \lambda \hat{V} + \frac{\lambda^2 \hat{V}^2}{2} + \frac{\lambda \hat{T}}{2} + \frac{\lambda^2 \hat{T} \hat{V}}{2} + \frac{\lambda^2 \hat{T}^2}{8} + O(\lambda^3)\right) \times \\
&\quad \left(1 + \frac{\lambda \hat{T}}{2} + \frac{\lambda^2 \hat{T}^2}{8} + O(\lambda^3)\right) \\
&= 1 + \lambda (\hat{T} + \hat{V}) + \frac{\lambda^2}{2} (\hat{T}^2 + \hat{V}^2 + \hat{T} \hat{V} + \hat{V} \hat{T}) + O(\lambda^3)
\end{aligned}$$

Con lo cual, si comparamos con la expresión 2.15 podemos afirmar que

$$e^{\lambda \hat{H}} = e^{\lambda (\hat{T} + \hat{V})} = e^{\lambda \hat{T}/2} e^{\lambda \hat{V}} e^{\lambda \hat{T}/2} + O(\lambda^3)$$

De esta forma, podemos aplicar el operador de evolución temporal como una sucesión de operadores que no mezclan términos no conmutativos. Reescribiendo $\lambda = -idt$, obtenemos la expresión final del operador de evolución, que constituye la aproximación de *Split-Operator* de tercer orden en el tiempo.

$$e^{-i\hat{H}dt} \simeq e^{-i\frac{\hat{T}dt}{2}} e^{-i\hat{V}dt} e^{-i\frac{\hat{T}dt}{2}} \quad (2.16)$$

El esquema de propagación resultante implica transformar la función de onda ψ a un espacio apropiado antes de aplicarle cada uno de los tres operadores del *Split-Operator*. Antes de aplicar cada uno de los términos cinéticos $e^{-i\frac{\hat{T}dt}{2}}$ debemos transformar ψ al espacio de momentos, y al aplicar el término asociado al potencial $e^{-i\hat{V}dt}$ debemos volver a la representación espacial. En la representación adecuada, cada uno de estos operadores es diagonal y su aplicación sobre ψ implica un costo computacional $O(N)$.

La operación que vincula ambas representaciones de ψ es la transformada de Fourier:

$$\begin{aligned}
\psi(p) &= F\psi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \psi(x) e^{-ipx} dx \\
\psi(x) &= \bar{F}\psi(p) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} \psi(p) e^{ipx} dp
\end{aligned}$$

de modo que el esquema de propagación en un paso temporal dt queda finalmente definido como

$$\psi(x, dt) = e^{-i\hat{H}dt} \psi(x, 0) \simeq \bar{F} e^{-i\frac{\hat{T}dt}{2}} F e^{-i\hat{V}dt} \bar{F} e^{-i\frac{\hat{T}dt}{2}} F \psi(x, 0)$$

y la propagación temporal completa se obtiene aplicando sucesivamente el *Split-Operator*. En este esquema, al aplicar dos operadores sucesivos las transformaciones F y \bar{F} de los extremos dan lugar a la identidad:

$$\begin{aligned}
\psi(x, 2dt) &= e^{-i\hat{H}dt} e^{-i\hat{H}dt} \psi(x, 0) \\
&\simeq \bar{F} e^{-i\frac{\hat{T}dt}{2}} F e^{-i\hat{V}dt} \bar{F} e^{-i\frac{\hat{T}dt}{2}} \underbrace{F \bar{F}}_{\mathbb{I}} e^{-i\frac{\hat{T}dt}{2}} F e^{-i\hat{V}dt} \bar{F} e^{-i\frac{\hat{T}dt}{2}} F \psi(x, 0) \\
&\simeq \bar{F} \underbrace{e^{-i\frac{\hat{T}dt}{2}} F e^{-i\hat{V}dt} \bar{F} e^{-i\frac{\hat{T}dt}{2}}}_{\text{paso 2}} \underbrace{e^{-i\frac{\hat{T}dt}{2}} F e^{-i\hat{V}dt} \bar{F} e^{-i\frac{\hat{T}dt}{2}}}_{\text{paso 1}} F \psi(x, 0)
\end{aligned}$$

con lo cual tendremos que computar dos transformadas de Fourier por cada paso de evolución temporal en nuestra integración³. Si utilizamos un algoritmo FFT [16] (*Fast Fourier Transform*) para computar las transformadas, esto implica un costo computacional $O(N \ln N)$ por cada transformación.

2.7 Condiciones de borde

La representación tridiagonal del operador Hamiltoniano (ecuación 2.8) pone de manifiesto una condición de borde predefinida en nuestras implementaciones. La ecuación diferencial 2.8 definida para cada punto de la grilla numérica, es en realidad inválida para los dos puntos extremos. Esto se debe, por ejemplo en el caso del último punto de la grilla, a que no podemos utilizar el valor de la función ψ evaluada en un punto posterior. La ausencia de este punto (y cualquier punto posterior) en el cálculo, equivale a la imposición de un punto de amplitud nula en esa posición, es decir un borde rígido⁴. La ecuación diferencial resultante para el borde de una grilla de J puntos es

$$-\frac{1}{2} \frac{-2\psi_J^n + \psi_{J-1}^n}{\Delta x^2} + \hat{V}_J \psi_J^n = i \frac{\psi_J^{n+1} - \psi_J^n}{\Delta t}$$

En la sección 3.2.1 veremos ejemplos de reflexión de la función de onda, como resultado de este efecto en los bordes.

2.7.1 Condiciones de borde transparentes

En algunos problemas, la existencia de bordes rígidos puede producir efectos indeseados debido al rebote de la función de onda en los extremos. Habitualmente es posible evitar estos efectos aumentando el tamaño de la grilla numérica. Sin embargo, esto produce un incremento en el costo computacional del cálculo. La solución alternativa que vamos a estudiar consiste en *emular* la continuación de la grilla numérica más allá de su extremo, mediante la aplicación de la

3. Sin contar la primera transformada y la última antitransformada, que no se anulan entre sí.

4. También equivalente a un potencial infinito en el borde

condición de borde apropiada. Siguiendo el procedimiento de *Carjan et al.* [17], consideramos la integral de probabilidad en el intervalo de la grilla numérica

$$\rho = \int_a^b |\psi|^2 dx$$

y calculamos su variación temporal [18]

$$\frac{d\rho}{dt} = \frac{i}{2} [\psi^* \frac{\partial \psi}{\partial x} - \psi \frac{\partial \psi^*}{\partial x}]_a^b = F_a - F_b$$

donde F_a y F_b son los respectivos flujos de probabilidad en ambos bordes. Vamos a centrarnos para nuestro análisis en el borde derecho de la grilla, y haremos la suposición de que el comportamiento de la función de onda cerca de este extremo es de la forma $\psi = \psi_0 e^{ik_x x}$, donde ψ_0 y k_x son parámetros complejos desconocidos. Bajo esta suposición, el flujo queda expresado

$$F_b = Re(k_x) |\psi(b)|^2$$

En la medida en que nuestro parámetro $Re(k_x)$ permanezca positivo, tendremos flujo de probabilidad saliente en el borde.

Para el paso n de integración temporal en una grilla de J puntos espaciales, la forma funcional planteada nos impone la siguiente relación para ψ en el borde

$$\frac{\psi_{J+1}^n}{\psi_J^n} = \frac{\psi_J^n}{\psi_{J-1}^n} = e^{ik_x \Delta x} \quad (2.17)$$

El punto ψ_{J+1}^n por supuesto no existe en nuestra grilla numérica⁵. Pero podemos extrapolar su valor de la ecuación 2.17

$$\psi_{J+1}^n = \frac{\psi_J^n}{\psi_{J-1}^n} \psi_J^n = e^{ik_x \Delta x} \psi_J^n \quad (2.18)$$

De esta forma, podemos rearmar la ecuación diferencial 2.8 del punto ψ_J^n

$$\begin{aligned} -\frac{1}{2} \frac{\psi_{J+1}^n - 2\psi_J^n + \psi_{J-1}^n}{\Delta x^2} + \hat{V}_J \psi_J^n &= i \frac{\psi_J^{n+1} - \psi_J^n}{\Delta t} \\ -\frac{1}{2} \frac{(e^{ik_x \Delta x} - 2)\psi_J^n + \psi_{J-1}^n}{\Delta x^2} + \hat{V}_J \psi_J^n &= i \frac{\psi_J^{n+1} - \psi_J^n}{\Delta t} \end{aligned}$$

Así, podemos emular la existencia de un punto ψ_{J+1}^n sin necesidad de incluirlo en nuestra grilla numérica. La matriz del Hamiltoniano resultante tiene un punto de diferencia con respecto a 2.8

$$\hat{H} = \frac{-1}{2\Delta x^2} \begin{pmatrix} -2 & 1 & & & & 0 \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ 0 & & & & 1 & e^{ik_x \Delta x} - 2 \end{pmatrix} + \begin{pmatrix} V_1 & & & & & 0 \\ & V_2 & & & & \\ & & V_3 & & & \\ & & & \ddots & & \\ & & & & V_{N-1} & \\ 0 & & & & & V_N \end{pmatrix}$$

5. Como ya dijimos, en la ecuación diferencial del punto ψ_J^n esto es equivalente a $\psi_{J+1}^n = 0$.

2.7.2 Condiciones de borde periódicas

El método del Split-Operator (2.6) utiliza en su planteo la descomposición de la función de onda en ondas viajeras $e^{\pm i k x}$, por medio de una transformada de Fourier aplicada sobre el segmento finito definido por la grilla numérica. Como veremos en los ejemplos de la sección 3.2.2, la transformada discreta de Fourier⁶ otorga al desarrollo en serie una periodicidad que se corresponde con la grilla numérica. Dicho en otras palabras, la función de onda que sale por uno de los extremos de la grilla, reaparece inmediatamente por el otro.

Si bien este método cuenta –como veremos más adelante– con algunas características numéricas prometedoras, la condición de periodicidad intrínseca que impone en la evolución puede significar un problema en la aplicación a potenciales específicos no periódicos.

6. Equivalente a una serie de Fourier.

2.8 Resumen de métodos

Cuadro 2.2: Resumen comparado de las características principales de los métodos presentados.

Método	Error espacial	Error temporal	Costo
Diagonalización	$O(\Delta x^{2r+2})$	-	$O(N^3)$
Esquema FTCS	$O(\Delta x^2)$	$O(\Delta t)$	$O(N)$
Operador de evolución	$O(\Delta x^{2r+2})$	$O(\Delta t^M)$	$O(N.M(N))$
Split-Operator	-	$O(\Delta t^3)$	$O(N \ln N)$

3

Análisis y resultados de las implementaciones

En este capítulo vamos a mostrar algunos casos sencillos de ejecución de los métodos de evolución implementados. A manera de ejemplo, vamos a poder observar algunas animaciones típicas obtenidas para distintas condiciones de borde y potenciales. Para algunos casos conocidos, vamos a poder comparar nuestra evolución numérica con la solución analítica, para verificar la convergencia de los métodos. Por último, haremos una medición de los costos computacionales de los métodos, y compararemos los resultados con las estimaciones teóricas de la tabla 2.2. En todos los casos, los parámetros de ejecución (en principio Δt , Δx , N) se han determinado de forma de corroborar la conservación de la norma unitaria de ψ durante la evolución temporal¹. En la implementación del método explícito 2.13 podemos indicar un orden M fijo para el desarrollo en serie en t del operador de evolución temporal, o bien podemos indicar una precisión fija, de manera que M aumente hasta conseguir la precisión pedida².

3.1 Benchmarks

Algunos ejemplos conocidos de potenciales sencillos nos proveen una forma de verificar los distintos esquemas de integración temporal estudiados hasta el momento. Este tipo de pruebas es conocido como *Benchmark*.

-
1. Esta condición es verificada en las implementaciones.
 2. En este caso, el programa desarrolla en serie el operador de evolución temporal hasta que todos los elementos del último orden son menores en módulo a la precisión fija deseada.

3.1.1 Paquete Gaussiano libre

Consideremos la propagación de un paquete de ondas Gaussiano en un potencial libre ($V(x, t) = 0$). Su forma funcional está dada por

$$\psi(x, t = 0) = \frac{1}{\sqrt{\sqrt{\pi}\sigma_x}} e^{-\frac{1}{2}\left(\frac{x-x_0}{\sigma_x}\right)^2 + ik_0 x} \quad (3.1)$$

donde σ_x es el ancho inicial del paquete, k_0 su velocidad, y x_0 la posición inicial de su centro.

En este caso, contamos con la solución analítica completa del problema [19], es decir que conocemos la expresión de $\psi(x, t)$ para todos los valores de x y t . Vamos a utilizar la expresión de la densidad de probabilidad³

$$|\psi(x, t)|^2 = \frac{e^{\frac{(x-k_0 t)^2}{\sigma_x^2 + t^2/\sigma_x^2}}}{\sqrt{\pi(\sigma_x^2 + t^2/\sigma_x^2)}}, \quad (3.2)$$

para comparar con la solución numérica obtenida por el método explícito 2.13, aplicado al Hamiltoniano tridiagonal con condiciones de borde transparentes de la sección 2.7.1. La noción de error que vamos a establecer para realizar la comparación entre funciones estará vinculada a la diferencia absoluta entre las densidades de probabilidad, sumada a lo largo de toda la grilla numérica:

$$\epsilon(t) = \int_{\text{grilla}} ||\psi_{\text{num}}(x, t)|^2 - |\psi(x, t)|^2| dx \quad (3.3)$$

donde ψ_{num} y ψ son las funciones de onda numérica y analítica, respectivamente.

El ejemplo específico que vamos a ilustrar es el de un paquete Gaussiano centrado en la grilla numérica ($x_0 = 15$ au) que se mueve hacia la derecha hasta desaparecer por el borde transparente. En las Figuras 3.1 y 3.2 podemos observar la evolución del paquete, comparada con el cálculo analítico, para grillas numéricas de 300 y 1500 puntos respectivamente. Finalmente, en la Figura 3.3 presentamos los resultados de los cálculos de error en función del tiempo, para varios tamaños de la grilla numérica que van de 300 a 1500 puntos.

3. La expresión de un paquete de ondas Gaussiano utilizada por C. Cohen-Tannoudji difiere de nuestra definición por un factor $\sqrt{2}$ en la dispersión espacial σ_x .

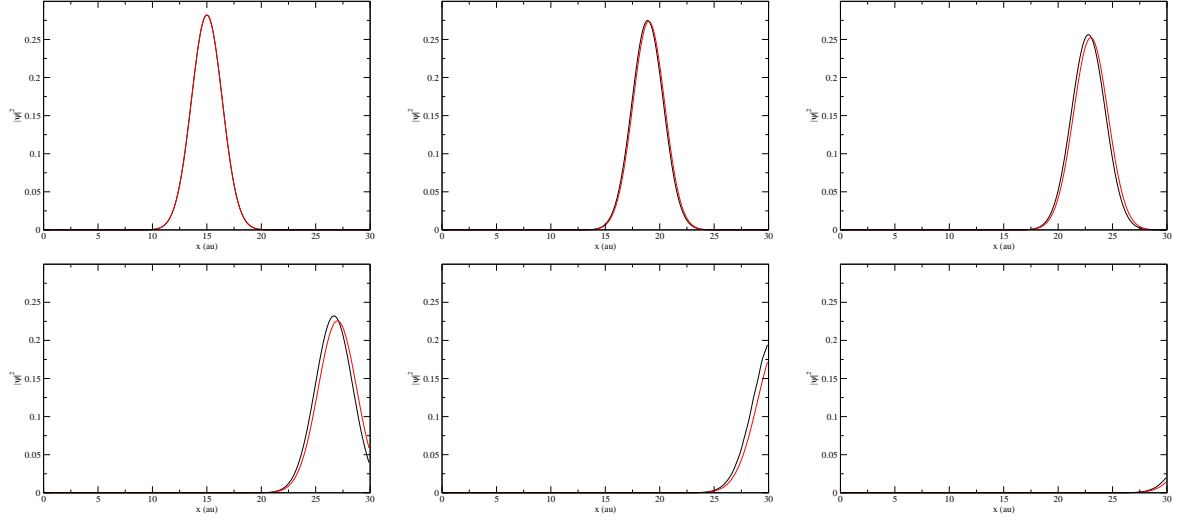


Figura 3.1: $|\psi|^2$ vs x para la evolución temporal de un paquete gaussiano ($x_0 = 15$ au, $\sigma_x = 2$ au, $k_0 = 4$ au), en un potencial libre. En negro mostramos la solución numérica, calculada con el esquema de operador de evolución temporal 2.13, y con condiciones de borde transparentes. En rojo mostramos la solución analítica del problema [19]. Los tiempos mostrados son $t = 0 ; 1 ; 2 ; 3 ; 4 ; 5$ au. $N = 300$, $\Delta x = 0,1$ au.

