

Implementación de un programa computacional para el cálculo del flujo electrónico en el H_2^+

Marlon Daniel Suárez Ruiz

Trabajo de grado para optar al título de Químico

Director

Jhon Fredy Pérez Torres

Doctor en Química Teórica

Universidad Industrial de Santander

Facultad de Ciencias

Escuela de Química

Bucaramanga

2021

Tabla de contenidos

Introducción	8
1 Justificación	9
2 Estado del Arte	9
3 Marco Teórico	12
3.1 Densidad de probabilidad y flujo electrónico	12
3.2 Aproximación de Born-Oppenheimer	13
3.3 El problema del flujo dentro de la BOA	14
3.4 Construcción del hamiltoniano completo para el H ₂ ⁺	15
3.4.1 Descripción del sistema	15
3.4.2 Separación de la componente temporal	16
3.4.3 Separación del centro de masa nuclear y movimiento relativo: $R = \vec{R}_A - \vec{R}_B $	17
3.4.4 Alineamiento en el eje Z	19
3.5 Valor esperado y Teorema de Ehrenfest	22
3.6 Método “time-shift flux density”	23
4 Objetivos	25
4.1 Objetivo general	25
4.2 Objetivos específicos	25
5 Metodología	25
5.1 Simulación de la dinámica molecular de H ₂ ⁺	26
5.2 Cálculo del orbital molecular $1s\sigma_g$ para cada tiempo t de la dinámica	28
5.3 Cálculo del flujo por el método Okuyama	29

IMPLEMENTACIÓN PROGRAMA PARA CALCULAR FLUJO EN H ₂ ⁺	3
5.4 Calibración del método	32
6 Resultados y Análisis	33
6.1 Simulación de la dinámica molecular de H ₂ ⁺	34
6.2 Cálculo del orbital molecular 1σ _g para cada tiempo t de la dinámica	36
6.3 Densidades de flujo electrónico	36
6.4 Resultados de la calibración	38
7 Conclusiones	43
Referencias bibliográficas	45
Apéndices	46

Lista de figuras

Figura	1. Diagrama inicial del sistema H ₂ ⁺	15
Figura	2. Sistema H ₂ ⁺ incluyendo centro de masas	18
Figura	3. Sistema alineado al eje Z	20
Figura	4. Diagrama de flujo correspondiente a la generación de la dinámica molecular semi-clásica	27
Figura	5. Diagrama de flujo del cálculo del orbital molecular	28
Figura	6. Diagrama de flujo para el calculo del flujo electrónico por el método Okuyama .	30
Figura	7. Diagrama de flujo del funcionamiento del programa <code>flux.exe</code>	31
Figura	8. Diagrama de flujo para la calibración del método	33
Figura	9. Valores esperados de la dinámica	35
Figura	10. Energías de las dinámicas generadas	36
Figura	11. Comparación de densidades de probabilidad para algunos tiempos	37
Figura	12. Densidades y flujos: FOE vs time-shift flux	39
Figura	13. Resultados integral I(t): δt general	40
Figura	14. Resultados integral I(t): δt valores pequeños	41
Figura	15. Flujo Okuyama / Flujo FOE. 0-50 fs	42
Figura	16. Flujo Okuyama / Flujo FOE. 0-15 fs	42

Lista de apéndices

Apéndice A	Código del script general	47
Apéndice B	Código de los programas en fortran utilizados	50
Apéndice C	Scripts de referencia para NWCHEM	60

RESUMEN

TÍTULO: Implementación de un programa computacional para el cálculo del flujo electrónico en el H₂⁺ *

AUTOR: Marlon Daniel Suárez Ruiz **

PALABRAS CLAVES: Flujo electrónico, dinámica cuántica molecular, química cuántica, ion molecular H₂⁺.

DESCRIPCIÓN:

El observable flujo electrónico o corriente electrónica aún estando presente en las publicaciones originales de Schrödinger ha tenido poco uso en la historia de la química cuántica por su problema dentro de la aproximación de Born-Oppenheimer donde da valores de cero, éste podría ser de gran utilidad para describir todos los procesos químicos en una teoría de reactividad más completa basada en la mecánica cuántica. Con la llegada del siglo XXI se han propuesto alternativas para calcularlo mientras se espera poder desarrollar tecnologías que puedan resolver la ecuación de Schrödinger para sistemas de múltiples electrones. Una de ellas, llamada time-shift flux propuesta por Okuyama y Takatsuka en el 2009 seguía sin corroborarse con los resultados matemáticamente exactos del ion molecular H₂⁺. En este proyecto se construyó un script que calcula un flujo electrónico para el mismo sistema usando el método de Okuyama a lo largo de una dinámica semi-clásica y lo comparó con los valores de referencia exactos. Se encontró una relación entre el parámetro Δt y el flujo calculado, y se da como óptimo valor $\Delta t = 2u.a.$ para generar buenos flujos electrónicos sin sacrificar demasiado los tiempos de cómputo. Por su facilidad de uso, se muestra este método como una posible herramienta para la enseñanza o estudio de diferentes procesos.

*Proyecto de investigación

**Facultad de Ciencias. Escuela de Química. Director Ph. D. Jhon Fredy Pérez Torres.

Abstract

Title: Computational programm implementation for computation of electronic flux in molecular ion H₂⁺ *

Author: Marlon Daniel Suárez Ruiz **

Key Words: electronic flux, quantum molecular dynamics, quantum chemistry, molecular ion H₂⁺.

Description: Electronic flux or electronic current as observable, despite had been in the original Shrödinger publications haven't shown no use along quantum chemistry history because its problem inside the Born-Oppenheimer aproximation where gets values of zero, this observable could be very useful to describe every process inside a whole reactivity theory based in quantum mechanics. With the arrival of the 21st century there have been proposed new alternatives to compute it while new technologies are expected to solve the Schrödinger equation for multiple electrons. One of those, called time-shift flux, proposed by Okuyama & Takatsuka in 2009 hasn't been proved with the mathematical exact results of the molecular ion H₂⁺. In this proyect a script has been built to compute electronic flux for the same system using Okuyama's method along a semi-classic dynamic and compares it with the reference exact data. It has been found a relationship between the parameter Δt and the flux calculated, and the optimal value of $\Delta t = 2$ a.u. which generates good electronic fluxes without costing too much computing time. Because its easy aplication, this method is shown as a posible tool to teach or study different processes.