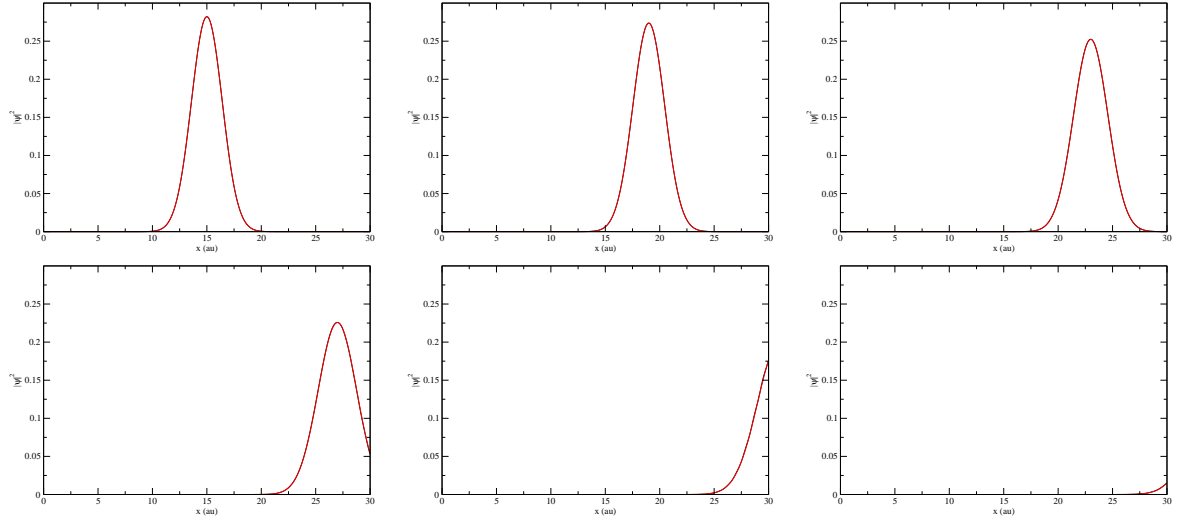


Figura 3.2: $|\psi|^2$ vs x para la evolución temporal de un paquete gaussiano ($x_0 = 15$ au, $\sigma_x = 2$ au, $k_0 = 4$ au), en un potencial libre. En negro mostramos la solución numérica, calculada con el esquema de operador de evolución temporal 2.13, y con condiciones de borde transparentes. En rojo mostramos la solución analítica del problema [19]. Los tiempos mostrados son $t = 0 ; 1 ; 2 ; 3 ; 4 ; 5$ au. $N = 1500$, $\Delta x = 0,02$ au.

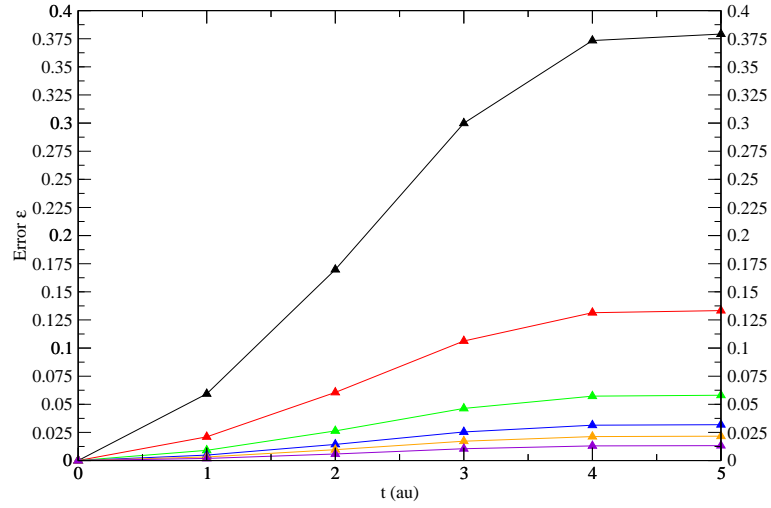


Figura 3.3: Errores en función del tiempo, calculados a partir de la expresión 3.3, para la propagación por el esquema de operador de evolución temporal 2.13 (discretización espacial tridiagonal), de un paquete de ondas Gaussiano ($x_0 = 15$ au, $\sigma_x = 2$ au, $k_0 = 4$ au) en un potencial libre. Los colores corresponden a distintos tamaños de la grilla numérica utilizada. Negro: 300 puntos. Rojo: 500 puntos. Verde: 750 puntos. Azul: 1000 puntos. Naranja: 1200 puntos. Violeta: 1500 puntos.

En todos los casos el error de cálculo resulta una función creciente del tiempo. Sin embargo, en la medida en que aumentamos el número de puntos de la grilla numérica, vemos que el orden de crecimiento se reduce de forma marcada ⁴ : de $\epsilon \sim 0,175$ (300 puntos) a $\epsilon \sim 0,006$ (1500 puntos), si nos centramos en el tiempo $t = 2$ au, donde el paquete de ondas casi no ha comenzado a desaparecer. Más adelante, en el capítulo 4, haremos un análisis detallado de este tipo de problemas, centrando nuestra atención en el vínculo existente entre el tamaño de la grilla numérica y los errores de representación.

3.1.2 Paquete Gaussiano confinado

La evolución numérica del paquete de ondas Gaussiano de la ecuación 3.1 sin la inclusión de bordes transparentes corresponde, como fue discutido en la sección 2.7, a la integración temporal en un potencial tipo pozo infinito. En este caso podemos realizar un cálculo de error por comparación con el método de evolución temporal por proyección en autoestados de \hat{H} , estudiado en la sección 2.1. No será necesario, para este problema en particular, realizar numéricamente la diagonalización del operador Hamiltoniano, dado que sus autoestados son las funciones 3.4 mencionadas en la sección anterior. Por lo tanto, la evolución de ψ estará dada por la fórmula 2.1, y el único cálculo numérico involucrado será la proyección en autoestados del paquete Gaussiano inicial, dada por la ecuación 2.2.

4. Para tener una idea del significado de los valores de error, podemos observar que para dos funciones normalizadas totalmente desvinculadas entre sí, la fórmula 3.3 arrojará un valor máximo $\epsilon_{max} = 2$.

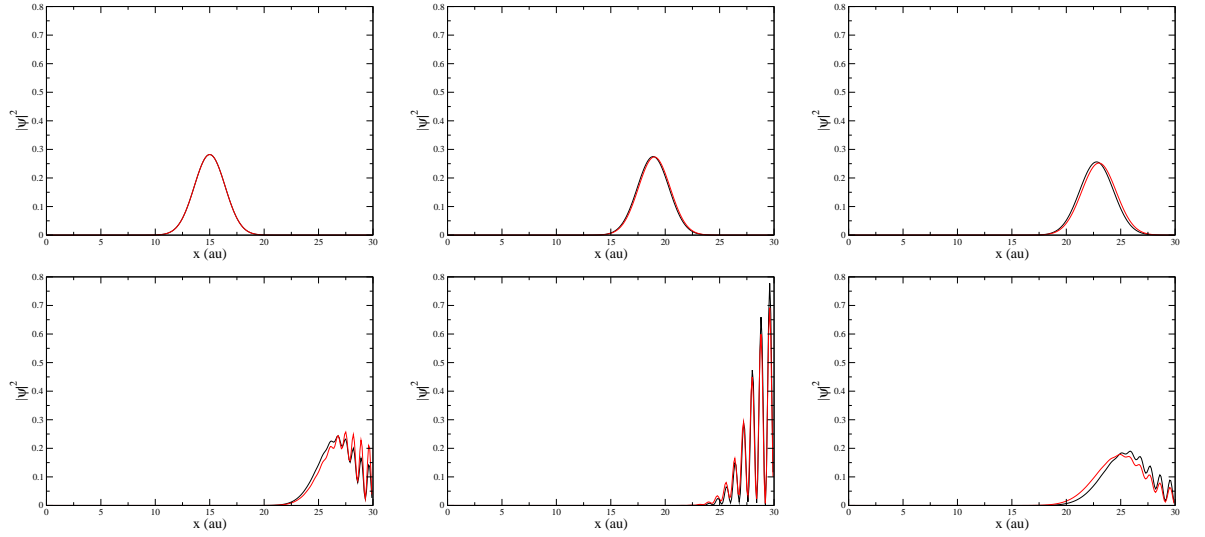


Figura 3.4: $|\psi|^2$ vs x para la evolución temporal de un paquete Gaussiano ($x_0 = 15$ au, $\sigma_x = 2$ au, $k_0 = 4$ au) en un pozo infinito. En negro mostramos la solución numérica, calculada con el esquema explícito 2.13 (discretización espacial tridiagonal). En rojo mostramos la solución obtenida por proyección en los autoestados de \hat{H} . Los tiempos mostrados son $t = 0 ; 1 ; 2 ; 3 ; 4 ; 5$ au. $N = 300$, $\Delta x = 0,1$ au.

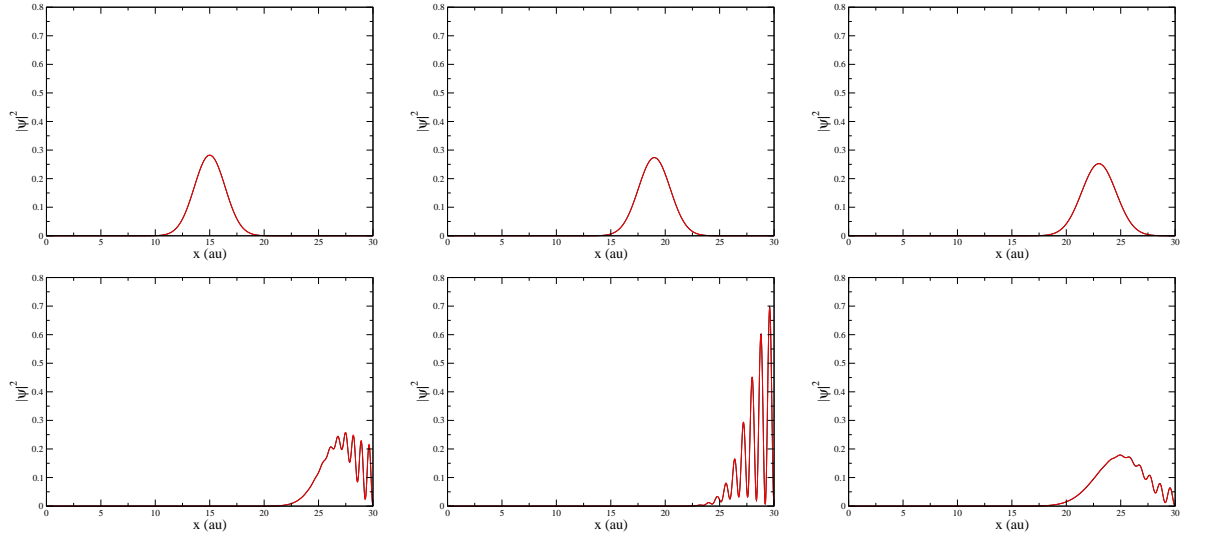


Figura 3.5: $|\psi|^2$ vs x para la evolución temporal de un paquete Gaussiano ($x_0 = 15$ au, $\sigma_x = 2$ au, $k_0 = 4$ au) en un pozo infinito. En negro mostramos la solución numérica, calculada con el esquema explícito 2.13 (discretización espacial tridiagonal). En rojo mostramos la solución obtenida por proyección en los autoestados de \hat{H} . Los tiempos mostrados son $t = 0 ; 1 ; 2 ; 3 ; 4 ; 5$ au. $N = 1500$, $\Delta x = 0,02$ au.

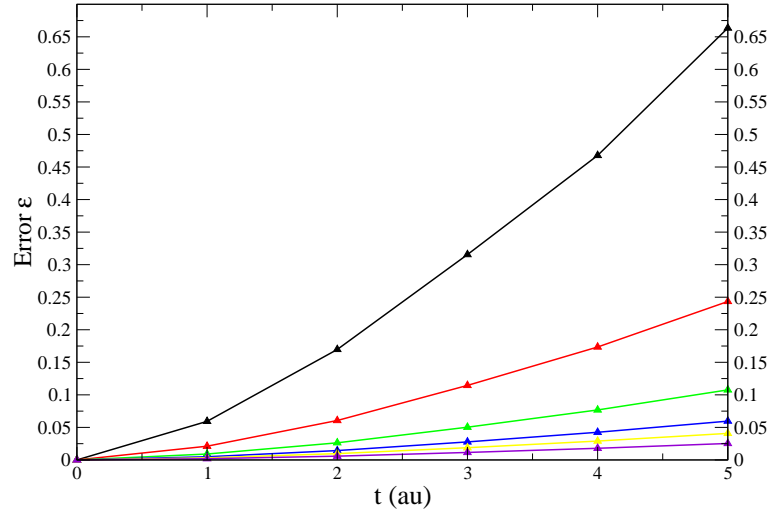


Figura 3.6: Errores en función del tiempo, calculados a partir de la expresión 3.3, para la propagación por el esquema explícito 2.13 (discretización espacial tridiagonal) de un paquete de ondas Gaussiano ($x_0 = 15$ au, $\sigma_x = 2$ au, $k_0 = 4$ au) en un pozo infinito. Los colores corresponden a distintos tamaños de la grilla numérica utilizada. Negro: 300 puntos. Rojo: 500 puntos. Verde: 750 puntos. Azul: 1000 puntos. Naranja: 1200 puntos. Violeta: 1500 puntos.

3.1.3 Autoestados del pozo infinito

Los autoestados de una caja de potencial de largo L son funciones conocidas de la forma

$$\varphi_n(x) = \sqrt{\frac{2}{L}} \sin\left(\frac{n\pi x}{L}\right) \quad (3.4)$$

cuya densidad de probabilidad debe permanecer inalterada a lo largo de la evolución temporal. En las Figuras 3.7a y 3.8a mostramos la evolución temporal de los dos primeros autoestados del pozo infinito de potencial ($n = 1$ y $n = 2$ en la ecuación 3.4), para una grilla de 300 puntos. Luego, en las Figuras 3.7b y 3.8b podemos observar la evolución temporal del error según la ecuación 3.3 para ambos autoestados, con distintos tamaños de grilla.

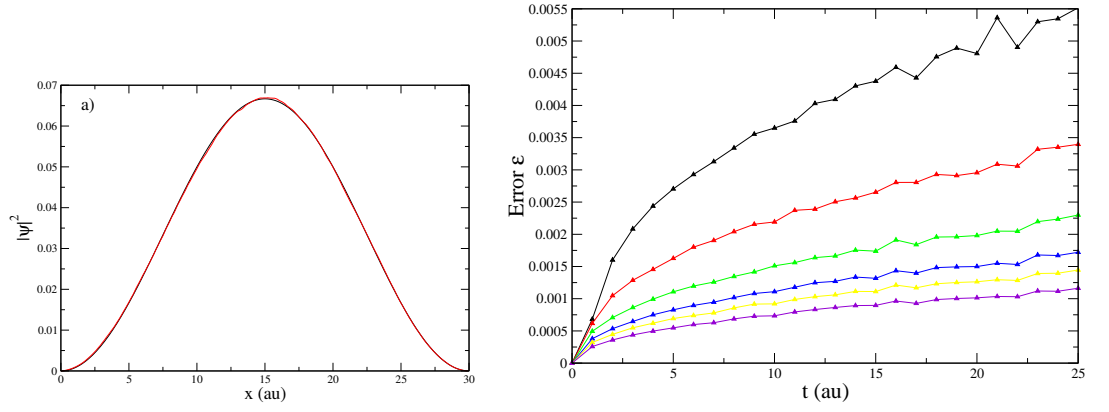


Figura 3.7: a) $|\psi|^2$ vs x para la evolución temporal del primer autoestado del pozo infinito de potencial 3.4, calculada con el esquema explícito 2.13. Los tiempos mostrados son $t = 0$ au (en negro) y $t = 25$ au (en rojo). $N = 300$, $\Delta x = 0,1$ au.

b) Errores en función del tiempo, calculados a partir de la expresión 3.3. Los colores corresponden a distintos tamaños de la grilla numérica utilizada. Negro: 300 puntos. Rojo: 500 puntos. Verde: 750 puntos. Azul: 1000 puntos. Naranja: 1200 puntos. Violeta: 1500 puntos.

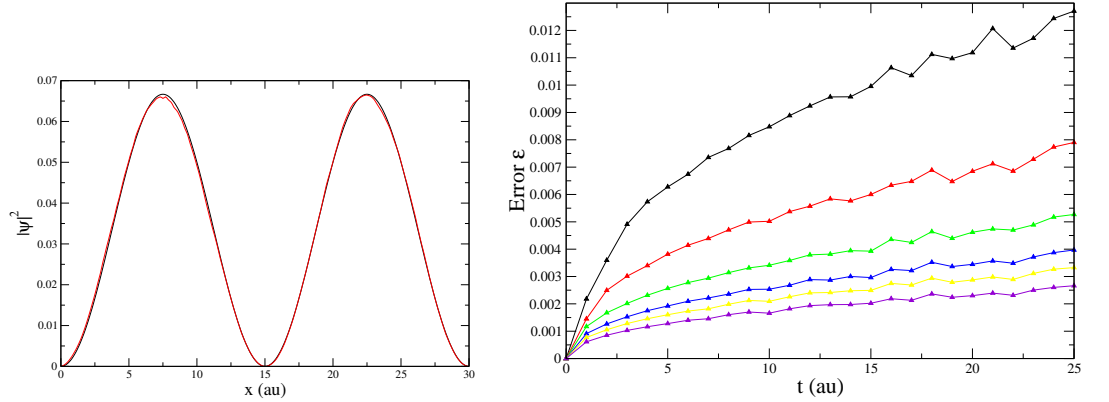


Figura 3.8: a) $|\psi|^2$ vs x para la evolución temporal del segundo autoestado del pozo infinito de potencial 3.4, calculada con el esquema explícito 2.13. Los tiempos mostrados son $t = 0$ au (en negro) y $t = 25$ au (en rojo). $N = 300$, $\Delta x = 0,1$ au.

b) Errores en función del tiempo, calculados a partir de la expresión 3.3. Los colores corresponden a distintos tamaños de la grilla numérica utilizada. Negro: 300 puntos. Rojo: 500 puntos. Verde: 750 puntos. Azul: 1000 puntos. Naranja: 1200 puntos. Violeta: 1500 puntos.

Consistentemente con los cálculos realizados sobre paquetes Gaussianos, los errores crecen con el tiempo de evolución, pero es posible reducirlos mejorando la resolución de nuestra grilla numérica⁵.

5. El orden de desarrollo temporal del operador de evolución temporal 2.13 puede ser incrementado también para reducir el error de cálculo, al costo de un aumento en el tiempo computacional.

3.2 Animaciones de propagaciones libres con distintas condiciones de borde

3.2.1 Bordes rígidos (Método explícito)

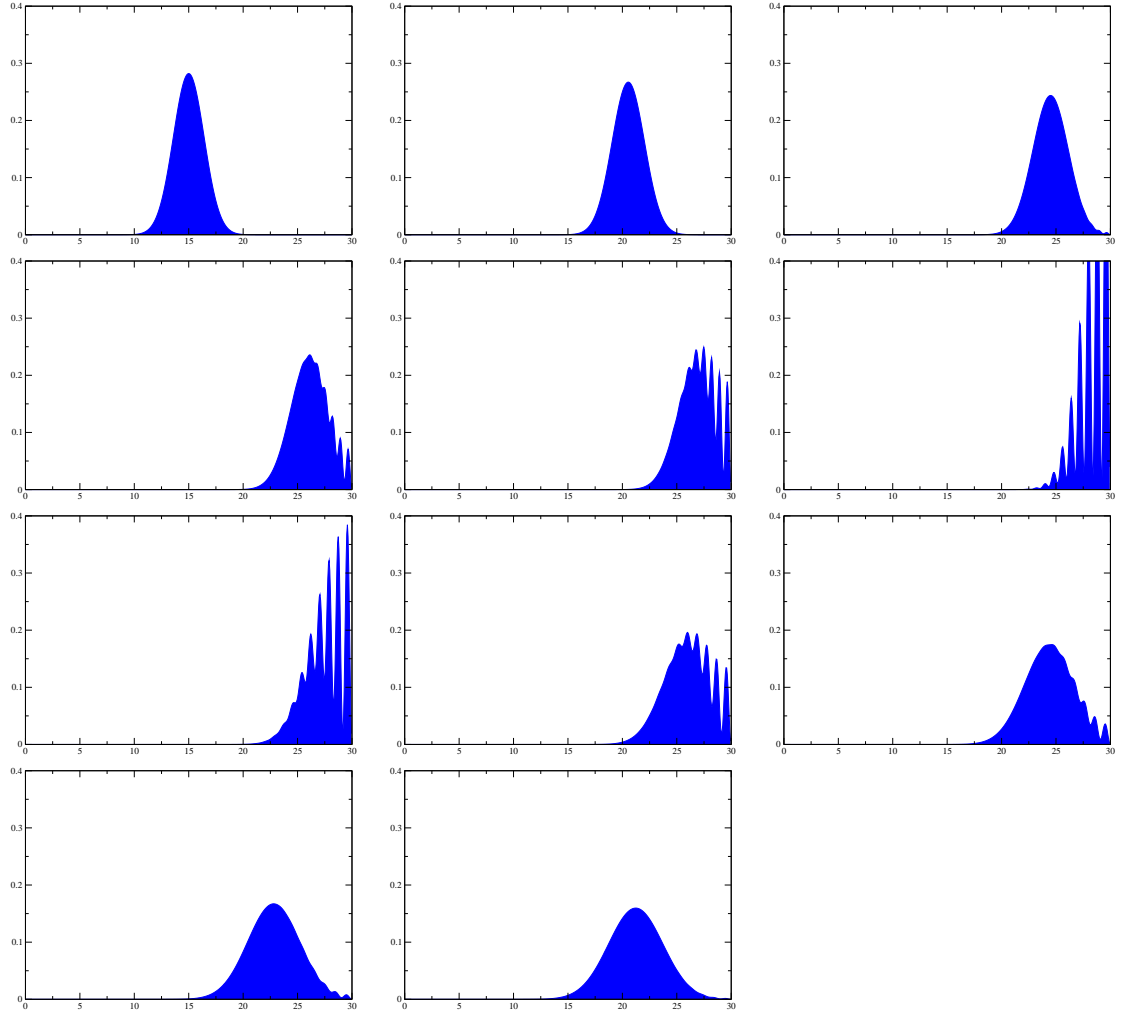


Figura 3.9: $|\psi|^2$ vs x para la reflexión de un paquete gaussiano libre ($x_0 = 15$ au, $\sigma_x = 2$ au, $k_0 = 4$ au), sobre el extremo de la grilla numérica (discretización espacial tridiagonal). Los tiempos mostrados son $t = 0 ; 1,4 ; 2,4 ; 2,8 ; 3,0 ; 4,0 ; 4,4 ; 4,8 ; 5,2 ; 5,6 ; 6,0$ au. $N = 500$, $\Delta x = 0,06$ au.

3.2.2 Condiciones transparentes (Método explícito con TBC)

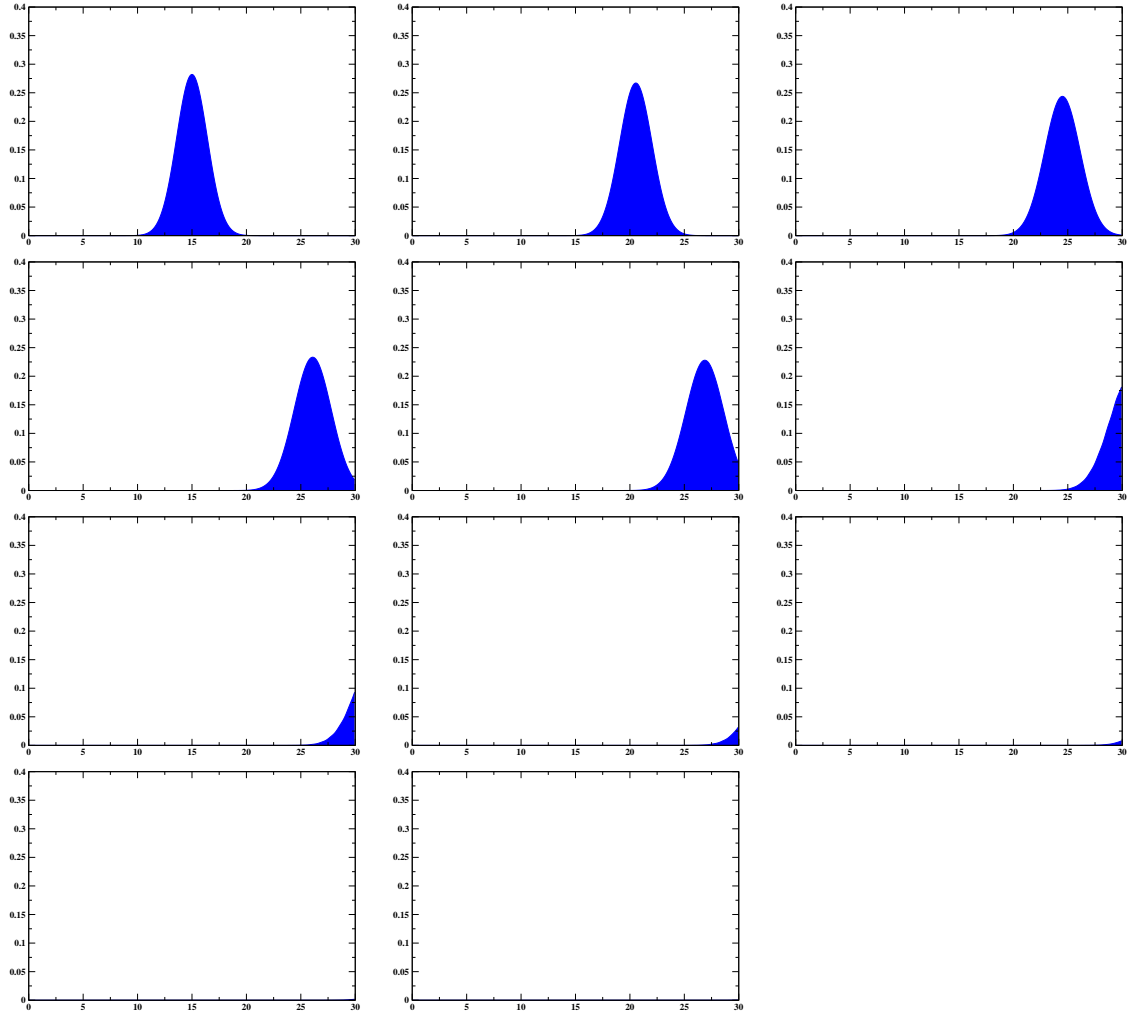


Figura 3.10: $|\psi|^2$ vs x para la incidencia de un paquete gaussiano libre ($x_0 = 15$ au, $\sigma_x = 2$ au, $k_0 = 4$ au), sobre el extremo transparente de la grilla numérica (discretización espacial tridiagonal). Los tiempos mostrados son $t = 0 ; 1,4 ; 2,4 ; 2,8 ; 3,0 ; 4,0 ; 4,4 ; 4,8 ; 5,2 ; 5,6 ; 6,0$ au. $N = 500$, $\Delta x = 0,06$ au.

3.2.3 Condiciones periódicas (Split-Operator)

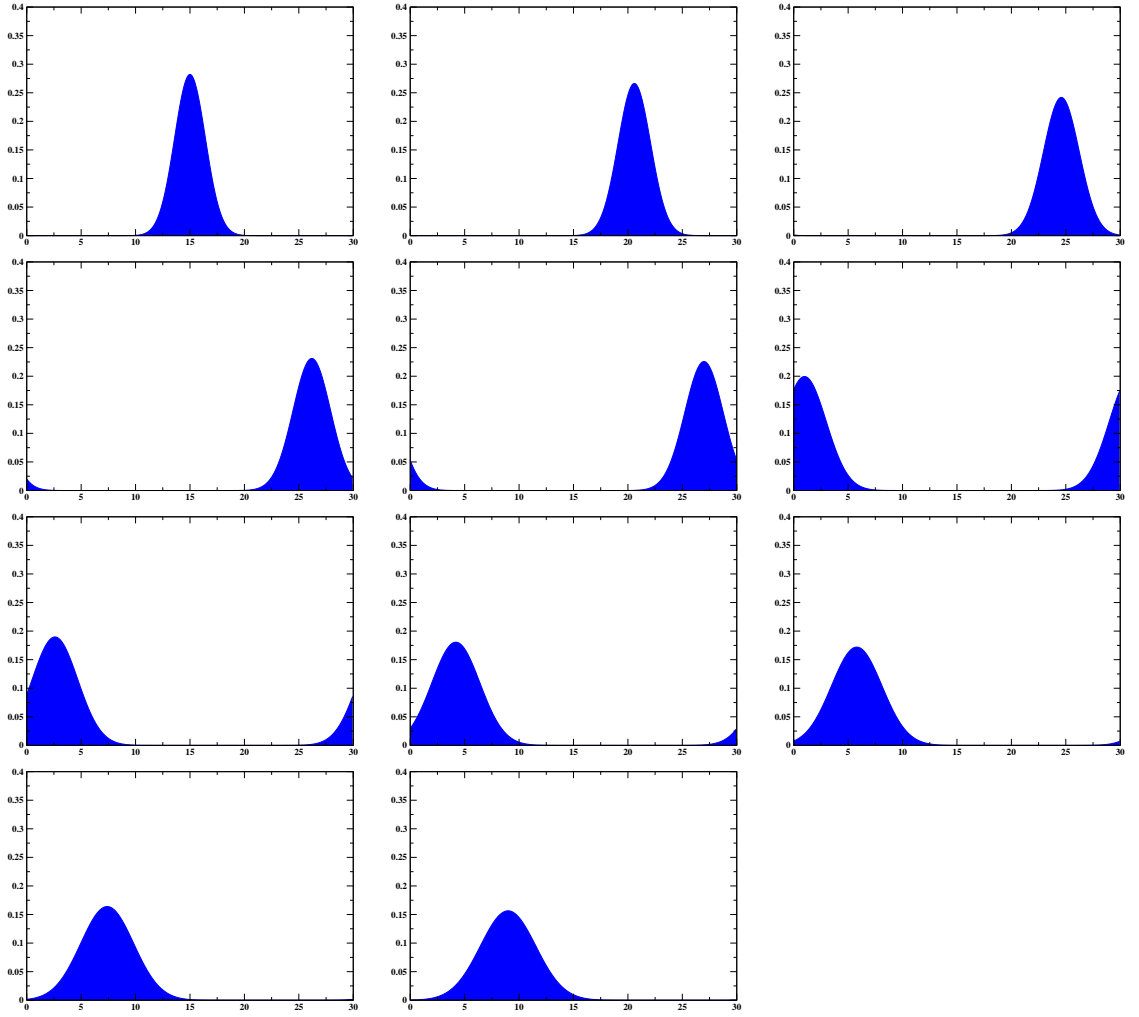


Figura 3.11: $|\psi|^2$ vs x para la evolución temporal de un paquete gaussiano libre ($x_0 = 15$ au, $\sigma_x = 2$ au, $k_0 = 4$ au), mediante el método de *Split-Operator*. La condición de borde de este método es intrínsecamente periódica (ver sección 2.7.2). Los tiempos mostrados son $t = 0 ; 1,4 ; 2,4 ; 2,8 ; 3,0 ; 4,0 ; 4,4 ; 4,8 ; 5,2 ; 5,6 ; 6,0$ au. $N = 500$, $\Delta x = 0,06$ au.

Como anticipamos en 2.7.2, la propagación del método *Split-Operator* es intrínsecamente periódica. Esto puede significar un problema si nuestro deseo es que el paquete de ondas interactúe una única vez con un potencial no periódico (e.g., en un problema de dispersión).

3.3 Comparación del tiempo de ejecución de los métodos

En la Figura 3.12 podemos ver mediciones del costo computacional (tomado como el tiempo de cálculo) en función del tamaño N de la grilla numérica. En el caso del método explícito 2.13 (en azul) el comportamiento puede ser aproximado por una función cuyo orden asintótico es $O(N^2 \ln N)$. A partir de la tabla 2.2, podemos inferir un comportamiento $M(N) \sim N \cdot \ln(N)$ para el orden de desarrollo del operador de evolución temporal⁶.

En el caso del método de Split-Operator (en rojo), si bien el comportamiento es más errático debido a la utilización de diversos esquemas de optimización en la implementación de las transformaciones de Fourier, podemos aproximar con una función de orden asintótico $O(N \cdot \ln N)$, en concordancia con las predicciones teóricas de la tabla 2.2.

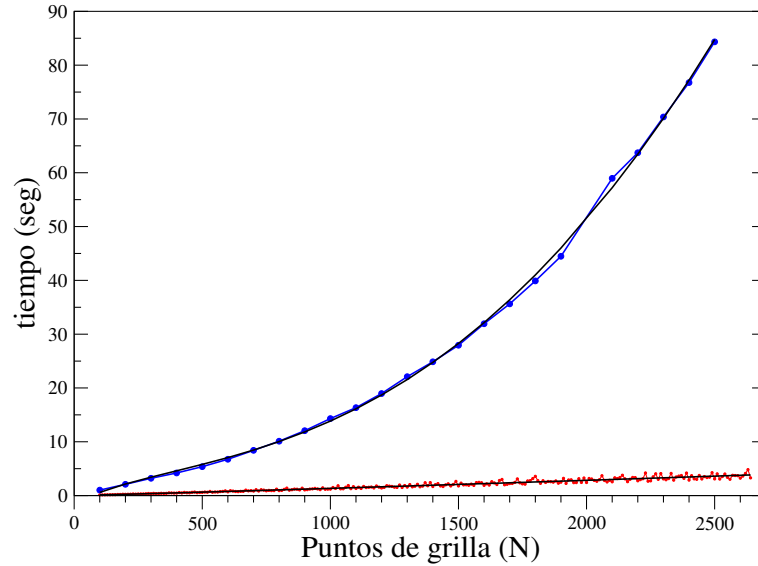


Figura 3.12: Tiempo de ejecución en función de la cantidad de puntos de la grilla (N). En azul: implementación del método explícito (Operador de evolución temporal 2.13). La discretización espacial de \hat{H} es tridiagonal (1er orden). En rojo: implementación del método Split-Operator. En negro: regresiones. Las corridas fueron realizadas en un procesador Intel Core Duo T2300, con los siguientes parámetros: $dt = 0,002$ au, $L = N \cdot \Delta x = 30$ au (fijo), tiempo total $t_f = 0,6$ au.