*Bachelor Thesis

**Faculty of Sciences. School of Chemistry. Director Ph. D. Jhon Fredy Pérez Torres.

Introducción

En química, al describir los movimientos de electrones en la formación y rompimiento de enlaces usamos comúnmente las flechas curvas como herramienta diaria, ya sea un par de electrones si la flecha es completa (\curvearrowright), o un sólo electrón si la flecha tiene media cabeza (\rightarrow). Este símbolo aparentemente tan simple carga con un significado más profundo en mecánica cuántica. Éstos movimientos se describen con los observables **probabilidad de densidad de corriente** (EFD, por sus siglas en inglés), **corriente electrónica** o **flujo electrónico** (j_e), y están relacionados directamente con la probabilidad electrónica a través de la ecuación de continuidad. Sin embargo, esta propiedad actualmente sólo es conocida “exactamente” para sistemas monoelectrónicos a pesar de haber sido mencionada por primera vez por Schrödinger (1926d). El principal problema que se presenta al analizar esta propiedad es su desvanecimiento dentro la aproximación de Born-Oppenheimer (BOA, por sus siglas en inglés) (1927), aproximación que ha sido uno de los ejes principales de la mecánica cuántica desde su formulación. Este desvanecimiento surge del significado y desarrollo de los llamados “estados estacionarios” en los que se da por sentado que la densidad de probabilidad no cambia en el transcurso del tiempo, lo que lleva a que no se genere un flujo electrónico. Es por esto que aunque resultara evidente que el movimiento natural de la molécula generara un flujo electrónico inherente, durante el siglo XX esta propiedad fue nula siguiendo la BOA. Tampoco era viable abordar el problema desde una perspectiva netamente mecano-cuántica, por la limitación tecnológica que, aún hoy, tenemos para solucionar sistemas con múltiples electrones (problema que espera solucionarse con el desarrollo de computadoras cuánticas en las siguientes décadas). Sólo fue hasta inicios del siglo XXI que investigadores como Nafie (1997), Barth et al. (2009), Okuyama & Takatsuka (2009), Diestler (2012), desarrollaron metodologías y modelos para procesos específicos (Steiner y Fowler, 2001; Barth y Jörn Manz, 2006; Andrae y col., 2011). Sin embargo sus simulaciones aún debían ser calibradas con datos experimentales para tomar validez. Tal trabajo fue realizado con la metodología desarrollada por Diestler (2012), llamada Coupled Channels, usando para ello los valores simulados exclusivamente por la mecánica cuántica de la

molécula de H₂⁺ (Barth y col., 2009). Pero, otra de las metodologías propuestas, llamada Time-Shift Flux Density (Okuyama y Takatsuka, 2009) aún no ha sido calibrada, es por esto que este proyecto se plantea la tarea de validar esta metodología usando como referencia el ion molecular H₂⁺, de manera similar a como Diestler (2012) lo realizó para validar su teoría Coupled-Channels.

1. Justificación

Este proyecto se realiza para optar al título de químico en la Universidad Industrial de Santander (UIS), resolviendo el problema anteriormente expuesto mediante herramientas de química computacional. El tema escogido hace parte de la línea de estudio del Grupo de Investigación en Estructura Electrónica (GIEE) perteneciente al Laboratorio de espectroscopia atómica y molecular (LEAM), que requiere el aprendizaje y desarrollo de habilidades investigativas y computacionales, éstas últimas van ganando más importancia a medida que las ciencias se van adentrando inevitablemente en el campo de la Big Data, y que se han visto evolucionar enormemente a lo largo del siglo XXI. Con el desarrollo de este proyecto se podrá comprobar la viabilidad del método de Okuyama y Takatsuka para calcular la densidad de flujo electrónico como una nueva herramienta con un coste computacional más bajo y que aprovecha herramientas ya existentes y popularizadas en el área. El GIEE se fortalecerá con el desarrollo de este proyecto y además, será de gran utilidad para todos los investigadores que estén interesados en el análisis de esta propiedad mecanocuántica especialmente en moléculas sencillas, lo cual es necesario para avanzar en la consolidación de una teoría de reactividad basada en la mecánica cuántica. Los avances de este proyecto se pensaban socializar en el Encuentro Nacional de Químicos Teóricos y Computacional (ENQTC), pero, por el reciente problema de salubridad pública este evento ha sido aplazado hasta nuevo aviso*.

2. Estado del Arte

Conocer el comportamiento y estructura de la materia ha sido desde siempre un aspecto que ha cautivado la naturaleza humana que, tanto químicos y físicos han desarrollado con habilidad

*<https://sites.google.com/view/viiienqtc-vectccm/iniciohome> [consultado 01/Jul/2020]

a lo largo de los años. Sin esa curiosidad no se hubiese desarrollado ni la mecánica clásica ni la mecánica cuántica, las cuales, juntas, pueden describir la mayoría de sistemas que hacen parte de la materia y todo lo que llamamos realidad. Aunque con el paso de los años químicos y físicos divergieran sus enfoques de estudio, todos estos aportes quedaron para el desarrollo de nuevas áreas, como las que nos compete ahora, la dinámica cuántica molecular.

A través de los años hemos aprendido que los enlaces químicos son covalencia o transferencia de electrones, y la formación o ruptura de éstos involucra una dinámica conjunta entre núcleos y electrones que puede ocurrir de manera casi instantánea. Para observar experimentalmente estos cambios se ha hecho enormes esfuerzos en desarrollar técnicas que puedan seguir su evolución en el tiempo como espectroscopia láser, difracción de rayos x o bombardeo electrónico (Bratos y col., 2004). Sin embargo, aunque en las últimas dos décadas se han mejorado estas técnicas de manera que se ha empezado a alcanzar escalas de femto y attosegundos, su aplicabilidad sigue en desarrollo (Hermann y col., 2020). Para comprender mejor los enlaces se usa como alternativa las simulaciones computacionales, sin embargo, si se quieren obtener resultados precisos como lo dicta la teoría, hay que resolver la ecuación de Schrödinger y para sistemas multielectrónicos sigue siendo un reto que se espera cumplir con el inminente desarrollo de computadoras cuánticas.

Actualmente la teoría más aceptada para describir los átomos y moléculas en química es la mecánica cuántica. Ésta, durante todo el siglo XX, luego de su postulación, ha permitido describir y predecir el comportamiento de protones, neutrones, electrones, núcleos atómicos, átomos y moléculas, así como sus respectivas energías entre otros observables mecano-cuánticos. Cabe resaltar que con la llegada del nuevo milenio y los desarrollos tecnológicos en computación se han hecho avances importantes en una rama de estudio relativamente nueva dentro de la mecánica cuántica: dinámica cuántica molecular. Ésta se plantea el estudio de procesos, llámense vibraciones, transiciones de estado, reacciones químicas en el tiempo, usando para ello observables bien conocidos de mecánica cuántica como densidad de probabilidad (ρ), momento (p), energía y otros no tan conocidos como densidad de corriente electrónica o densidad de flujo electrónico (EFD, por sus siglas en inglés) y flujo electrónico (las cuales se discutirán más adelante). Éstas últimas a pesar de

haber sido descritas en los artículos originales de Schrödinger (1926d) durante todo el siglo XX recibieron, poca o nula atención, debido a, irónicamente la que fue causante de la mayoría de avances en mecánica cuántica de estados estacionarios: la aproximación de Born-Oppenheimer (BOA). Es justamente la densidad de flujo electrónico (EFD) objeto de estudio de este proyecto, que describe el movimiento electrónico dentro de un sistema de una manera más sensible que la densidad de probabilidad, pero que sigue siendo muy elusiva al momento de calcularla, obteniendo valores de 0 dentro de la BOA.

Como punto de partida se tiene como referencia el ión molecular H_2^+ , uno de los sistemas que actualmente se pueden resolver de manera precisa, del cuál se hizo un tratamiento netamente mecano-cuántico (Barth y col., 2009) que al día de hoy se ha establecido como sistema estándar ya que su solución se considera “matemáticamente exacta”.

Pero, siendo la BOA un elemento clave dentro de la teoría cuántica, y que aporta la mayoría de observables necesarios para el estudio de la dinámica cuántica molecular, se han desarrollado metodologías que aprovechan sus beneficios para calcular esa propiedad tan elusiva, la EFD o el flujo electrónico.

Uno de los primeros métodos que utiliza la BOA para abordar el problema, publicado por Nafie (1997), usa la teoría de perturbaciones sobre una superposición de dos estados para obtener una corriente electrónica de la transición entre éstos estados.

Por otro lado, Dennis Diestler publicó un par de artículos en los que fundamenta (Diestler, 2012) y prueba (Diestler y col., 2012) su teoría llamada “Coupled-Channels” para calcular el flujo electrónico y nuclear conjuntamente dentro de la BOA.

Unos años antes, Okuyama y Takatsuka presentaron su “time-shift flux density” (2009) como una alternativa para calcular el flujo electrónico (ver 3.6), mostrando una comparación topológica como prueba de su validez, pero no una comparación cuantitativa como sí lo hizo Diestler (2012). Esto último debido a que a la fecha de su publicación no se habían publicado los resultados de Barth et al. (2009) sobre el sistema H_2^+ .

Más recientemente Diestler y col. (2018) publicaron un estudio bastante completo de la dinámica

del H₂⁺ con métodos aproximativos, donde se puede observar que actualmente calcular la EFD y el flujo electrónico sigue siendo un problema a resolver.

3. Marco Teórico

Uno de los cambios de paradigmas más importantes del siglo XX fue el desarrollo de la mecánica cuántica (Levine, 2001), postulada por Erwing Schrödinger (1926a; 1926b; 1926c; 1926d) y complementada por otros grandes científicos como Werner Heisenberg, Paul Dirac, Wolfgang Pauli; desde entonces la mecánica cuántica cambió el transcurso del desarrollo de la química y física.

3.1. Densidad de probabilidad y flujo electrónico

La mecánica cuántica se fundamenta en la ecuación de Schrödinger,

$$i\hbar \frac{\partial}{\partial t} \psi(\mathbf{r}, t) = \left[-\frac{\hbar^2}{2m} \nabla^2 + V(\mathbf{r}, t) \right] \psi(\mathbf{r}, t) \quad (1)$$

Cuya solución plantea la existencia de una función de onda Ψ , la cual describe el comportamiento de una partícula (protones, neutrones, núcleos o electrones) o sistema de partículas (átomos, moléculas) y con la que se puede obtener todas las características físicas del sistema. A diferencia de la mecánica clásica, en la mecánica cuántica el equivalente a la posición clásica, \mathbf{r} , es la densidad de probabilidad de posición, o simplemente densidad de probabilidad, ρ (Levine, 2001) definida como

$$\rho(\mathbf{r}, t) = \psi^*(\mathbf{r}, t) \psi(\mathbf{r}, t) = |\psi(\mathbf{r}, t)|^2 \quad (2)$$

y que hace referencia a la probabilidad de encontrar la partícula en una región del espacio respecto a un punto \mathbf{r} en un tiempo t , por ejemplo, dónde es más probable encontrar el electrón. La densidad de probabilidad ρ se normaliza integrando sobre todo el espacio

$$\int |\psi(\mathbf{r}, t)|^2 d^3r = 1 \quad (3)$$

Sin duda alguna, la densidad de probabilidad es la representación más común de los electrones en la química moderna, y es usada para evaluar las transiciones electrónicas comparando las gráficas de estas funciones en diferentes estados (Nafie, 1997). Sin embargo, existe otra función que puede darnos más información sobre los cambios en la densidad de probabilidad, la densidad de probabilidad de corriente electrónica, densidad de corriente o densidad de flujo electrónico \mathbf{j}_e (Schiff, 1986, pág. 26), que se define como

$$\mathbf{j}_e(\mathbf{r}, t) = \frac{\hbar}{2im} [\psi^*(\mathbf{r}, t) \nabla \psi(\mathbf{r}, t) - (\nabla \psi^*(\mathbf{r}, t)) \psi(\mathbf{r}, t)] \quad (4)$$

La cantidad $\mathbf{j}_e(\mathbf{r}, t)$ corresponde a un campo vectorial con magnitud y dirección definidas en cada punto del espacio de la función de onda y describe el “flujo espacial”[†] de la probabilidad electrónica como resultado del cambio en el tiempo. Si tenemos en cuenta que la probabilidad no se puede perder o ganar en un sistema cerrado, como lo es una molécula estable, la densidad de probabilidad es una cantidad que se conserva y obedece la ecuación de continuidad, relacionando así la densidad de probabilidad y la densidad de corriente:

$$\frac{\partial}{\partial t} \rho(\mathbf{r}, t) + \nabla \cdot \mathbf{j}(\mathbf{r}, t) = 0 \quad (5)$$

3.2. Aproximación de Born-Oppenheimer

La aproximación de Born-Oppenheimer (BOA por sus siglas en inglés) en términos simples consiste en la separación de la dependencia de los movimientos electrónicos y nuclear para facilitar la resolución de la ecuación de Schrödinger (ver 1). Ésta se fundamenta en que, los electrones, al ser muchísimo más ligeros que los núcleos (la masa del electrón es aproximadamente 1800 veces menor que la del protón) se acoplan “instantáneamente” al movimiento de estos últimos, de forma que se pueda calcular las energías electrónicas para posiciones nucleares fijas. Esto se representa cómo

$$\psi(\mathbf{r}, t) = \phi(\mathbf{r}; \mathbf{R}) \chi(t) \quad (6)$$

[†]no hay que olvidar que la magnitud que fluye es la probabilidad y no los electrones como tal

donde ϕ corresponde a la función de onda electrónica, χ a la función de onda nuclear, \mathbf{r} es el vector posición de los electrones, \mathbf{R} es la distancia internuclear relativa a la posición de los núcleos, y t es tiempo. El hecho de que \mathbf{R} esté luego de un punto y coma significa que la función ϕ depende paramétricamente de ese valor, que en este caso viene proporcionado por la función nuclear χ .

Aunque esta una suposición que pueda considerarse leve, normalmente se discute poco sobre ésta y se acepta casi dogmáticamente, tiene consecuencias profundas. Es ésta aproximación sobre la que se basan la mayoría de avances en química cuántica (Cramer, 2004, pág. 111).

3.3. El problema del flujo dentro de la BOA

Aunque el flujo electrónico fue descrito por primera vez en las publicaciones originales de Schrödinger (1926d), poca utilidad se le ha dado debido a que dentro de la BOA, toma valores de cero. Esto se puede demostrar fácilmente tomando la ecuación (4) y (6) de la siguiente forma

$$\mathbf{j}_e(\mathbf{r}, t) = \frac{\hbar}{2im} [\phi(\mathbf{r})\chi^*(t)\nabla\phi(\mathbf{r})\chi(t) - (\nabla\phi(\mathbf{r})\chi^*(t))\phi(\mathbf{r})\chi(t)] \quad (7)$$

Teniendo en cuenta que los estados estacionarios, son por lo tanto reales se llega a