6. Esto es cierto al menos si queremos mantener fija la precisión.

4

Problemas numéricos

4.1 Propagación de paquetes de alta energía

En el capítulo 2 estudiamos varios esquemas de integración temporal de la ecuación de Schrödinger 1.1. El problema numérico que veremos a continuación no está vinculado a la utilización de un esquema de integración particular, sino que es consecuencia directa de la descripción discretizada que poseemos de la función de onda. El síntoma que se manifiesta es una reducción en la velocidad de propagación de la función de onda respecto del valor esperado teóricamente, más evidente en la medida en que avanzamos hacia altas energías.

4.1.1 Síntomas

A modo de ejemplo, consideraremos la evolución temporal libre ($V(x, t) = 0$) de un paquete de ondas Gaussiano, dado por la ecuación 3.1, con parámetros $\sigma_x = 1$ au, $k_0 = 10$ au, y $x_0 = 10$ au. Utilizaremos para los cálculos una grilla numérica de $N = 500$ puntos, con un paso espacial $\Delta x = 0,06$ au. En la Figura 4.1a podemos ver la densidad de probabilidad numérica $|\psi|^2$ en función de x para distintos tiempos de la evolución: $t = 0 ; 0,15 ; 0,3 ; 0,45 ; 0,6 ; 0,75$ au.

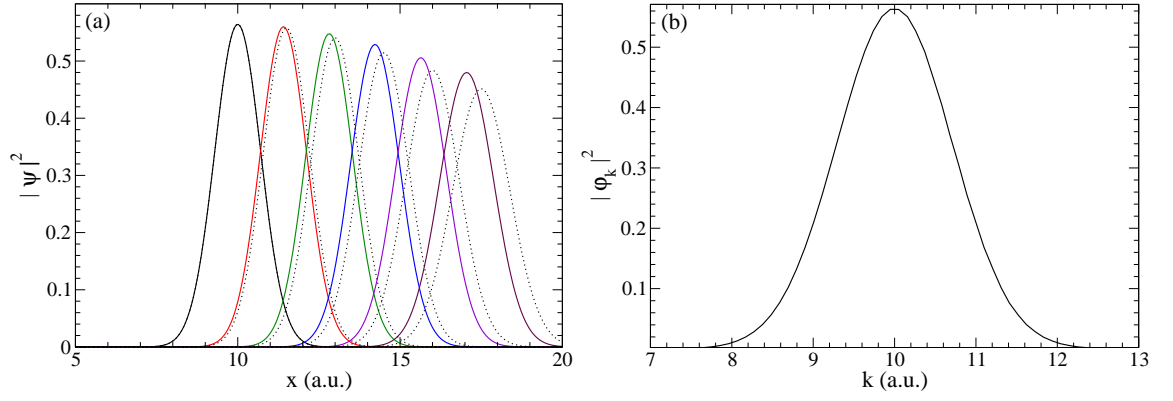


Figura 4.1: (a) Propagación de un paquete de ondas libre, $x_0 = 10$ au, $\sigma_x = 1$ au, $k_0 = 10$ au. Línea sólida: propagación numérica para tiempos $t = 0 ; 0,15 ; 0,3 ; 0,45 ; 0,6 ; 0,75$ au. Línea Punteada: solución analítica. (b) Distribución de momentos del paquete de ondas.

En la Figura 4.1a podemos ver un ligero retardo del paquete de ondas numérico respecto al analítico. Sin embargo, la distribución de momentos del paquete de ondas, que se muestra en la Figura 4.1b, no revela aún ninguna anomalía: la transformada de Fourier es otra función Gaussiana, centrada en $k_0 = 10$ au, y de ancho $\sigma_k = \sigma_x^{-1} = 1$ au.

Un ejemplo más dramático se muestra en la Figura 4.2a, correspondiente a un paquete de velocidad $k_0 = 45$ au.

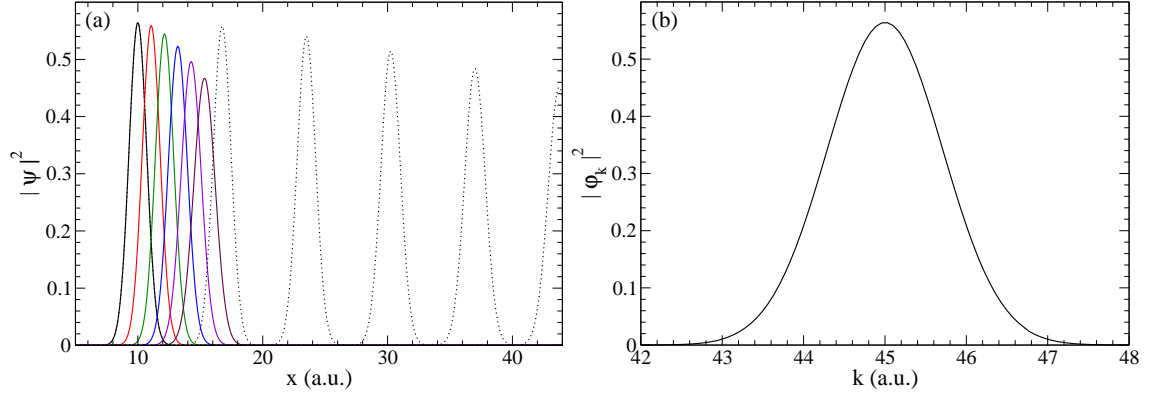


Figura 4.2: (a) Propagación de un paquete de ondas libre, $x_0 = 10$ au, $\sigma_x = 1$ au, $k_0 = 45$ au. Línea sólida: propagación numérica para tiempos $t = 0 ; 0,15 ; 0,3 ; 0,45 ; 0,6 ; 0,75$ au. Línea Punteada: solución analítica. (b) Distribución de momentos del paquete de ondas.

A tiempo $t = 0,75$ au, el centro del paquete de ondas numérico se desplazó alrededor de 5,3 au de su posición inicial, indicando una velocidad de $k \approx 7,07$ au, apenas un sexto del valor teórico. La transformada de Fourier de ψ se muestra en la Figura 4.2b, pero nuevamente no da indicios sobre el origen del problema.

A fin de resumir los diferentes resultados, hemos comparado las velocidades

numéricas¹ con las velocidades teóricas. Para el cálculo hemos utilizado una derivada espacial de 3-puntos en el esquema de propagación. Los resultados se muestran en la Figura 4.3, donde el retardo de los paquetes numéricos se hace evidente.

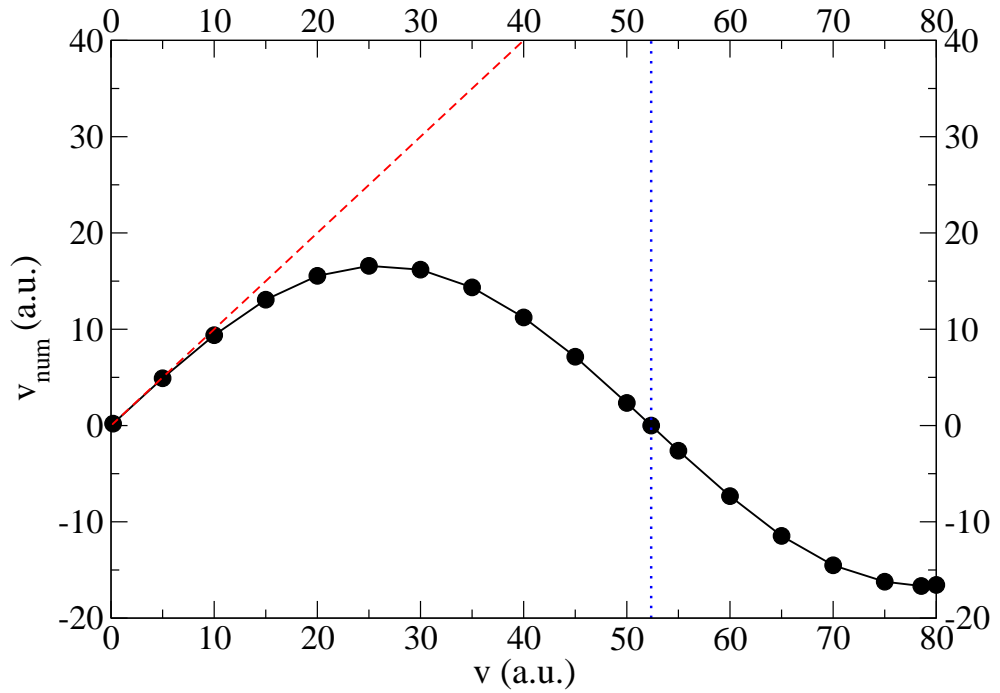


Figura 4.3: Velocidades numéricas v_{num} de los paquetes de ondas propagados, comparados con los valores teóricos v . Primer orden en las derivadas espaciales, y $\Delta x = 0,06$ au. La línea rayada corresponde al comportamiento ideal ($v_{\text{num}} = v$). La línea punteada marca la velocidad $v = 52,36$ au, para la cual $v_{\text{num}} = 0$.

La curva v_{num} vs. v de la Figura 4.3 resulta sinusoidal, en lugar de ser una línea recta de pendiente unitaria. Únicamente para velocidades bajas el resultado de la propagación es razonable. Encontramos que la máxima velocidad alcanzada por nuestro paquete de ondas es $v_{\text{num}} \approx 17$ au. Para una velocidad teórica $v = 52,36$ au, el paquete de ondas numérico resultará estacionario. Y para velocidades mayores, pasará a tener velocidad negativa.

4.1.2 Explicación y solución

El origen del problema está relacionado con la discretización espacial, que imposibilita la representación de funciones cuya longitud de onda sea menor al intervalo discreto Δx . En términos del número de onda, esto es equivalente a

$$k_{\text{max}} \Delta x = \pi \quad (4.1)$$

En la medida en que nuestro paquete de ondas tenga componentes de momento mayor a k_{max} , la evolución temporal no será exacta. En nuestros cálculos, $\Delta x = 0,06$ au, con lo cual la ecuación 4.1 prevé una velocidad máxima representable $k_{\text{max}} = 52,36$ au. La evolución de un paquete de ondas con este

1. Dadas por el cociente entre el desplazamiento del centro del paquete y el tiempo.

valor de velocidad media se muestra en la Figura 4.4a. El resultado numérico es efectivamente un paquete estacionario.

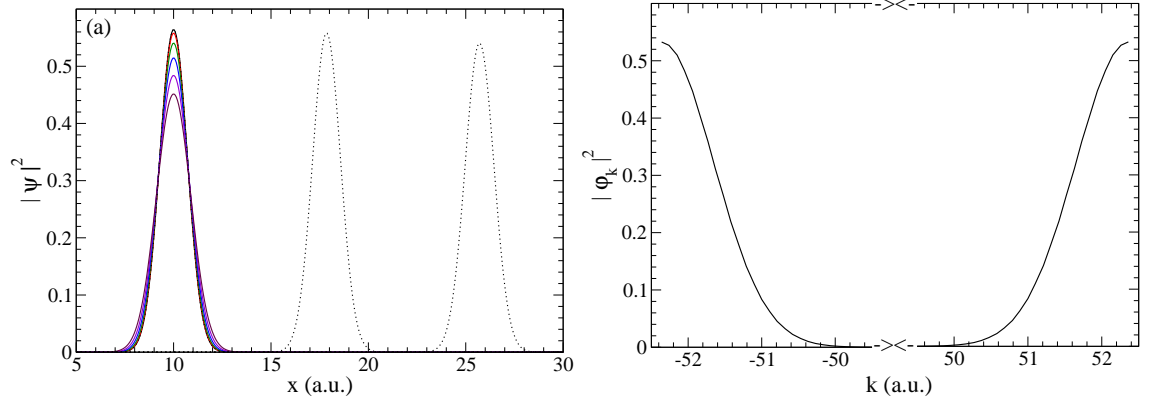


Figura 4.4: (a) Propagación de un paquete de ondas libre, $x_0 = 10$ au, $\sigma_x = 1$ au, $k_0 = 52,36$ au. Línea sólida: propagación numérica para tiempos $t = 0 ; 0,15 ; 0,3 ; 0,45 ; 0,6 ; 0,75$ au. Línea Punteada: solución analítica (únicamente para $t = 0 ; 0,15 ; 0,3$ au). (b) Distribución de momentos del paquete de ondas.

La transformada de Fourier del paquete de ondas, mostrada en la Figura 4.4b, exhibe el mismo problema de *aliasing* debido a que responde a la misma discretización. Toda componente del paquete de ondas cuyo número de onda k_i está afuera del intervalo $(-\pi/\Delta x, \pi/\Delta x)$ es tratada como $k_i = k'_i + 2\pi n/\Delta x$, con $n \in N$ tal que $k'_i \in (-\pi/\Delta x, \pi/\Delta x)$, produciendo un recorte en las velocidades, y reduciendo así la velocidad de grupo del paquete de ondas.

A partir de la ecuación 4.1, tenemos una condición para las funciones de onda representables: el mayor momento involucrado k_m no debe superar el límite representable² $k_{max} = \pi/\Delta x$. Sin embargo, en los ejemplos de las Figuras 4.1 y 4.2 esta condición se cumple de sobra, y el retardo de los paquetes de ondas es evidente. Para evitar estos efectos, será necesario establecer condiciones adicionales sobre la discretización.

Consideremos la ecuación de Schrödinger 1.1 para una evolución libre ($V(x, t) = 0$). Las soluciones son de la forma $\varphi_k = \sin(kx)$, donde la derivada segunda en 1.1 corresponde a

$$\frac{\partial^2}{\partial x^2} \varphi_k = -k^2 \varphi_k \quad (4.2)$$

En el esquema de diferencias finitas de 3-puntos 2.7, esto es

$$\begin{aligned} \frac{\partial^2 \varphi_k}{\partial x^2}(x) &= \frac{\varphi_k(x + \Delta x) - 2\varphi_k(x) + \varphi_k(x - \Delta x)}{\Delta x^2} + O(\Delta x^2) \\ &= \frac{\sin(k(x + \Delta x)) + \sin(k(x - \Delta x)) - 2\varphi_k(x)}{\Delta x^2} + O(\Delta x^2) \\ &= \frac{2\sin(kx)\cos(k\Delta x) - 2\varphi_k(x)}{\Delta x^2} + O(\Delta x^2) \\ &= -\frac{2(1 - \cos(k\Delta x))}{\Delta x^2} \varphi_k(x) + O(\Delta x^2) \end{aligned}$$

2. Es suficiente pedir que los coeficientes acompañando a estados de mayor momento sean despreciables. Esto es, según *Goldberg et al.* [20], con probabilidad 10^{-5} .

donde hemos hecho uso de la identidad $\sin(\alpha + \beta) = \sin \alpha \cos \beta + \sin \beta \cos \alpha$. Dado que $k\Delta x$ es pequeño, podemos aproximar $\cos(k\Delta x) \approx 1 - \frac{(k\Delta x)^2}{2} + \frac{(k\Delta x)^4}{24}$ de manera que

$$\frac{\partial^2 \varphi_k}{\partial x^2}(x) \approx -k^2(1 - k^2\Delta x^2/12)\varphi_k(x)$$

Para que esta última expresión constituya una buena aproximación de la ecuación 4.2, debemos requerir

$$\Delta x^2 \ll \frac{12}{k_m^2} \quad (4.3)$$

De esta forma, debemos escoger el intervalo de discretización Δx de acuerdo a la distribución de momentos que vamos a representar.

Una alternativa para aumentar nuestra resolución es mejorar el esquema de discretización de la derivada espacial. Este hecho se evidencia en el gráfico v_{num} vs. v de la Figura 4.5, donde podemos observar el límite de representación para varios esquemas de discretización de la derivada espacial, incluido el esquema split-operator.

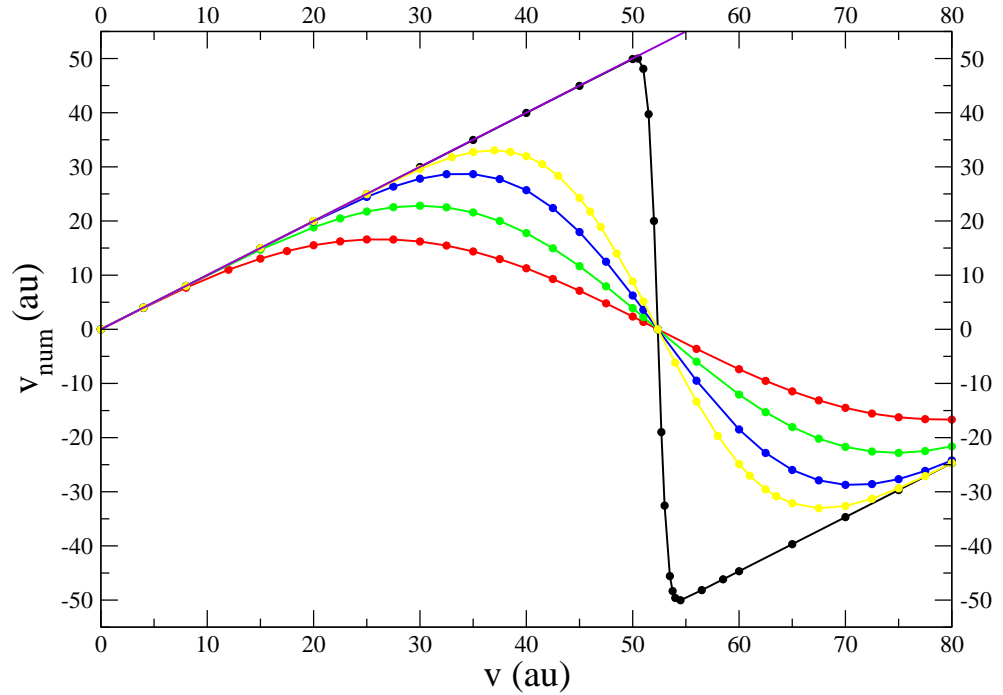


Figura 4.5: Velocidades numéricas v_{num} de los paquetes de ondas propagados, comparadas con los valores teóricos v . Mostramos las curvas v_{num} vs v para varios esquemas de discretización de la derivada espacial, con $\Delta x = 0,06$ au. Rojo: derivada de 3-puntos. Verde: derivada de 5-puntos. Azul: derivada de 9-puntos. Amarillo: derivada de 15-puntos. Negro: esquema Split-Operator. Violeta: teórico.

Evidentemente la cota de la ecuación 4.3 aumenta en la medida en que mejoramos la representación de la derivada espacial en nuestro esquema de integración. En este sentido, los métodos tienden hacia el resultado del Split-Operator³, aco-

3. El método del Split-Operator no implica aproximaciones sobre la derivada espacial, más allá de la discretización propia de la grilla numérica finita.

tado únicamente por la condición 4.1.

4.2 Proyección de paquetes de baja energía

Un problema distinto ocurre en la integración de paquetes de baja energía. Esta vez el problema no se manifiesta en la propagación temporal en sí, sino en la proyección de la función de onda ψ en la base de autoestados del Hamiltoniano.

Como ya discutimos en la sección 2.1, para \hat{H} independiente del tiempo la evolución temporal de ψ puede desarrollarse como

$$\psi(x, t) = \sum_k e^{-iE_k t} c_k(0) \varphi_k(x) \quad (2.1)$$

donde los coeficientes $c_k(0)$ se calculan como proyección en la base de autoestados (ecuación 2.2). Los coeficientes de proyección c_k son útiles por varias razones. Si \hat{H} no depende del tiempo, la invarianza de los coeficientes a lo largo de la evolución temporal provee una buena prueba sobre la calidad de los métodos de integración y de los autoestados obtenidos numéricamente. En el caso de procesos de colisión, donde $\hat{H} = \hat{H}(t)$, los coeficientes c_k (ahora también dependientes del tiempo) proveen una descripción sobre la probabilidad de cada uno de los canales inelásticos de scattering.

La ecuación de Schrödinger 1.1 está definida normalmente en un dominio ilimitado, con condiciones de borde definidas. En general es necesario restringir el problema original a un intervalo finito, a fin de que sea tratable numéricamente. También es necesario reducir la cantidad de autoestados de la base a un número finito. En el esquema de propagación numérica por diagonalización (sección 2.1) la receta de integración numérica es la siguiente:

- Para un potencial dado, se construye la matriz de \hat{H} por discretización de la coordenada espacial.
- Por diagonalización numérica [12] se obtiene un número finito de autoestados φ_k y autoenergías E_k .
- La función de onda inicial $\psi(x, 0)$ se proyecta numéricamente en los autoestados φ_k , y se obtienen los coeficientes c_k .
- Finalmente se calcula la función $\psi(x, t)$ para cualquier tiempo utilizando la fórmula 2.1.

Si bien el esquema de diagonalización es bastante directo, los resultados pueden diferir bastante de lo esperado. Como ejemplo, mostramos en la Figura 4.6b el valor de los coeficientes c_k en función de la energía E_k , para el paquete de ondas Gaussiano dado en la ecuación 3.1, con velocidad $k_0 = 0,5$ au y centrado en la grilla numérica ($N = 1100$ puntos, $x_0 = x_{medio} = 33$ au).

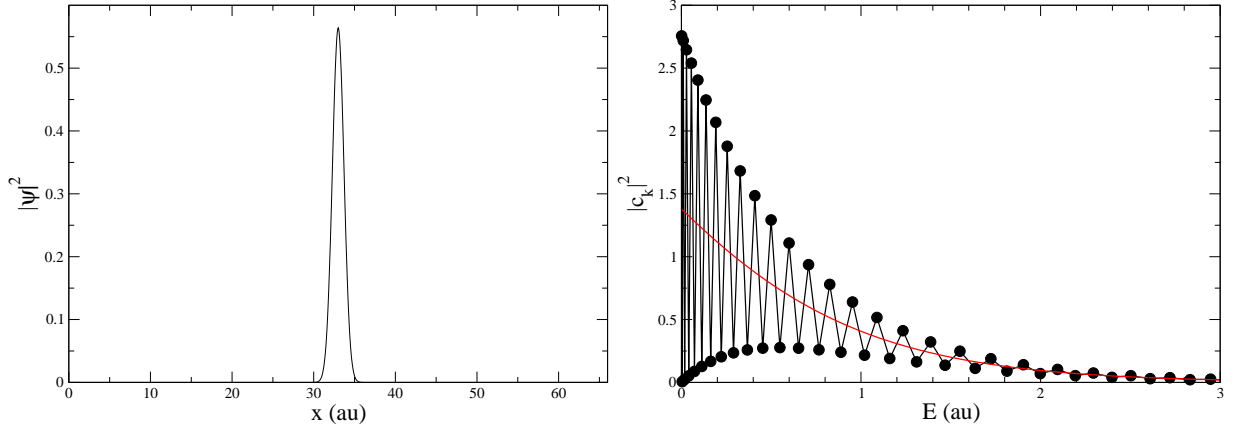


Figura 4.6: a) Paquete de ondas Gaussiano inicial 3.1 con $k_0 = 0,5$ au, $x_0 = 33$ au. b) Descomposición espectral numérica. Curva suave: descomposición correcta.

La descomposición espectral de la figura muestra importantes oscilaciones, cuando debería ser una exponencial decreciente suave. La evidencia más clara de que las oscilaciones son un efecto espúreo se muestra en la Figura 4.7b, donde la descomposición espectral del paquete de ondas ha variado radicalmente como consecuencia del cambio de la posición inicial $x_0 = 23$ au. La descomposición en el espectro de energías no puede depender de la posición del paquete, por tanto los coeficientes c_k calculados deben ser físicamente incorrectos.

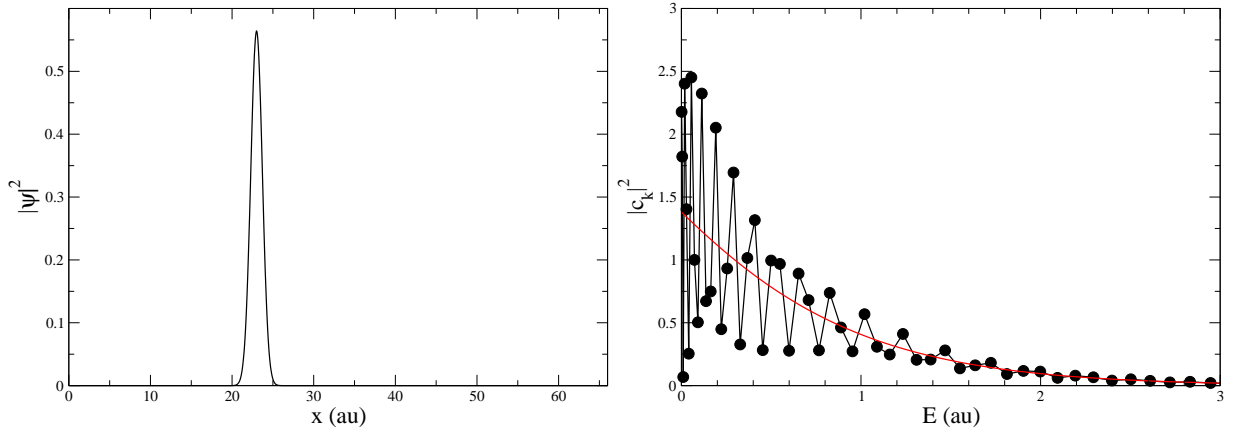


Figura 4.7: a) Paquete de ondas Gaussiano inicial 3.1 con $k_0 = 0,5$ au, $x_0 = 23$ au. b) Descomposición espectral numérica. Curva suave: descomposición correcta.

4.2.1 Explicación

El origen de las oscilaciones reside en la simetría existente en el potencial respecto a su centro. Es un hecho conocido que los autoestados de un potencial simétrico son pares o impares frente a la inversión de la coordenada espacial respecto al eje de simetría [21]. Esto es cierto incluso en el caso del potencial libre, cuando éste es discretizado en una grilla numérica. En el espacio de coordenadas, una matriz numérica finita de \hat{H} implica que tanto el operador como sus autoestados son nulos afuera de los confines de la matriz (ver sección 2.7).

El resultado inevitable es que no estamos resolviendo la ecuación de Schrödinger para el potencial deseado, sino para una versión del mismo confinada a una caja: estamos imponiendo condiciones de contorno rígidas ($\psi(0, t) = \psi(x_{\max}, t) = 0$) en los bordes de nuestra grilla numérica.

Siguiendo con el ejemplo del potencial libre, esto lleva a la obtención de autoestados con paridad definida, alternadamente par e impar. El paquete de ondas inicial centrado en la grilla numérica no constituye una función estrictamente par, debido a la presencia del factor de velocidad e^{ik_0x} (ecuación 3.1). Sin embargo para bajas energías, este factor tiende a cero y el paquete de ondas se va asemejando a una función par. Dado que la proyección entre estados de distinta paridad es nula, los resultados numéricos oscilan alrededor de los valores correctos, de acuerdo a la paridad de los autoestados del Hamiltoniano. Cuando desplazamos al paquete del centro de la grilla, como se ve en la Figura 4.7, la simetría espacial no desaparece completamente. El eje de simetría de la función inicial se encuentra corrido, y por tanto las proyecciones no son ya entre funciones de paridad definida. Es posible reducir las oscilaciones mediante este corrimiento de la función de onda inicial, pero no eliminarlas por completo.

Como ya mencionamos, el efecto discutido hasta el momento sobre las proyección en autoestados no se evidencia fácilmente porque no tiene consecuencias en la propagación temporal de la función. De hecho, en las Figuras 4.6a y 4.7a, el paquete reconstruido a partir de la expresión 2.1 es indistinguible de su expresión analítica. Asimismo, podemos afirmar que este problema está completamente desvinculado del método de integración temporal escogido, dado que los espectros de proyección mostrados en las figuras corresponden al estado inicial de la función, previo a la evolución temporal.

Si bien la evolución temporal de la función de onda no se ve afectada por este fenómeno, en procesos dependientes del tiempo ($\hat{H}(t)$) es importante realizar una correcta proyección, dado que los coeficientes espectrales corresponden a las probabilidades de los distintos procesos.

La solución a nuestro problema de proyección consistirá en romper la simetría de los estados estacionarios. Para ello, en el capítulo siguiente vamos a desarrollar algunos algoritmos involucrados en la generación de bases de autoestados del Hamiltoniano con condiciones de borde especiales, sin una simetría definida.

5

Generación de bases arbitrarias

En este capítulo vamos a desarrollar brevemente un esquema de obtención de autoestados para un potencial $V_2(x)$ confinado en una región, como se muestra en la Figura 5.1. Nuestro objetivo es sortear los problemas de proyección en una base, presentados en el capítulo 4.

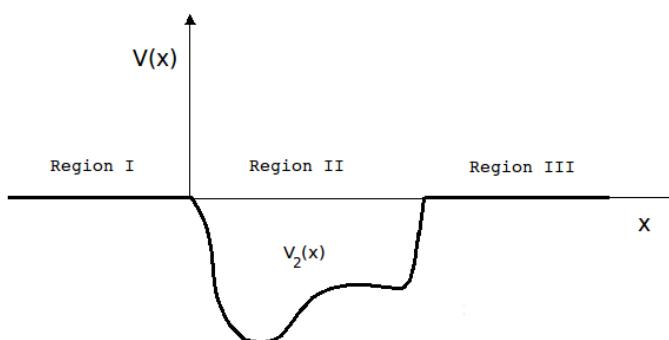


Figura 5.1: Diagrama de un potencial arbitrario $V_2(x)$ confinado en la región II, entre regiones de potencial libre.

Debemos tener cuidado al elegir las condiciones de borde en el algoritmo de generación de autoestados, para evitar la imposición de bordes rígidos espúreos en los extremos de la grilla numérica, y así solucionar los problemas de proyección planteados en el capítulo 4. En el caso de un potencial cóncavo, es posible que existan autoestados ligados. Debemos tomar en cuenta esta posibilidad, y plantear un esquema distinto de obtención de estos estados, ya que no presentan las mismas condiciones de borde que los autoestados del continuo.

5.1 Generación de estados no ligados (dispersión)

Los estados que vamos a buscar son los correspondiente al problema de dispersión¹. Para una partícula incidiendo desde la izquierda sobre el potencial $V_2(x)$, la forma funcional más general de un autoestado de energía positiva está dada en la ecuación 5.1, e ilustrada en la Figura 5.2.

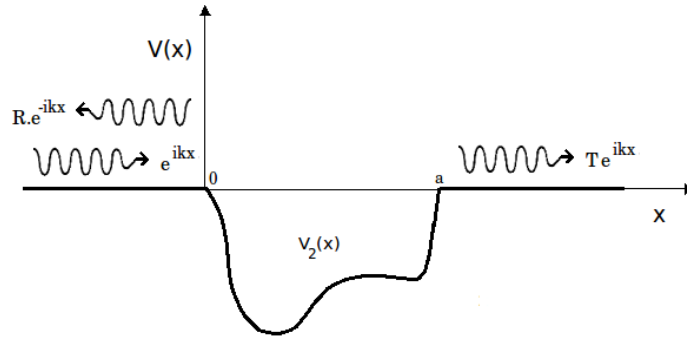


Figura 5.2: Diagrama de las condiciones de borde correspondiente al problema de Scattering desde la izquierda.

$$\varphi_k^L(x) = \begin{cases} \varphi_1(x) = e^{ikx} + R e^{-ikx} & \text{si } x \in \text{Región I} \\ \varphi_2(x) & \text{si } x \in \text{Región II} \\ \varphi_3(x) = T e^{ik(x-a)} & \text{si } x \in \text{Región III} \end{cases} \quad (5.1)$$

Por continuidad de $\varphi_k(x)$ y su derivada en los límites de la región II, las condiciones de borde resultantes para la región 2 son:

$$\begin{aligned} \varphi_1(0) = \varphi_2(0) &\rightarrow \varphi_2(0) = 1 + R \\ \varphi_1'(0) = \varphi_2'(0) &\rightarrow \varphi_2'(0) = ik(1 - R) \\ \varphi_2(a) = \varphi_3(a) &\rightarrow \varphi_2(a) = T \\ \varphi_2'(a) = \varphi_3'(a) &\rightarrow \varphi_2'(a) = ikT \end{aligned}$$

El esquema de cálculo de los autoestados es el siguiente [22]:

1. Para una energía E dada², debemos dar un valor de prueba (*guess*) inicial para R . Podemos utilizar el valor teórico correspondiente a un potencial cuadrado de las mismas dimensiones que la región 2.
2. Integramos $\varphi_2(x)$, a partir de la ecuación de Schrödinger 1.1, utilizando Runge-Kutta³ de orden 4 (ver Apéndice F) desde $x = 0$ hasta $x = a$.
3. A partir de la función $\varphi_2(x)$ obtenida por integración, calculamos el valor de T : $T = \varphi_2(a)$ o bien $T = -\frac{i}{k} \varphi_2'(a)$

1. Scattering

2. $E = \frac{1}{2}k^2$

3. En principio podríamos utilizar cualquier método de integración espacial.

4. Volvemos a integrar la función $\varphi_2(x)$ en sentido inverso (desde $x = a$ hasta $x = 0$), y calculamos un nuevo valor para R : R^{new} .
 $R^{new} = \varphi_2^{new}(0) - 1$ o bien $R^{new} = \frac{i}{k} \varphi_2'^{new}(0)$
5. Construimos la función $f(R, R^{new}) = (Re(R) - Re(R^{new}))^2 + (Im(R) - Im(R^{new}))^2$. La idea ahora es utilizar un algoritmo de búsqueda de extremos en funciones (en el apéndice E se encuentra el caso específico de *Steepest-Descent*) para minimizar f y obtener así un nuevo valor para R , repitiendo el procedimiento descrito desde el primer paso hasta lograr la precisión deseada.