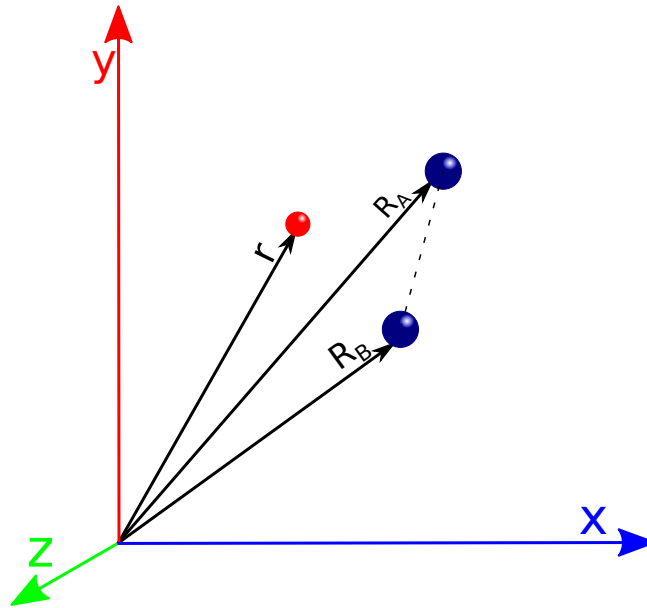
$$\mathbf{j}_e(\mathbf{r}, t) = \frac{\hbar}{2im} [\phi(\mathbf{r})\nabla\phi(\mathbf{r}) - (\nabla\phi(\mathbf{r}))\phi(\mathbf{r})]\chi^*(t)\chi(t) = 0 \quad (8)$$

Este problema fue una gran limitante para definir un flujo electrónico en reacciones que ocurren en un mismo punto desde un enfoque de superficie de energía potencial (PES, por sus siglas en inglés), y por esto mismo durante casi todo el siglo XX las corrientes electrónicas se estudiaban únicamente con campos electromagnéticos externos o pulsos láser que inducían un cambio de estado (Okuyama y Takatsuka, 2009). A pesar de esto, con la llegada de siglo XXI y todos los avances computacionales que trajo consigo, algunos investigadores de este campo desarrollaron algunos métodos alternativos para desvelar esta propiedad tan elusiva, algunos dentro de la BOA con todas las ventajas que tiene, y otros renunciado a ella.

3.4. Construcción del hamiltoniano completo para el H_2^+

3.4.1 Descripción del sistema El sistema H_2^+ está compuesto de 2 protones, los núcleos de hidrógeno, y un único electrón. Podemos representar las posiciones de estas partículas en el espacio por sus vectores posición en un sistema cartesiano de 3 dimensiones, tal como se muestra en la figura 1, de modo que \vec{R}_A da la posición del protón A, \vec{R}_B da la posición del protón B, y \vec{r} da la posición del electrón. Adicionalmente, cada protón tiene su respectiva masa, $M_A = M_B = m_p$.

Figura 1. Diagrama inicial del sistema H_2^+ .



El estado de este sistema va a estar descrito por la función de estado $\tilde{\Psi}(\vec{R}_A, \vec{R}_B, \vec{r}, t)$, de forma que la ecuación de Schrödinger dependiente del tiempo correspondiente toma la siguiente forma:

$$i\hbar \frac{\partial}{\partial t} \tilde{\Psi}(\vec{R}_A, \vec{R}_B, \vec{r}, t) = \hat{H} \tilde{\Psi}(\vec{R}_A, \vec{R}_B, \vec{r}, t)$$

donde el Hamiltoniano es:

$$\hat{H} = \frac{P_A^2}{2M_A} + \frac{P_B^2}{2M_B} + \frac{P_e^2}{2m_e} + \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\vec{R}_A - \vec{R}_B|} - \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\vec{r} - \vec{R}_A|} - \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\vec{r} - \vec{R}_B|}$$

siendo

$$P_A = \frac{\hbar}{i} \nabla_A, \quad P_B = \frac{\hbar}{i} \nabla_B, \quad P_e = \frac{\hbar}{i} \nabla_e$$

e : carga elemental

ε_0 : permitividad del vacío

El Hamiltoniano anterior (\hat{H}) consta de las componentes de energía cinética de todas las partículas, y sus interacciones coulómbicas, primero entre núcleos, luego entre el electrón con cada uno de ellos.

3.4.2 Separación de la componente temporal Nótese que \hat{H} no depende del tiempo, por tanto se puede separar la función temporal tal como:

$$\tilde{\Psi}(\vec{R}_A, \vec{R}_B, \vec{r}, t) = \Psi(\vec{R}_A, \vec{R}_B, \vec{r})\Theta(t)$$

Reemplazando $\tilde{\Psi}$ en la ecuación de Shrödinger obtenemos

$$i\hbar\Psi\frac{\partial}{\partial t}\Theta(t) = (\hat{H}\Psi)\Theta(t)$$

$$\hat{H}\Psi = E\Psi$$

Dado que el hamiltoniano no depende de t, entonces

$$i\hbar\Psi\frac{\partial\Theta(t)}{\partial t} = E\Psi\Theta(t)$$

$$i\hbar d\Theta(t) = E\Theta(t)dt$$

$$\int \frac{d\Theta(t)}{\Theta(t)} = -\frac{iE}{\hbar} \int dt$$

$$\ln(\Theta(t)) + c_1 = -\frac{iE}{\hbar}t + c_2$$

definiendo:

$$c = c_2 - c_1$$

$$\ln(\Theta(t)) = -\frac{iE}{\hbar}t + c$$

$$\Theta(t) = e^{-iEt/\hbar+c} = A_0 e^{-iEt/\hbar}$$

Finalmente, la función de estado toma la forma

$$\tilde{\Psi}(\vec{R}_A, \vec{R}_B, \vec{r}, t) = \Psi(\vec{R}_A, \vec{R}_B, \vec{r}) A_0 e^{-iEt/\hbar}$$

Ahora debemos calcular $\Psi(\vec{R}_A, \vec{R}_B, \vec{r})$ y E

3.4.3 Separación del centro de masa nuclear y movimiento relativo: $R = |\vec{R}_A - \vec{R}_B|$ Si definimos dos nuevas cantidades vectoriales para poder realizar una separación del centro de masa de nuestro sistema, tal como

$$\vec{R} = \vec{R}_A - \vec{R}_B, \quad \vec{R}_{CM} = \frac{1}{M}(M_A \vec{R}_A + M_B \vec{R}_B)$$

con $M = M_A + M_B$, suponiendo que $m_e \ll m_p$. Usando la regla de la cadena para derivadas, obtenemos

$$\nabla_A = \nabla_R \left(\frac{d\vec{R}}{d\vec{R}_A} \right) + \nabla_{CM} \left(\frac{d\vec{R}_{CM}}{d\vec{R}_A} \right)$$

$$\nabla_A = \nabla_R + \frac{M_A}{M} \nabla_{CM}$$

$$\nabla_B = \nabla_R \left(\frac{d\vec{R}}{d\vec{R}_B} \right) + \nabla_{CM} \left(\frac{d\vec{R}_{CM}}{d\vec{R}_B} \right)$$

$$\nabla_B = -\nabla_R + \frac{M_B}{M} \nabla_{CM}$$

Con estas dos igualdades para ∇_A y ∇_B transformamos $\frac{P_A^2}{2M_A}$ y $\frac{P_B^2}{2M_B}$