Debemos repetir el proceso para distintos valores de E y para condiciones de incidencia desde la derecha, como se ilustra en la ecuación 5.2, a fin de obtener una base de proyección.

$$\varphi_k^R(x) = \begin{cases} \varphi_1(x) = T e^{-ikx} & \text{si } x \in \text{Región I} \\ \varphi_2(x) & \text{si } x \in \text{Región II} \\ \varphi_3(x) = e^{-ik(x-a)} + R e^{ik(x-a)} & \text{si } x \in \text{Región III} \end{cases} \quad (5.2)$$

En el apéndice G.3 mostramos la implementación completa del esquema en Fortran, para las condiciones 5.1.

5.2 Resultados de la proyección en la base generada

A modo de ejemplo, hemos realizado un cálculo de autoestados aplicado a un potencial tipo pozo finito. Como ya explicamos en el capítulo 4, la diagonalización numérica del Hamiltoniano arroja como resultado una serie de autoestados discretizados, de amplitud nula en los bordes de la grilla. Un ejemplo de este tipo de autoestados puede verse en la Figura 5.3.

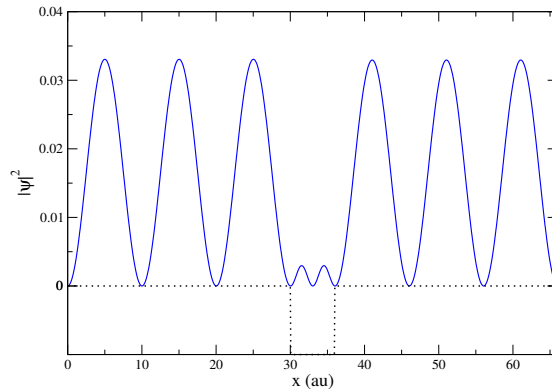


Figura 5.3: Cuarto autoestado del Hamiltoniano, obtenido por diagonalización numérica. $E_4 \approx 0,049$ au. $k_4 \approx 0,31$ au. En línea punteada mostramos el potencial tipo pozo finito (centrado en la grilla, de ancho 6 au, y profundidad 0,5 au). $\Delta x = 0,06$ au, $N = 1100$ puntos.

Por otro lado, los autoestados de la Figura 5.4 fueron generados por el procedimiento descrito en la sección anterior, para la misma energía encontrada en el autoestado de la Figura 5.3. Comparando las figuras, podemos comprobar que las condiciones de borde de los autoestados de ambas bases son efectivamente muy distintas.

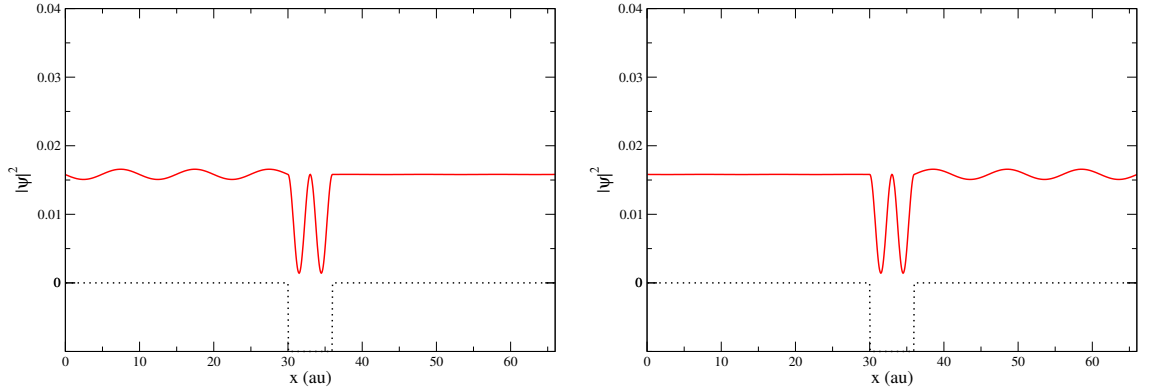


Figura 5.4: Autoestados generados a través del esquema de cálculo explicado en la sección 5.1. $E \approx 0,049$ au. $k \approx 0,31$ au. En línea punteada mostramos el potencial tipo pozo finito (centrado en la grilla, de ancho 6 au, y profundidad 0,5 au). $\Delta x = 0,06$ au, $N = 1100$ puntos.

- (a) Condiciones 5.1 (incidencia desde la izquierda).
- (b) Condiciones 5.2 (incidencia desde la derecha).

Hemos logrado obtener una base de autoestados del potencial sin confinamiento, evitando así la aparición de una simetría definida. En el caso particular de un potencial libre, el procedimiento de generación de autoestados dará como resultado ondas planas $e^{\pm ikx}$ en reemplazo de los estados estacionarios del pozo infinito (ecuación 3.4). La proyección sobre estas funciones nos permitió mostrar en el capítulo anterior la descomposición correcta de un paquete gaussiano libre: un espectro suave e invariante frente a traslaciones del paquete de ondas, como se muestra en la Figura 5.5.

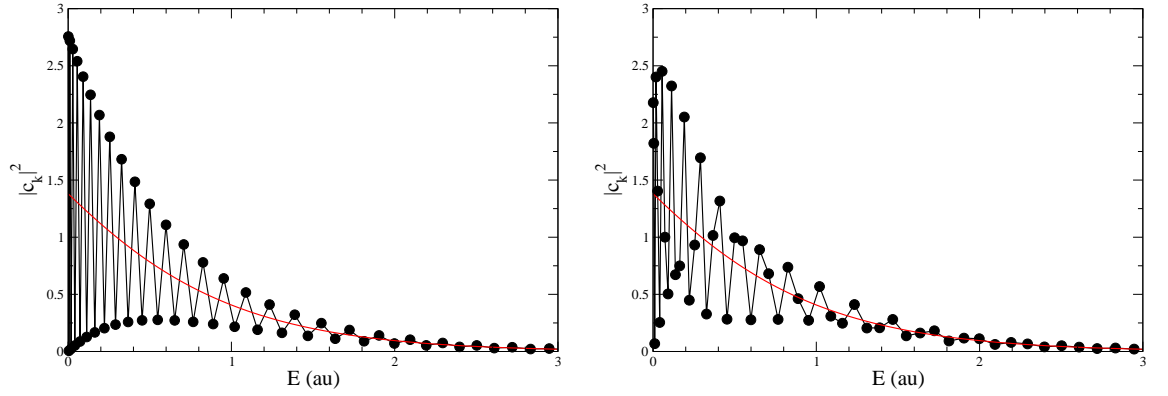


Figura 5.5: Descomposición espectral numérica del paquete de ondas Gaussiano 3.1 con $k_0 = 0,5$ au. Negro: descomposición incorrecta, proyección sobre autoestados de la caja. Rojo: descomposición correcta, proyección sobre base generada. $N = 1100$ puntos, $\Delta x = 0,06$ au. a) $x_0 = 33$ au. b) $x_0 = 23$ au.

La generación de autoestados por diagonalización numérica es una operación que puede llegar a tener un costo computacional $O(N^3)$. En los cálculos involucrados en la Figura 5.5 se obtuvieron por este procedimiento 1100 autofunciones (tantas como puntos en la grilla, con energías entre 1×10^{-3} au y 556 au), de las cuales sólo 52 aproximadamente eran significativas en la reconstrucción del paquete gaussiano original (con energía en los límites de la Figura 5.5). En este sentido, el algoritmo de generación de bases de la sección 5.1 se muestra muy conveniente, ya que nos permite construir una cantidad arbitraria de autoestados con energías de nuestra elección. El costo computacional involucrado es aproximadamente $O(N)$ por cada autoestado. Es decir que $O(N^2)$ sería el costo de obtener la cantidad $N = 1100$ de funciones generadas en la diagonalización de \hat{H} .

6

Aplicación práctica: emisión electrónica en una superficie metálica por incidencia rasante de un pulso laser ultracorto

Cuando un pulso laser incide de forma rasante sobre una superficie metálica, un electrón de la banda de valencia¹ en un estado i puede pasar a un estado f del continuo, siendo eyectado. La evolución temporal electrónica de este proceso está determinada por la ecuación de Schrödinger dependiente del tiempo (TDSE²)

$$i\frac{\partial\psi}{\partial t}(\vec{r}, t) = [H_0 + V_{laser}(z, t)]\psi(\vec{r}, t) \quad (6.1)$$

donde H_0 incluye, en el modelo Jellium³, la interacción electrón-superficie como un pozo cuadrado finito de potencial $V_{pozo} = E_{Fermi} + E_{trabajo}$. El término V_{laser} corresponde al pulso laser, de duración τ y polarizado linealmente en la dirección \hat{z} perpendicular a la superficie, cuyo perfil temporal está dado por

$$\vec{F}(t) = F_0 \sin(\omega t + \varphi) \sin^2(\pi t/\tau) \hat{z} \quad (6.2)$$

El vínculo entre ambas magnitudes está dado por $V_{laser}(z, t) = \vec{r} \cdot \vec{F}(t) = zF(t)$ en el gauge de longitud. Debemos restringir la frecuencia ω a valores altos, a fin de poder despreciar el potencial de superficie generado en respuesta al campo externo. Si bien la ecuación de Schrödinger 6.1 es tridimensional, es posible

1. En las superficies metálicas, los electrones emitidos pueden provenir de dos fuentes diferentes: las capas internas de los átomos superficiales y la banda de valencia del metal. Dado que los electrones de valencia se hallan menos ligados a la superficie, se espera que provean la contribución dominante en el proceso de ionización.

2. Time-Dependent Schrödinger-Equation

3. El modelo Jellium no contiene información sobre la estructura de bandas del sólido, pero ha probado describir adecuadamente la interacción electrón-superficie en colisiones rasantes.

obtener una forma unidimensional de la misma a partir de la invarianza en la interacción laser–superficie en la dirección paralela a la superficie, consecuencia de la incidencia rasante.

La probabilidad diferencial de emisión de electrones de la superficie puede ser expresada en términos de la matriz de transición T_{if} , que se obtiene a partir de las proyecciones entre estados ligados y continuos. En un trabajo previo [8], *Faraggi et al.* compararon exitosamente este tipo de cálculo (TDSE) con otros métodos (IJV⁴ perturbativo), para el caso de la banda de valencia del $Al(111)$. El esquema general del cálculo es similar al que desarrollamos en el capítulo 2:

- 1) Escribimos el operador Hamiltoniano, reemplazando el pozo del Jellium por un potencial suave, para evitar problemas numéricos en los bordes del pozo.
- 2) Calculamos los autoestados del pozo.
- 3) Evolucionamos temporalmente cada estado ligado del pozo, utilizando un operador Hamiltoniano que incluya al potencial del laser.
- 4) Al final de la evolución ($t > \tau$), podemos calcular los elementos T_{if} a partir de la proyección de la función de onda en los estados del continuo.

$$T_{if} = \langle \psi_i(x, t) | \varphi_f(x) \rangle$$

- 5) La probabilidad diferencial de emisión entre estados i (ligado) y f (continuo) es

$$\frac{\partial P_{if}}{\partial E_f} = \frac{|T_{if}|^2}{\Delta E_f} \quad (6.3)$$

donde el factor ΔE_f da cuenta de la densidad de estados.

Sin embargo, al momento de calcular los elementos de la matriz T_{if} , nos encontramos con los problemas de proyección mencionados en los capítulos 4 y 5. Para sortear estos problemas, de acuerdo a lo estudiado, debemos generar dos conjuntos de estados del continuo φ_k^R y φ_k^L , con condiciones de borde salientes hacia derecha e izquierda respectivamente. De esta forma, la ecuación 6.3 resulta

$$\frac{\partial P_{if}}{\partial E_f} = \frac{|T_{if}^L|^2 + |T_{if}^R|^2}{\Delta E_f}$$

con $T_{if}^L = \langle \psi_i(x, t) | \varphi_f^L(x) \rangle$ y $T_{if}^R = \langle \psi_i(x, t) | \varphi_f^R(x) \rangle$.

6.0.1 Resultados

A modo de ejemplo, vamos a realizar un cálculo TDSE sobre un potencial sencillo tipo Jellium, de profundidad $V_{\text{pozo}} = 0,5$ au y ancho $L = 1,5$ au. Hemos ideado este potencial modelo, que cuenta con un solo estado ligado de energía $E_b \simeq -0,31$ au, para evitar interferencias entre niveles y atacar específicamente los problemas de proyección. En este sentido, el cálculo del $Al(111)$ estudiado en [8] resulta más complejo porque el Jellium que lo modela contempla 108

4. *Impulsive Jellium Volkov*

estados ligados. Consideraremos un pulso laser dado por la forma funcional 6.2 con los parámetros que se muestran en la Figura 6.1.

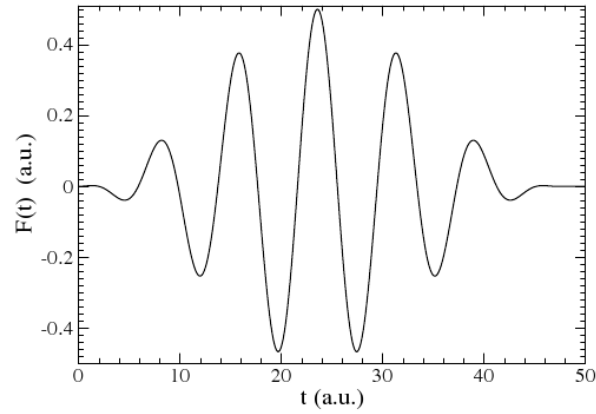


Figura 6.1: Perfil temporal del pulso laser (ecuación 6.2). $F_0 = 0,5$ $w = 0,8$ $\tau = 47,12$ au.

En la Figura 6.2 podemos observar la probabilidad diferencial de emisión electrónica obtenida para nuestro único estado ligado inicial, a partir de la interacción con el pulso laser de la Figura 6.1.

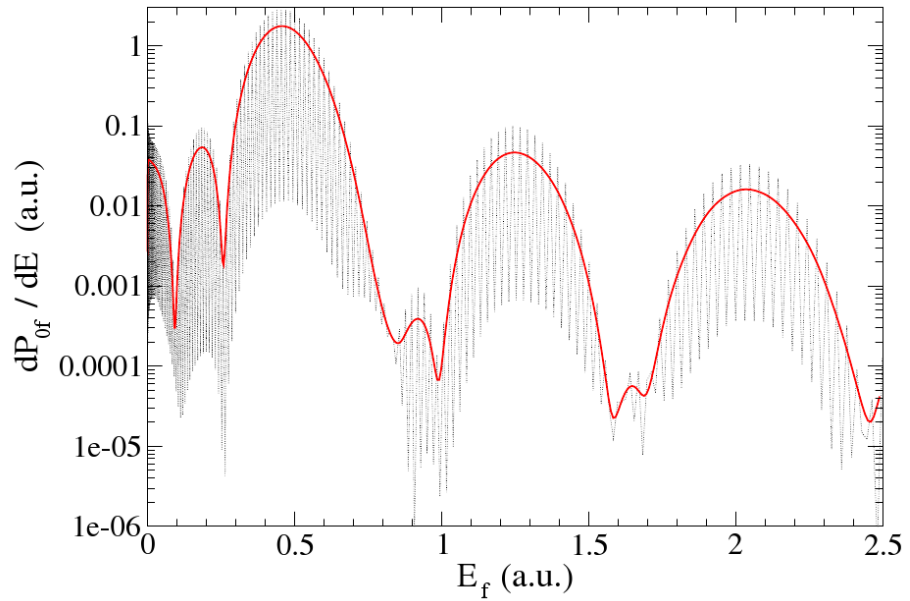


Figura 6.2: Probabilidad diferencial de emisión electrónica del pulso laser de la Figura 6.1 en interacción con una superficie metálica (modelo Jellium), en función de la energía electrónica. Línea punteada: proyección en autoestados obtenidos por diagonalización del Hamiltoniano. Línea sólida: proyección en base φ_f^R y φ_f^L generada siguiendo el procedimiento descrito en el capítulo 5.

El espectro de energía de emisión de electrones ionizados en una superficie metálica consiste generalmente de una serie de picos separados por una energía $\Delta_E = \omega$. Cada pico está vinculado a un orden distinto de ionización multifotónica [23]. En nuestro ejemplo, la diferencia de energía entre los tres picos

principales es 0,8 au, coincidente con la frecuencia ω del laser. Por otro lado, el ancho de los picos crece inversamente con la duración τ del pulso laser. Para un pulso de duración infinita, el espectro consistirá en una serie de funciones delta ubicadas en las energías de ionización multifotónica $E_m = E_b + m\omega$, donde hemos despreciado el término de energía ponderomotriz. Para pulsos cortos, es posible distinguir un número de picos secundarios entre los picos principales dado por la relación entre los períodos del laser y de su envolvente $n = \frac{T}{\tau} = \frac{2\pi/\omega}{\tau}$.

Como se vio en el capítulo 4, la presencia de grandes oscilaciones en la probabilidad diferencial de emisión electrónica obtenida es producto de la proyección de una función par (el único estado ligado del pozo) sobre estados alternativamente pares e impares. Nuevamente, la generación de una base sin paridad definida brinda una solución a los problemas de proyección, que en este caso implican oscilaciones de hasta tres órdenes de magnitud.

A lo largo de este trabajo desarrollamos e implementamos varios algoritmos relacionados con la resolución numérica de la ecuación de Schrödinger dependiente del tiempo.

- ▷ Discretización espacial del Hamiltoniano: derivada segunda centrada n -diagonal.
- ▷ Métodos de evolución temporal de la función de onda: FTCS, diagonalización, desarrollo explícito del operador de evolución temporal, Split-Operator.
- ▷ Condiciones de borde especiales: TBC, contornos periódicos.

Durante el estudio de estos esquemas surgieron problemas numéricos de alcance general, no exclusivos de la integración de esta ecuación particular. El estudio de sus orígenes y soluciones dio lugar a una publicación [7].

En el caso de paquetes de ondas de alta energía, el problema que se evidenció responde al *aliasing* espacial en la descripción de la función de onda. La consecuencia visible es una disminución en la velocidad de los paquetes de ondas por truncamiento de las componentes de mayor energía. Para evitar estos efectos debemos tener en cuenta el límite de representación de energías impuesto por el paso espacial Δx según la ecuación 4.1. Sin embargo, con excepción del Split-Operator, los métodos de evolución temporal estudiados conllevan la utilización de esquemas de diferenciación espacial, orientados a expresar numéricamente al operador Hamiltoniano, estableciendo de esta forma cotas de truncamiento en energías menores a las establecidas por la condición 4.1. A partir de la Figura 4.5 pudimos observar que en la medida en que mejoramos nuestra descripción de las derivadas espaciales (aumentando el orden de su desarrollo en serie), el truncamiento en las energías tiende a la restricción 4.1, es decir a la propia del método de Split-Operator. Este método, si bien numéricamente atractivo, presenta condiciones de borde intrínsecamente periódicas, que pueden resultar problemáticas en el cálculo sobre un potencial no periódico.

Un problema distinto se evidenció en paquetes de ondas de baja energía, con independencia de la evolución temporal. Al proyectar la función de onda en la base de autoestados del Hamiltoniano numérico (obtenidos por diagonalización)

encontramos una distribución espectral oscilatoria y dependiente de la posición del paquete de ondas, un resultado físicamente incorrecto. El problema se origina en la imposición de condiciones de contorno incorrectas para los autoestados de la base de proyección, que surge de la dimensión finita del Hamiltoniano numérico.

Para sortear el problema de proyección, hemos dedicado el capítulo 5 a desarrollar un algoritmo sencillo de generación de una base de estados correctamente condicionados en los bordes (su implementación se encuentra en el anexo G.3). De esta forma evitamos los problemas de proyección, como se pudo observar en la Figura 5.5.

Los mismos esquemas numéricos estudiados fueron utilizados en el capítulo 6 para realizar un cálculo de emisión electrónica en una superficie metálica por incidencia rasante de un pulso laser ultracorto. En éste, la sección eficaz de ionización está dada a partir de las proyecciones de la función de onda en los autoestados del potencial, el cálculo de las cuales presenta los problemas vistos en el capítulo 4. En la Figura 6.2 vimos cómo estos problemas de proyección pueden solucionarse siguiendo la receta de generación de bases del capítulo 5.

A

Unidades atómicas

Cuando un estudio es realizado dentro del área de la física atómica, es conveniente utilizar el sistema de unidades conocido como **unidades atómicas** (au) [24], dado que simplifica el manejo de factores constantes en las ecuaciones. En este sistema, los valores numéricos de las siguientes constantes físicas corresponden a la unidad. A continuación damos su equivalencia en el Sistema Internacional (SI).

- ▷ Masa del electrón $m_e = 9,109 \times 10^{-31} \text{ kg}$
- ▷ Carga del electrón $e = 1,602 \times 10^{-19} \text{ C}$
- ▷ Radio de Bohr $a_0 = 5,291 \times 10^{-11} \text{ m}$
- ▷ Constante de Planck $\hbar = \frac{h}{2\pi} = 1,054 \times 10^{-34} \text{ Js}$
- ▷ Energía de Hartree $E_h = 4,359 \times 10^{-18} \text{ J}$
- ▷ Constante de Coulomb $\frac{1}{4\pi\epsilon_0} = 8,987 \times 10^{-9} \frac{\text{Nm}^2}{\text{C}^2}$

de las cuales se derivan,

- ▷ au (tiempo) $= \frac{\hbar}{E_h} = 2,418 \times 10^{-17} \text{ s}$
- ▷ au (velocidad) $= \frac{a_0 E_h}{\hbar} = 2,187 \times 10^6 \frac{\text{m}}{\text{s}}$
- ▷ au (fuerza) $= \frac{E_h}{a_0} = 8,238 \times 10^{-8} \text{ N}$

Adicionalmente, la velocidad v el momento $p = m_e v$ y el número de onda $k = p/\hbar$ resultan numéricamente indistinguibles.

La ecuación de Schrödinger dependiente del tiempo para el proceso de dispersión¹ electrónica en un átomo con un electrón es

$$i\frac{\partial\Psi}{\partial t}(\vec{r}_1, \vec{r}_2, t) = H_{sistema}\Psi(\vec{r}_1, \vec{r}_2, t)$$

donde el Hamiltoniano no relativista del sistema está dado por

$$H_{sistema} = \sum_{i=1}^2 \left(-\frac{1}{2}\nabla_i^2 - \frac{Z}{r_i} \right) + \frac{1}{|\vec{r}_1 - \vec{r}_2|},$$

\vec{r}_1 y \vec{r}_2 son las coordenadas de ambos electrones, y Z es la carga nuclear. La función de onda electrónica puede ser expandida en armónicos esféricos acoplados para cada combinación de momentos angulares totales orbital L y de spin S :

$$\Psi^{LS}(\vec{r}_1, \vec{r}_2, t) = \sum_{l_1, l_2} \frac{P_{l_1 l_2}^{LS}(r_1, r_2, t)}{r_1 r_2} W_{l_1 l_2}^L(\hat{r}_1, \hat{r}_2)$$

$$W_{l_1 l_2}^L(\hat{r}_1, \hat{r}_2) = \sum_{m_1, m_2} C_{m_1 m_2 0}^{l_1 l_2 L} Y_{l_1 m_1}(\hat{r}_1) Y_{l_2 m_2}(\hat{r}_2)$$

donde $C_{m_1 m_2 m_3}^{l_1 l_2 l_3}$ es un coeficiente de Clebsch-Gordan, y $Y_{lm}(\hat{r})$ es un armónico esférico. Por sustitución de Ψ^{LS} en la ecuación de Schrödinger dependiente del tiempo, obtenemos el siguiente conjunto de ecuaciones diferenciales en derivadas parciales dependientes del tiempo [25]

$$i\frac{\partial P_{l_1 l_2}^{LS}}{\partial t}(r_1, r_2, t) = T_{l_1 l_2}(r_1, r_2) P_{l_1 l_2}^{LS}(r_1, r_2, t) + \sum_{l'_1, l'_2} V_{l_1 l_2 l'_1 l'_2}^L(r_1, r_2) P_{l'_1 l'_2}^{LS}(r_1, r_2, t) \quad (B.1)$$

donde

$$T_{l_1 l_2}(r_1, r_2) = \sum_{i=1}^2 \left(-\frac{1}{2} \frac{\partial^2}{\partial r_i^2} + \frac{l_i(l_i + 1)}{2r_i^2} - \frac{Z}{r_i} \right) \quad (B.2)$$

1. *Scattering*

y el operador de acoplamiento está dado por (los detalles sobre la notación de Wigner se encuentran en el apéndice C)

$$V_{l_1 l_2 l'_1 l'_2}^L(r_1, r_2) = (-1)^{L+l_2+l'_2} \sqrt{(2l_1+1)(2l'_1+1)(2l_2+1)(2l'_2+1)} \times \\ \sum_{\lambda} \frac{r_{\leq}^{\lambda}}{r_{\geq}^{\lambda+1}} \begin{pmatrix} l_1 & \lambda & l'_1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} l_2 & \lambda & l'_2 \\ 0 & 0 & 0 \end{pmatrix} \left\{ \begin{matrix} L & l'_2 & l'_1 \\ \lambda & l_1 & l_2 \end{matrix} \right\} \quad (\text{B.3})$$

Por extensión de la ecuación de Schrödinger 1.1 al caso bidimensional (x_1, x_2) , obtenemos un esquema de resolución de las ecuaciones B.1, de la misma forma funcional. En este caso, la expresión del potencial \hat{V} debe incluir los términos de acoplamiento dados por B.2 y B.3.

Las relaciones de simetría entre los coeficientes de Clebsch-Gordan se vuelve más evidente en la notación de símbolos 3j de Wigner, dada por [26]

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = \frac{(-1)^{j_1-j_2-m_3}}{\sqrt{2j_3+1}} C_{m_1 m_2 -m_3}^{j_1 j_2 j_3} \quad (\text{C.1})$$

La notación de símbolos 6j de Wigner suele aparecer cuando se considera el acoplamiento de tres estados. Una fórmula útil para calcular símbolos 6j está dada por [27]

$$\begin{aligned} \left\{ \begin{matrix} j_a & j_b & j_c \\ j_d & j_e & j_f \end{matrix} \right\} &= \Delta(j_a j_b j_c) \Delta(j_a j_e j_f) \Delta(j_d j_b j_f) \Delta(j_d j_e j_c) \times \\ &\sum_k \frac{(-1)^k (k+1)!}{(k-j_a-j_b-j_c)!(k-j_a-j_e-j_f)!} \times \\ &\frac{1}{(k-j_d-j_b-j_f)!(k-j_d-j_e-j_c)!(j_a+j_b+j_d+j_e-k)!} \times \\ &\frac{1}{(j_b+j_c+j_e+j_f-k)!(j_c+j_a+j_f+j_d-k)!} \end{aligned} \quad (\text{C.2})$$

donde

$$\Delta(j_a j_b j_c) = \sqrt{\frac{(j_a+j_b-j_c)!(j_a-j_b+j_c)!(-j_a+j_b+j_c)!}{(j_a+j_b+j_c+1)!}} \quad (\text{C.3})$$

En el desarrollo de los algoritmos de integración numérica indicamos el orden de aproximación de un cálculo en términos de los pasos de integración temporal (Δt) o espacial (Δx). En este caso, nos referimos al comportamiento asintótico del error en la medida en que el paso de integración tiende a cero (asíntota infinitesimal), y lo notamos de forma abreviada como $O(f(\Delta x))$.

La misma notación (conocida como *Ómicron de Landau*) puede ser utilizada para describir el comportamiento asintótico de una función cuando el parámetro tiende a infinito (asíntota infinita). En nuestra descripción de los algoritmos empleamos esta notación para describir el costo computacional de un cálculo (ya sea en operaciones matemáticas elementales necesarias¹, como en memoria ocupada) en la forma $O(f(N))$, donde N representa el tamaño de la grilla numérica utilizada.

1. Conocido más específicamente como *tiempo computacional*.

E

Esquema de convergencia Steepest-Descent

Los algoritmos de búsqueda de extremos en una función $g(x)$ de una variable, pueden entenderse en términos de métodos generales de búsqueda de raíces, dada la relación

$$f(x_{ext}) = \frac{dg}{dx}(x_{ext}) = 0$$

El esquema general de convergencia para estos métodos puede expresarse

$$x_{k+1} = x_k \pm \Delta x_k$$

donde x_k es el punto correspondiente a la iteración k -ésima en el esquema, y el signo depende del extremo que estamos buscando. El paso de iteración Δx_k dependerá de la función f y del método específico utilizado ¹.

Este esquema puede generalizarse al caso de funciones de más variables, como [13]

$$\vec{x}_{k+1} = \vec{x}_k + \Delta \vec{x}_k = \vec{x}_k \pm a \frac{\vec{\nabla} g(\vec{x}_k)}{|\vec{\nabla} g(\vec{x}_k)|} \quad (\text{E.1})$$

conocido como esquema Steepest-Descent, con a un parámetro pequeño positivo, ajustable para lograr convergencia.

1. Por ejemplo, para el método de Newton, $\Delta x_k = -f(x_k)/f'(x_k)$. Para el método de la secante, $\Delta x_k = -\frac{(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})} f(x_k)$

F

Método de integración de Runge-Kutta

El método de Runge-Kutta nos permitirá encontrar numéricamente la forma funcional de una función $y(x)$ que cumple una ecuación diferencial del tipo

$$\frac{dy}{dx} = g(y, x)$$

La expansión general del método es¹

$$y(x + \Delta x) = y(x) + \sum_j^m \alpha_j c_j$$
$$c_k = \Delta x \cdot g \left(y + \sum_{i=1}^{k-1} \nu_{ki} c_i; x + \Delta x \sum_{i=1}^{k-1} \nu_{ki} \right)$$

El método de Runge-Kutta de uso más común se obtiene al truncar este desarrollo en los términos $O(\Delta x^4)$. El esquema resultante es

$$y(x + \Delta x) = y(x) + \frac{1}{6}(c_1 + 2c_2 + 2c_3 + c_4)$$
$$c_1 = \Delta x \cdot g(y, x)$$
$$c_2 = \Delta x \cdot g(y + c_1/2; x + \Delta x/2)$$
$$c_3 = \Delta x \cdot g(y + c_2/2; x + \Delta x/2)$$
$$c_4 = \Delta x \cdot g(y + c_3; x + \Delta x)$$

1. Para ver un desarrollo más detallado del método consultar [22].

G.1 Split Operator (C++)

El código en C++ ilustra una implementación sencilla de evolución temporal, utilizando el Split-Operator de la ecuación 2.16. Los parámetros *transformar* y *antitransformar* constituyen planes predefinidos de la librería FFTW [28], encargada de realizar las transformadas de Fourier. La operación involucrada en estos cálculos es

$$Y_k = \sum_{n=0}^{N-1} X_n e^{-2\pi i n k / N}$$

donde los momentos positivos son alojados uniformemente ($k_n = 2\pi n / L$) en la primera mitad del array, y los negativos en orden inverso en la segunda mitad¹. Debemos tener en cuenta que, en esta implementación de FFT, la función de onda aparece multiplicada por un factor N luego de un par *Transformada-Antitransformada*.

Por razones de claridad, la implementación que mostramos no está optimizada en el sentido de efectuar sólo dos transformaciones² por paso temporal de integración, como se describe en 2.6. Una implementación óptima requeriría distinguir entre el primero paso temporal, el último, y los pasos intermedios.

1. Es decir, $k_n = -k_{N-n}$.
 2. Un par transformada-antitransformada.

```

void kinetic_evolution() {
»   fftw_complex ctmp;

»   for(int n=1;n<=N/2;n++) {
»       double k = 2.0*M_PI*n/L;
»       ctmp = (-1.0)*I*dt*power(k,2)/4.0;
»       psi[n] = psi[n]*cexp(ctmp);
»       psi[N-n] = psi[N-n]*cexp(ctmp);
»   }
}

void potential_evolution() {
»   fftw_complex ctmp;

»   for(int n=0;n<N;n++) {
»       ctmp = I*V(n*dx)*(-1.0)*dt;
»       psi[n] = psi[n]*cexp(ctmp);
»   }
}

void evolution() {
»   t = t + dt;

»   fftw_execute(transformar);
»   for(int n=0;n<N;n++) psi[n] = psi[n] * (1.0/N);
»   kinetic_evolution();
»   fftw_execute(antitransformar);
»
»   potential_evolution();
»
»   fftw_execute(transformar);
»   for(int n=0;n<N;n++) psi[n] = psi[n] * (1.0/N);
»   kinetic_evolution();
»   fftw_execute(antitransformar);
}

```

G.2 Esquema FTCS con TBC (C++)

El código en C++ ilustra una implementación sencilla del esquema FTCS de integración temporal. (ver 2.8). Dado que \hat{H} está representado por una matriz tridiagonal, podemos reducir el costo computacional de su almacenamiento, utilizando un array de dimensión $3 \times N$, y definiendo una rutina de producto matricial (apply_operator) de manera acorde.