$$\begin{aligned} \frac{P_A^2}{2M_A} + \frac{P_B^2}{2M_B} &= \frac{1}{2M_A} \left(\frac{\hbar}{i} \nabla_A \right) \left(\frac{\hbar}{i} \nabla_A \right) + \frac{1}{2M_B} \left(\frac{\hbar}{i} \nabla_B \right) \left(\frac{\hbar}{i} \nabla_B \right) \\ &= \frac{-\hbar^2}{2M_A} \left(\nabla_R + \frac{M_A}{M} \nabla_{CM} \right) \left(\nabla_R + \frac{M_A}{M} \nabla_{CM} \right) - \frac{\hbar^2}{2M_B} \left(-\nabla_R + \frac{M_B}{M} \nabla_{CM} \right) \left(-\nabla_R + \frac{M_B}{M} \nabla_{CM} \right) \\ &= \frac{-\hbar^2}{2M_A} \left[\nabla_R^2 + \frac{M_A}{M} (\nabla_R \nabla_{CM} + \nabla_{CM} \nabla_R) + \frac{M_A^2}{M^2} \nabla_{CM}^2 \right] - \frac{\hbar^2}{2M_B} \left[\nabla_R^2 - \frac{M_B}{M} (\nabla_R \nabla_{CM} + \nabla_{CM} \nabla_R) + \frac{M_B^2}{M^2} \nabla_{CM}^2 \right] \end{aligned}$$

Simplificando $\frac{1}{M} (\nabla_R \nabla_{CM} + \nabla_{CM} \nabla_R)$ tenemos

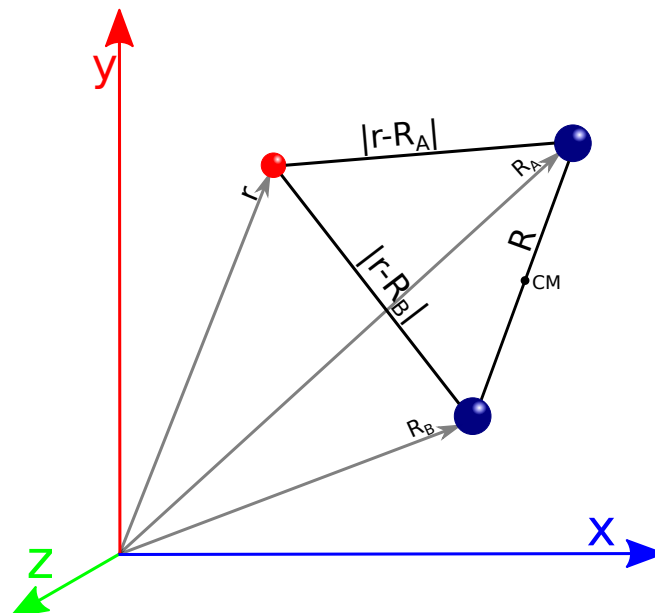
$$\frac{P_A^2}{2M_A} + \frac{P_B^2}{2M_B} = \frac{-\hbar^2}{2} \left(\frac{1}{M_A} + \frac{1}{M_B} \right) \nabla_R^2 - \frac{\hbar^2}{2} \left(\frac{M_A}{M^2} + \frac{M_B}{M^2} \right) \nabla_{CM}^2$$

Definiendo $\frac{1}{M} = \frac{1}{M_A} + \frac{1}{M_B}$, donde $M = \frac{M_A \cdot M_B}{(M_A + M_B)}$ es la masa nuclear reducida, obtenemos

$$\hat{H} = \frac{-\hbar^2}{2M} \nabla_R^2 - \frac{\hbar^2}{2M} \nabla_{CM}^2 - \frac{\nabla_e^2}{2m_e} - \frac{e^2}{4\pi\epsilon_0} \frac{1}{R} - \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\vec{r} - \vec{R}_A|} - \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\vec{r} - \vec{R}_B|}$$

El anterior hamiltoniano describe el mismo sistema de la figura 1 con el cambio de variables tal como se muestra en la siguiente figura 2.

Figura 2. Sistema H_2^+ incluyendo centro de masas.



Nuevamente, al aplicar separación y cambio de variables se puede separar una función que describe netamente el centro de masas nuclear

$$\Psi(\vec{R}_A, \vec{R}_B, \vec{r}) = \Psi(R, R_{CM}, \vec{r})$$

$$\Psi(\vec{R}_A, \vec{R}_B, \vec{r}) = \Psi(R, \vec{r})\gamma(R_{CM})$$

Obtenemos, a partir del hamiltoniano anterior para el centro de masas

$$\left[-\frac{\nabla_{CM}^2}{2M} \right] \gamma(R_{CM}) = E_{CM} \gamma(R_{CM})$$

Donde $\gamma(R_{CM})$ representa el comportamiento de onda viajera del sistema, E_{CM} es la energía cinética de todo el sistema, y se considera el sistema como una partícula/molécula libre, es decir en ausencia de fuerzas externas. De esta forma, el hamiltoniano para el momento interno de la molécula/sistema

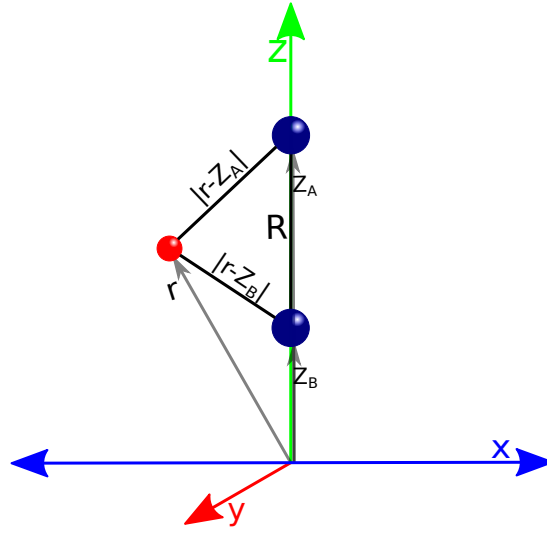
$$\left[\frac{-\hbar^2}{2M} \nabla_R^2 - \frac{\hbar^2}{2m_e} \nabla_r^2 + \frac{e^2}{4\pi\epsilon_0} \frac{1}{R} - \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\vec{r} - \vec{R}_A|} - \frac{e^2}{4\pi\epsilon_0} \frac{1}{|\vec{r} - \vec{R}_B|} \right] \Psi(\vec{R}, \vec{r}) = E \Psi(\vec{R}, \vec{r})$$

3.4.4 Alineamiento en el eje Z Ahora, suponiendo que los núcleos se muevan a lo largo del eje Z, como se muestra en la fig. 3 y la molécula NO rota, tenemos por lo tanto sólo vibración molecular en el eje Z.