Adicionalmente, ilustramos en este código una implementación de bordes transparentes (TBC) en el último punto de la grilla numérica ($N - 1$).

```
void create_hamiltonian() {
    >> double dx2 = dx*dx;

    >> //reseteamos el array bidimensional de H
    >> for(int i=0;i<=2;i++)
    >> >> for(int j=0;j<N;j++) H[i][j] = 0.0 + 0.0i;

    >> //construimos la matriz H salvo por los puntos extremos
    for(int i=0;i<=2;i++)
        for(int j=1;j<N;j++) {
            if (i == 1) H[i][j] = 1.0/dx2 + V(j*dx) + 0.0i;
            if ((i == 0) || (i == 2)) H[i][j] = -0.5/dx2 + 0.0i;
        }

    >> //primera fila de H
    H[0][0] = 0.0 + 0.0i;
    H[1][0] = 1.0/dx2 + V(0.0) + 0.0i;
    H[2][0] = -0.5/dx2 + 0.0i;

    >> //ultima fila de H
    H[0][N-1] = -0.5/dx2 + 0.0i;
    H[1][N-1] = 1.0/dx2 + V(dx*(N-1)) + 0.0i;
    H[2][N-1] = 0.0 + 0.0i;
}

//apply_operator devuelve en vdest el resultado del producto matricial H.v
void apply_operator(double _Complex M[3][N],
    double _Complex *v, double _Complex *vdest) {
    >> for(int j=0;j<N;j++) {
    >> >> vdest[j] = 0.0 + 0.0i;
    >> >> for(int i=0;i<=2;i++) vdest[j] = vdest[j] + M[i][j] * v[i+j-1];
    >> }
}

//calculamos kx
void TBC_hamiltonian() {
    double _Complex tmp;

    >> //calculamos kx en cada paso dt
    >> tmp = in[N-1]/in[N-2];
    >> kx = (-1.0)*I*clog(tmp)/dx;

    >> //nos aseguramos de que el flujo no sea entrante
    >> if (creal(kx) < 0) kx = 0.0 + 0.0i;
```



```

» //reescribimos el ultimo punto diagonal del hamiltoniano
» H[1][N-1] = 1.0/(dx*dx) + V(dx*(N-1)) - 0.5*cexp(I*kx*dx)/(dx*dx);
}

void propagate(ofstream *fout) {
» double _Complex taylor[Ntaylor][N];

» TBC_hamiltonian();

» //guardamos en taylor[0] la funcion de onda sin modificar
» for(int i=0;i<N;i++) taylor[0][i] = psi[i];

» //guardamos en taylor[n] el orden n del desarrollo temporal de exp(-iHdt)psi
» for(int n=0;n<Ntaylor-1;n++) {
» » apply_operator(H, taylor[n], taylor[n+1]);
» » for(int j=0;j<N;j++) taylor[n+1][j]=taylor[n+1][j]*(-1.0)*dt*I/double(n+1);
» }

» //finalmente sumamos todos los ordenes del desarrollo
» for(int n=1;n<Ntaylor;n++)
» » for(int j=0;j<N;j++) psi[j] = psi[j] + taylor[n][j];
}

```

G.3 Generación de estados continuos (Fortran 77)

```
»      !la funcion de onda y su derivada primera (array static de tamaño N)
»      complex*16 y1(N), y2(N)

»      !coeficientes A y B, de iteracion y teoricos
»      complex*16 Anew, Bnew
»      complex*16 Ateo, Bteo

»      !paso espacial y largo total de la grilla
»      real*8 dx, l

»      !numero de onda del autoestado inicial
»      real*8 rk0

»      !numero de onda del autoestado final
»      real*8 rkf

»      !paso del numero de onda
»      real*8 dk

»      !umbral de convergencia para la iteracion de steepest-descent
»      real*8 RTresol

»      !tamaño de la grilla numerica
»      integer npts

»      !cantidad de autoestados deseados
»      integer nwaves

»      !posiciones en la grilla de los extremos de la zona 2
»      integer ix1, ix2
»      real*8 x1, x2

»      !unidad imaginaria
»      zI = dcplx(0.0d0, 1.0d0)

»      !leemos de disco los parámetros iniciales
»      call readinput

»      do 10 iwave = 1, nwaves
»      »      !energia
»      »      rk = rk0 + (iwave-1)*dk
»      »      E = 0.5d0*(rk**2)
»      »      rk1 = rk

»      »      !calculamos un guess inicial Ateo para el coeficiente A.
»      »      call initbound

»      »      !integracion optimizada. usamos steepest-descent para converger
»      »      !hacia los coeficientes A y B
»      »      Anew = steepest(Ateo)

»      »      !una vez que tenemos los coeficientes A y B integramos con
»      »      !runge-kutta en toda la grilla para obtener el autoestado
»      »      y1(ix1-di) = 1.0d0 + Anew
```

```

» » y2(ix1-di) = zI*rk1*(1.0d0 - Anew)
» » call rkuttar2l(E, ix1, 1)
» » call rkuttal2r(E, ix1, npts)

» » !escribimos en disco la función obtenida
» » call writewave
10» continue
» end

» !steepest toma un guess inicial y devuelve el coeficiente
» !obtenido por convergencia
» complex*16 function steepest(Ainit)
» » a = 0.1d0
» » h = 1e-5
» » Ar = dreal(Ainit)
» » Ai = dimag(Ainit)
» » g_0 = g(Ar, Ai)

» » do 20 RTstep = 1, RTmax
» » » fx = f_1(Ar, Ai, h)
» » » fy = f_2(Ar, Ai, h)
» » » dg = dsqrt(fx**2 + fy**2)
» » » if (dg.le.RTresol) goto 30
» » » b = a/dg

» » » Ar = Ar - b*fx
» » » Ai = Ai - b*fy
» » » h = h/2.0d0
» » » g_1 = g(Ar, Ai)
» » » if (g_1.gt.g_0) a = a/2.0d0
» » » if (g_1.le.g_0) g_0 = g_1
20» » continue
30» » steepest = dcplx(Ar, Ai)
» return
» end

» !derivada parcial de g respecto a x
» real*8 function f_1(x, y, h)
» » f_1 = (g(x+dx, y)-g(x, y))/h
» return
» end

» !derivada parcial de g respecto a y
» real*8 function f_2(x, y, h)
» » f_2 = (g(x, y+dy)-g(x, y))/h
» return
» end

» !g es la funcion que debemos minimizar en el esquema steepest-descent
» real*8 function g(x, y)
» » Aold = dcplx(x, y)
» » Anew = integrate(Aold)
» » g=(dreal(Aold)-dreal(Anew))**2 + (dimag(Aold)-dimag(Anew))**2
» return
» end

» !va de x1 a x2 con el coeficiente Aold, usando runge-kutta
» !luego vuelve de x2 a x1 y calcula un nuevo coeficiente Anew
» complex*16 function integrate(Aold)

```

```

» »      !integracion outward (de x1 a x2)
» »      y1(ix1) = 1.0d0 + Aold
» »      y2(ix1) = zI*rk1*(1.0d0 - Aold)
» »      call rkuttal2r(E, ix1, ix2)
» »      Bnew = y1(ix2)

» »      !integracion inward (de x2 a x1)
» »      y2(ix2) = zI*rk1*Bnew
» »      call rkuttar2l(E, ix2, ix1)
» »      Anew = y1(ix1) - 1.0d0

» »      integrate = Anew
» return
» end

»      !integracion runge-kutta con energia E desde i0 hasta iend
»      subroutine rkuttal2r(E, i0, iend)
» »      do 100 i = i0, iend-1
» » »      x = i*dx
» » »      k11 = dx*g1(y1(i), y2(i), x, E)
» » »      k12 = dx*g2(y1(i), y2(i), x, E)
» » »      k21 = dx*g1(y1(i)+k11/two, y2(i)+k12/two, x+dx/two, E)
» » »      k22 = dx*g2(y1(i)+k11/two, y2(i)+k12/two, x+dx/two, E)
» » »      k31 = dx*g1(y1(i)+k21/two, y2(i)+k22/two, x+dx/two, E)
» » »      k32 = dx*g2(y1(i)+k21/two, y2(i)+k22/two, x+dx/two, E)
» » »      k41 = dx*g1(y1(i)+k31, y2(i)+k32, x+dx, E)
» » »      k42 = dx*g2(y1(i)+k31, y2(i)+k32, x+dx, E)
» » »      y1(i+1) = y1(i)+(k11 + two*k21 + two*k31 + k41)/6.0d0
» » »      y2(i+1) = y2(i)+(k12 + two*k22 + two*k32 + k42)/6.0d0
100» »      continue
»      return
»      end

»      !integracion runge-kutta con energia E desde iend hasta i0
»      subroutine rkuttar2l(E, i0, iend)
» »      !cambiamos derivada derecha por izquierda
» »      y2(i0) = -y2(i0)

» »      do 200 i = i0, iend+1, -1
» » »      x = i*dx
» » »      k11 = dx*g1(y1(i), y2(i), x, E)
» » »      k12 = dx*g2(y1(i), y2(i), x, E)
» » »      k21 = dx*g1(y1(i)+k11/two, y2(i)+k12/two, x-dx/two, E)
» » »      k22 = dx*g2(y1(i)+k11/two, y2(i)+k12/two, x-dx/two, E)
» » »      k31 = dx*g1(y1(i)+k21/two, y2(i)+k22/two, x-dx/two, E)
» » »      k32 = dx*g2(y1(i)+k21/two, y2(i)+k22/two, x-dx/two, E)
» » »      k41 = dx*g1(y1(i)+k31, y2(i)+k32, x-dx, E)
» » »      k42 = dx*g2(y1(i)+k31, y2(i)+k32, x-dx, E)
» » »      y1(i-1) = y1(i)+(k11 + two*k21 + two*k31 + k41)/6.0d0
» » »      y2(i-1) = y2(i)+(k12 + two*k22 + two*k32 + k42)/6.0d0
200» »      continue

» »      !cambiamos derivada izquierda por derecha
» »      do 201 i = i0, iend, -1
» » »      y2(i) = -y2(i)
201» »      continue
»      return
»      end

```

```

»      !derivada primera de y1
»      complex*16 function g1(y1, y2, x, E)
»      »      g1 = y2
»      return
»      end

»      !derivada segunda de y1
»      complex*16 function g2(y1, y2, x, E)
»      »      g2 = 2.0d0*(V(x)-E)*y1
»      return
»      end

```

Bibliografía

- [1] C. T. WHELAN. *Fragmentation processes in atomic collisions*. Science **286** (1999) 2457.
- [2] T. N. RESCIGNO *et al.* *Collisional breakup in a quantum system of three charged particles*. Science **286** (1999) 2474.
- [3] *TOP 500 Supercomputers*.
URL <http://www.top500.org/>
- [4] G. E. MOORE. *Cramming more components onto integrated circuits*. Electronics **38** (1965) 114.
- [5] J. COLGAN *et al.* *Benchmark Nonperturbative Calculations for the Electron-Impact Ionization of $Li(2s)$ and $Li(2p)$* . Phys. Rev. Lett. **87** (2001) 213201.
- [6] M. S. PINDZOLA *et al.* *The time-dependent close-coupling method for atomic and molecular collision processes*. J. Phys. B **40** (2007) R39.
- [7] A. E. GARRIZ *et al.* *Running into trouble with the time-dependent propagation of a wavepacket*. Eur. J. Phys. **31** (2010) 785.
- [8] M. N. FARAGGI *et al.* *Interaction of ultrashort laser pulses with metal surfaces: Impulsive jellium-Volkov approximation versus the solution of the time-dependent Schrödinger equation*. Phys. Rev. A **76** (2007) 012903.
- [9] M. N. FARAGGI. *Interacción de iones y pulsos electromagnéticos con superficies sólidas*. Tesis de doctorado, Universidad de Buenos Aires (2008).
- [10] A. L. CAVALIERI *et al.* *Attosecond spectroscopy in condensed matter*. Nature **449** (2007) 1029.
- [11] G. R. HADLEY. *Transparent boundary condition for the beam propagation method*. IEEE Journal of Quantum Electronics **28** (1992) 363.
- [12] E. ANDERSON *et al.* *LAPACK User's Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 3rd edition (1999).

- [13] T. PANG. *An Introduction to Computational Physics*. Cambridge University Press, 2nd edition (2006). Cap 3.
- [14] W. VAN DIJK Y F. M. TOYAMA. *Accurate numerical solutions of the time-dependent Schrödinger equation*. Phys. Rev. E **75** (2007) 036707.
- [15] J. J. SAKURAI. *Modern Quantum Mechanics*. Addison-Wesley (1994). Cap 2.
- [16] M. FRIGO Y S. G. JOHNSON. *Implementing FFTs in Practice*. C. S. Burrus. *Fast Fourier Transforms*.
- [17] N. CARJAN *et al.* *Efficient numerical solution of the time-dependent Schrödinger equation for deep tunneling*. Rom. Rep. Phys. **55** (2003) 555.
- [18] D. J. GRIFFITHS. *Introduction to Quantum Mechanics*. Prentice Hall (1995). Cap 1.
- [19] C. COHEN-TANNOUDJI *et al.* *Quantum Mechanics*. John Wiley and Sons (1973). Vol 1, Cap 1, Complemento G.
- [20] A. GOLDBERG *et al.* *Computer-Generated Motion Pictures of One-Dimensional Quantum-Mechanical Transmission and Reflection Phenomena*. Am. J. Phys. (1967).
- [21] L. I. SCHIFF. *Quantum Mechanics*. McGraw-Hill, 3rd edition (1968).
- [22] T. PANG. *An Introduction to Computational Physics*. Cambridge University Press, 2nd edition (2006). Cap 4.
- [23] F. H. M. FAISAL. *Theory of Multiphoton Processes*. Plenum, New York (1987).
- [24] H. A. BETHE Y E. E. SALPETER. *Quantum Mechanics of One- and Two- electron Atoms*. Springer-Verlag, Berlin (1957).
- [25] Y. D. WANG Y J. CALLAWAY. *Direct numerical approach to electron-hydrogen scattering. II. $L > 0$* . Phys. Rev. A **50** (1994) 2327.
- [26] W. R. JOHNSON. *Atomic Structure Theory*. Springer (2006). Cap 1.
- [27] W. R. JOHNSON. *Atomic Structure Theory*. Springer (2006). Cap 4.
- [28] M. FRIGO Y S. G. JOHNSON. *The Design and Implementation of FFTW3*. Proceedings of the IEEE **93** (2005) 216.

Publicaciones derivadas de este trabajo

1. *Running into trouble with the time-dependent propagation of a wavepacket*
A. E. Garriz, **A. Sztrajman** y D. Mitnik
Eur. J. Phys. **31** 785 (2010)
2. *Contribución resonante a la excitación de iones por impacto electrónico*
A. Sztrajman, D. Mitnik
Libro de resúmenes de la 93^o Reunión Nacional de Física, pp. 80 (2008)
3. *Problemas en la propagación temporal de la ecuación de Schrödinger*
A. E. Garriz, **A. Sztrajman** y D. Mitnik
Libro de resúmenes de la 93^o Reunión Nacional de Física, pp. 163 (2008)
4. *Métodos numéricos en física atómica: condiciones de borde transparentes*
A. Sztrajman, D. Mitnik
Libro de resúmenes de la 92^o Reunión Nacional de Física, pp. 202 (2007)

Agradecimientos

Quiero agradecer a las personas que, de forma directa o indirecta, me ayudaron a terminar este trabajo.

a Elsa y Jorge, que se bancaron de todo.

a Agus, que se ganó mi respeto.

a Laura Belluscio.

a Cristina Caputo y Ricardo Piegaia, con quienes voy a estar siempre agradecido.

a mis amigos desde siempre: Denchi, Janus, Pablo, Ine y Lk.

a Maximiliano Geier, que tomó tan buenas decisiones como las mías.

a Surak, que lo veo poco y lo quiero mucho.

al gato de Camila Cosse, y a su dueña.

a mis amigos de la facu: Nico, Pochi, Sabri, Dani, Daf, el Doc, Hernan, Chechu, Romi, Mica, Eli, Marie, Cynthia, Lucas, Mariano, Celeste, Pia, Ariel, Facu, Marián, Lucho, Andrés y Sebastián.

a Laura Ori, por ponerle tanta onda.

a María Emilia Ruiz, por la amistad y la complicidad.

a Darío Rodrigues, porque admiro que pueda vivir sin atajos.

a Iván Sidelnik, que leyó la primera version y me alentó.

a Adrián Rubio López, que me enseñó a usar una calculadora.

a Carlos Vigh, que me llevó a la tele.

a Daniel Ziella, que entendió esta tesis en 2 viajes de colectivo.

a Cinthia Ramos, por las oportunidades y la paciencia!

a Jorge Álvarez Juliá y Claudio Simeone, por la confianza.

a Oliverio Acosta, por enseñarme a dar clase.

al gran Edmundo Lavia, porque todos le debemos alguna materia.

a Matías Roncoroni, por ser la policía del estudio.

a los chicos: Vane, Yago, Marto, Fer y Ale.

al reo Lipper, un tipazo, y a Flora.

a Pepo Lapouble, personaje ilustre.

a Montaldo y los admines del DC, Fer, Damián y Andrés, por la buena onda.

a Daniel Gómez, por su ayuda.

a Jorge Miraglia y Diego Arbó, por la buena predisposición.

a Darío Mitnik, porque él también puso mucho trabajo acá.

a Liz.