Se puede mostrar que $Z_A = \frac{M_A}{M} R$ y $Z_B = \frac{-M_B}{M} R$ siendo $R = Z_A - Z_B$, además, si $M_A = M_B$, entonces $Z_A = \frac{R}{2}$ y $Z_B = \frac{-R}{2}$ de forma que nuestra ecuación de autovalores $\hat{H}\Psi = E\Psi$ se transforma en:

$$\left[\frac{-\hbar^2}{2M} \frac{d^2}{dR^2} - \frac{\hbar^2}{2m_e} \nabla_r^2 + \frac{e^2}{4\pi\epsilon_0} \frac{1}{R} - \frac{e^2}{4\pi\epsilon_0} \frac{1}{r_A} - \frac{e^2}{4\pi\epsilon_0} \frac{1}{r_B} \right] \Psi(R, \vec{r}) = E \Psi(R, \vec{r})$$

donde hicimos $r_A = |\vec{r} - \vec{Z}_A|$ y $r_B = |\vec{r} - \vec{Z}_B|$. Hay que tener en cuenta que la ecuación anterior NO es separable en R y \vec{r} , debido a la definición que hicimos anteriormente de r_A y r_B , pero veamos

Figura 3. Sistema alineado al eje Z.

que pasaría si proponemos una separación como en la ecuación 3.2

$$\Psi(R, \vec{r}) = \chi(R)\phi(\vec{r}; R)$$

Tenemos entonces

$$\left[\frac{-\hbar^2}{2M} \frac{d^2}{dR^2} - \frac{\hbar^2}{2m_e} \nabla_r^2 + \frac{e^2}{4\pi\epsilon_0} \frac{1}{R} - \frac{e^2}{4\pi\epsilon_0} \frac{1}{r_A} - \frac{e^2}{4\pi\epsilon_0} \frac{1}{r_B} \right] \chi(R)\phi(\vec{r}; R) = E\chi(R)\phi(\vec{r}; R)$$

$$\left[\frac{-\hbar^2}{2M} \frac{d^2}{dR^2} \chi(R)\phi(\vec{r}; R) - \chi(R) \frac{\hbar^2}{2m_e} \nabla_R^2 \phi(\vec{r}; R) - \frac{e^2}{4\pi\epsilon_0} \frac{1}{r_A} \chi(R)\phi(\vec{r}; R) - \frac{e^2}{4\pi\epsilon_0} \frac{1}{r_B} \chi(R)\phi(\vec{r}; R) + \frac{e^2}{4\pi\epsilon_0} \frac{1}{R} \chi(R)\phi(\vec{r}; R) \right] = E\chi(R)\phi(\vec{r}; R)$$

Reordenando:

$$\left[\frac{-\hbar^2}{2M} \frac{d^2}{dR^2} \chi(R)\phi(\vec{r}; R) + \chi(R) \left(\frac{-\hbar^2}{2m_e} \nabla_r^2 \phi(\vec{r}; R) - \frac{e^2}{4\pi\epsilon_0} \frac{1}{r_A} \phi(\vec{r}; R) - \frac{e^2}{4\pi\epsilon_0} \frac{1}{r_B} \phi(\vec{r}; R) + \frac{e^2}{4\pi\epsilon_0} \frac{1}{R} \phi(\vec{r}; R) \right) \right] = E\chi(R)\phi(\vec{r}; R)$$

El término entre paréntesis se puede calcular por separado para cada valor de R, dado que ϕ depende paramétricamente de R, es decir, fijamos R y calculamos ϕ , cambiamos el valor de R y recalculamos otro ϕ obteniendo al final un conjunto de ϕ : una ϕ para cada R. Esto se conoce como aproximación adiabática o aproximación de núcleos fijos

Nuestra ecuación de autovalores nuevamente toma la forma

$$\hat{H}_{el}\phi(\vec{r}; R) = V(R)\phi(\vec{r}; R)$$

donde $V(R)$ es la curva de energía potencial, y el \hat{H}_{el} es

$$\hat{H}_{el} = \frac{-\hbar^2}{2m_e} \nabla_r^2 - \frac{e^2}{4\pi\epsilon_0} \frac{1}{r_A} - \frac{e^2}{4\pi\epsilon_0} \frac{1}{r_B} + \frac{e^2}{4\pi\epsilon_0} \frac{1}{R}$$

De esta manera nuestra ecuación de Shrödinger

$$\left[\frac{-\hbar^2}{2M} \frac{d^2}{dR^2} (\chi(R)\phi(\vec{r}; R)) + \chi(R)\hat{H}_{el}\phi(\vec{r}; R) \right] = E\chi(r)\phi(\vec{r}; R)$$

Usando la aproximación adiabática $\hat{H}_{el}\phi(\vec{r}; R) = V(R)\phi(\vec{r}; R)$, tenemos que:

$$\begin{aligned} \left[\frac{-\hbar^2}{2M} \frac{d^2}{dR^2} (\chi(R)\phi(\vec{r}; R)) + \chi(R)V(R)\phi(\vec{r}; R) \right] &= E\chi(r)\phi(\vec{r}; R) \\ \left[\frac{-\hbar^2}{2M} \frac{d}{dR} \frac{d}{dR} (\chi(R)\phi(\vec{r}; R)) + \chi(R)V(R)\phi(\vec{r}; R) \right] &= E\chi(r)\phi(\vec{r}; R) \\ \left[\frac{-\hbar^2}{2M} \frac{d}{dR} \left(\chi(R) \frac{d\phi(\vec{r}; R)}{dR} + \phi(\vec{r}; R) \frac{d\chi(R)}{dR} \right) + \chi(R)V(R)\phi(\vec{r}; R) \right] &= E\chi(r)\phi(\vec{r}; R) \\ \left[\frac{-\hbar^2}{2M} \left(\frac{d\chi(R)}{dR} \frac{d\phi(\vec{r}; R)}{dR} + \chi(R) \frac{d^2\phi(\vec{r}; R)}{dR^2} + \frac{d\phi(\vec{r}; R)}{dR} \frac{d\chi(R)}{dR} + \phi(\vec{r}; R) \frac{d^2\chi(R)}{dR^2} \right) + \chi(R)V(R)\phi(\vec{r}; R) \right] &= E\chi(r)\phi(\vec{r}; R) \end{aligned}$$

Los términos $d\phi(\vec{r}; R)/dR$ y $d^2\phi(\vec{r}; R)/dR^2$ se llaman acoplamientos NO adiabáticos y son los que producen intercambio de momento ó energía entre los núcleos y electrones cuando se mueven. Si hacemos $d\phi(\vec{r}; R)/dR = 0$ y $d^2\phi(\vec{r}; R)/dR^2 = 0$ (Aproximación de Born-Oppenheimer) conseguimos

$$\left[\frac{-\hbar^2}{2M} \phi(\vec{r}; R) \frac{d^2\chi(R)}{dR^2} + \chi(R)V(R)\phi(\vec{r}; R) \right] = E\chi(r)\phi(\vec{r}; R)$$

Al multiplicar cada lado de la ecuación por $1/\phi(\vec{r}; R)$ obtenemos una ecuación nuclear

$$\left[\frac{-\hbar^2}{2M} \frac{d^2\chi(R)}{dR^2} + \chi(R)V(R) \right] = E\chi(r)$$

acompañada de una ecuación electrónica

$$\hat{H}_{el}\phi(\vec{r}; R) = V(R)\phi(\vec{r}; R)$$

Hemos separado el movimiento nuclear del movimiento electrónico y esto es válido si $d\phi(\vec{r};R)/dR = 0$ y $d^2\phi(\vec{r};R)/dR^2 = 0$, y ocurre en ciertos casos (la mayoría de casos) dado que $m_e \ll m_p$.

Finalmente, la función de estado total que hemos estado separando toma la forma de

$$\tilde{\Psi}(\vec{R}_A, \vec{R}_B, \vec{r}, t) = \gamma(R_{CM})\chi(R)\phi(\vec{r}; R)A_0e^{-iEt/\hbar}$$

donde, para vibraciones NO electrónicas

$$\chi(R)A_0e^{-iEt/\hbar} = \sum_n \tilde{\chi}_n(R)A_0e^{-iE_n t/\hbar} \cdot c_n = \chi(R, t)$$

Es decir, se hace una combinación lineal de las autofunciones de \hat{H}_{nuc} , $\chi_n(R)$ con energías E_n , y obtenemos un paquete de ondas vibracionales que se propagan en el estado basal electrónico $\phi(\vec{r}; R)$ de energía $V(R)$.

3.5. Valor esperado y Teorema de Ehrenfest

Debido a que $\rho(\mathbf{r}, t)$ es sólo la probabilidad de encontrar la partícula, resulta útil definir un “valor esperado” del correspondiente vector posición para la partícula. Este valor esperado no es más que la expectativa matemática para el resultado de una medición, o el promedio de los resultados de un gran número de mediciones, el valor más probable, y se describe como (Levine, 2001):

$$\langle \mathbf{r} \rangle = \int \psi^*(\mathbf{r}, t) \mathbf{r} \psi(\mathbf{r}, t) d^3r = \int \mathbf{r} \rho(\mathbf{r}, t) d^3r \quad (9)$$

El valor esperado de cualquier otra cantidad física significativa se puede hallar de manera similar si son funciones que dependen únicamente de las coordenadas \mathbf{r} de la partícula. Tal es el caso de momento lineal

$$\langle \mathbf{p} \rangle = \int \psi^*(\mathbf{r}, t) \frac{\hbar}{i} \nabla \psi(\mathbf{r}, t) d^3r \quad (10)$$

o la energía potencial

$$\langle V \rangle = \int \psi^*(\mathbf{r}, t) V(r) \psi(\mathbf{r}, t) d^3r = \int V(r) \rho(\mathbf{r}, t) d^3r \quad (11)$$

Como bien es sabido, la mecánica clásica termina por convertirse en una aproximación de la mecánica cuántica, de esta forma, si tomamos los valores esperados y observamos su comportamiento en el tiempo queda claro que concuerdan con las trayectorias clásicas de su propiedad

$$m \frac{d}{dt} \langle \mathbf{r} \rangle = \frac{d}{dt} \langle \mathbf{p} \rangle \quad (12)$$

De esta forma se puede trabajar con los valores esperados de una manera “semi-clásica”. Este arreglo se conoce como Teorema de Ehrenfest (Ehrenfest, 1927)

3.6. Método “time-shift flux density”

Este método, propuesto por Okuyama y Takatsuka (2009) al que llamaron “time-shift flux” es el objeto de estudio de este proyecto. Su propuesta se basa en considerar la dinámica como una integral de superposición de una función de onda propagándose en el tiempo, en lugar de la derivada temporal de la densidad electrónica (ver lado izquierdo de 1). De manera que al definir $\Psi(\vec{r}, t)$ dentro de un volumen Ω a partir de la ecuación de Schrödinger (ver 1) se obtiene,

$$\begin{aligned} i\hbar \frac{\partial}{\partial t} \int_{\Omega} \psi^*(\mathbf{r}, t - \Delta t/2) \psi(\mathbf{r}, t + \Delta t/2) d\mathbf{r} \\ = -\frac{\hbar^2}{2m} \int_{\Omega} d\mathbf{r} [\psi^*(t - \Delta t/2) \nabla^2 \psi(t + \Delta t/2) - \psi(t + \Delta t/2) \nabla^2 \psi^*(t - \Delta t/2)] \\ + \int_{\Omega} d\mathbf{r} \psi^*(t - \Delta t/2) (V(t + \Delta t/2) - V^*(t - \Delta t/2)) \psi(t + \Delta t/2) \end{aligned}$$

Si no se aplican campos externos, la dependencia del potencial $V(\mathbf{r}, t)$ surge únicamente del movimiento de los núcleos, y se reescribe como $V(\mathbf{r}, \mathbf{R}(t))$, y para valores Δt suficientemente pequeños, donde $(V(\mathbf{r}, \mathbf{R}(t + \Delta t/2)) - V^*(\mathbf{r}, \mathbf{R}(t - \Delta t/2)))$ se hace despreciable, resulta en

$$\begin{aligned} & \frac{\partial}{\partial t} \int_{\Omega} \psi^*(t - \Delta t/2) \psi(t + \Delta t/2) d\mathbf{r} \\ &= -\frac{\hbar}{2mi} \int_{\partial\Omega} d\mathbf{A} \nabla \cdot [\psi^*(t - \Delta t/2) \nabla \psi(t + \Delta t/2) - \psi(t + \Delta t/2) \nabla \psi^*(t - \Delta t/2)] \end{aligned}$$

donde $\partial\Omega$ es la superficie del volumen Ω , coordenadas en las cuales se expanden por \mathbf{A} . De la anterior ecuación se puede definir flujo complejo $\vec{J}(t, \Delta t)$ de manera similar a como se realiza para la definición formal de flujo (Schiff, 1986):

$$\vec{J}(t, \Delta t) = -\frac{\hbar}{2mi} [\psi^*(t - \Delta t/2) \nabla \psi(t + \Delta t/2) - \psi(t + \Delta t/2) \nabla \psi^*(t - \Delta t/2)] \quad (13)$$

Es ésta función de flujo, vectorial, la que denominan “time-shift flux”, representa la pérdida de superposición de la función de onda entre dos tiempos $t - \Delta t/2$ y $t + \Delta t/2$, o visto de otra manera, la dirección espacial de la deformación de la función de onda de $\psi(t - \Delta t/2)$ a $\psi(t + \Delta t/2)$. Tal como indican los autores, mediante el seguimiento de $\vec{J}(t, \Delta t)$ con una selección apropiada de Δt se podría comprender el comportamiento del flujo generado a partir de una función de onda $\Psi(t)$.

En cuanto a la aplicación del método, los autores proponen el uso de una dinámica semi-clásica utilizando el Teorema de Ehrenfest (ver 3.5) y relacionarla con densidades de probabilidad generadas dentro de la BOA para calcular el flujo. Sin embargo, en su publicación, aunque sugieren un rango de valores, no se define qué valor preciso de Δt debería considerarse en la practica ni se detallan datos sobre su comportamiento. Es por esto que se plantea como objetivo final la determinación del valor óptimo de Δt que produce que este “time-shift flux”, al que llamaremos flujo de Okuyama, tenga el comportamiento más cercano a los resultados de la función de onda exacta (FOE), la cuál actualmente sólo se conoce para el ion molecular H₂⁺.

4. Objetivos

4.1. Objetivo general

Desarrollar un programa computacional que permita calcular la densidad de flujo electrónica a partir de los resultados de dinámica molecular que proporciona el software de química cuántica NWCHEM.

4.2. Objetivos específicos

- ✓ Calcular el orbital molecular $1s\sigma_g$ del H₂⁺ en función del tiempo tras la ionización vertical del H₂ simulada con el NWCHEM.
- ✓ Implementar un programa computacional que permita calcular el flujo electrónico del H₂⁺ empleando el formalismo de Okuyama y Takatsuka a partir del orbital molecular $1s\sigma_g$.
- ✓ Determinar el valor óptimo del intervalo de tiempo Δt que surge en formalismo de Okuyama y Takatsuka comparando el flujo obtenido con el flujo exacto.

5. Metodología

Los objetivos anteriormente mencionados se cumplieron mediante las siguientes actividades:

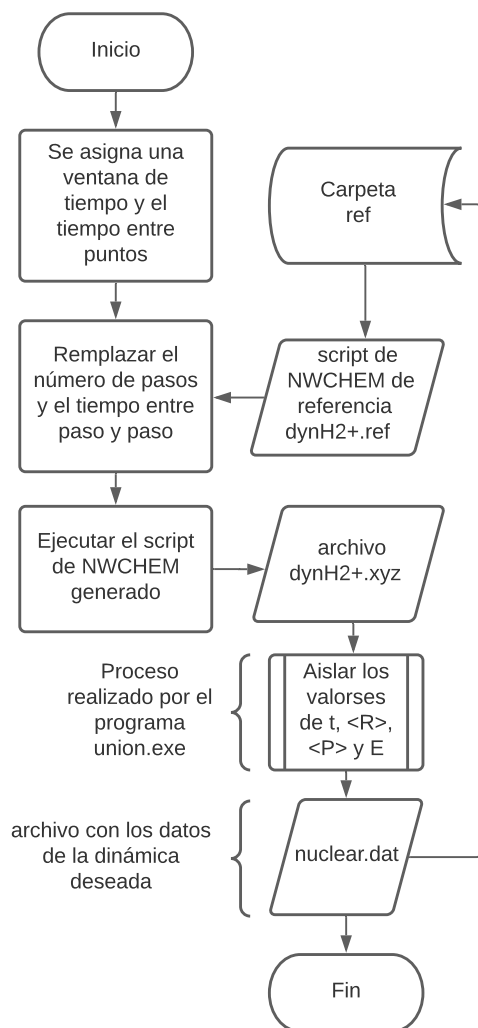
- ✓ Simulación una dinámica molecular semi-clásica para el ión molecular H₂⁺.
- ✓ Cálculo del orbital molecular $1s\sigma_g$ del ión molecular para cada tiempo t de la dinámica.
- ✓ Cálculo de la densidad de flujo $j_e^{\text{Okuyama}}(\mathbf{r}, t)$ a partir del orbital molecular para cada tiempo t de la dinámica.
- ✓ Búsqueda de condiciones en las que la densidad de flujo calculada por el método de Okuyama, $j_e^{\text{Okuyama}}(\mathbf{r}, t)$, mejor concuerdan con la densidad de flujo $j_e^{\text{exacta}}(\mathbf{r}, t)$ obtenida de la función de onda exacta.

Los procedimientos estándar se realizaron en el software de química computacional de alto rendimiento de código abierto NWCHEM, desarrollado por el Environmental Molecular Sciences Laboratory (EMSL) del Pacific Northwest National Laboratory (PNNL) de Washington, Estados Unidos. Este software ya estaba disponible en el GIEE, y por ser de código abierto es de fácil distribución. Las actividades anteriormente descritas se explicaran a continuación con ayuda de diagramas de flujo para dejar en claro el funcionamiento del código construido que se encuentra como anexo al final del documento.

5.1. Simulación de la dinámica molecular de H₂⁺

Tal como se puede observar en la fig. 4, la simulación de la dinámica empieza estableciendo unos parámetros: la ventana de tiempo a estudiar que corresponde a la duración que le daremos a la dinámica, y el tiempo entre los puntos que se generarán en la dinámica. Este par de parámetros son importantes porque luego (ver 5.4) serán unas de las condiciones a verificar en el último paso. En el código estos parámetros se declaran como `timegap` y `dtime`.

Declarados estos parámetros, se procede a reemplazar dichos valores en un script de referencia de NWCHEM llamado `dynH2+.ref` almacenado previamente en la carpeta **ref** que especifica la geometría de la molécula, base a utilizar y tareas a realizar. La base utilizada para esta simulación de la dinámica fue `aug-cc-pvtz`, el resto del contenido del script de referencia se muestra en los anexos. Esta carpeta almacena varios archivos de referencia que se mencionarán más adelante y también recibirá algunos archivos de datos importantes. El script generado, `dynH2+.nw` es usado por el módulo Gaussian Basis AIMD (dinámica molecular *ab initio* con bases gaussianas) de NWCHEM para simular la dinámica nuclear usando el algoritmo de Verlet y el potencial electrónico calculado con un método de bases gaussianas especificado (SCF en nuestro caso) y luego imprime en un archivo `.xyz` coordenadas, velocidades y energía total del sistema. Para el caso del ión molecular H₂⁺, el método SCF es más que suficiente debido a su simplicidad (2 protones y 1 electrón). El archivo generado se llamó `dynH2+.xyz` tal como se observa en la fig 4. El módulo Gaussian Basis AIMD de NWCHEM permite modificar otras opciones configurables que se detallan en su documentación.

Figura 4. Diagrama de flujo correspondiente a la generación de la dinámica molecular semi-clásica.

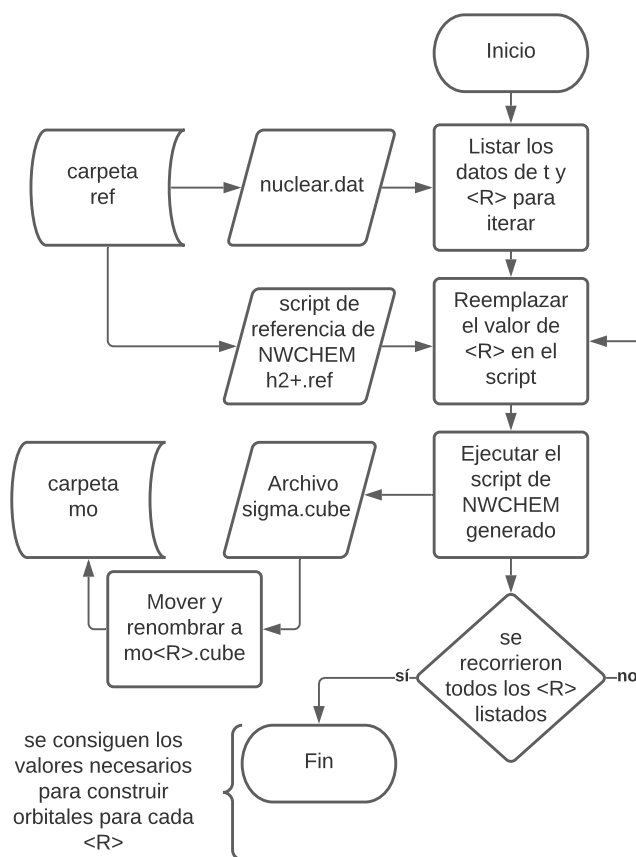
En nuestro caso no fue necesario el uso de dichas opciones adicionales, puesto que se obtuvieron resultados de gran concordancia con las opciones por defecto.

Ahora, aunque ya generamos los valores de la dinámica, siguiendo la fig 4, todavía falta organizar dichos valores puesto que el formato en el archivo `dynH2+.xyz` no es el más amigable y adecuado para nuestro interés, por lo que mediante el uso de un programa en `fortran` que llamamos `union.exe` se logra reorganizar dichos valores en un formato mucho más simple y fácil de entender en un archivo llamado `nuclear.dat`. Este archivo consta de 4 columnas, cada una para el tiempo (t), el valor esperado de la posición ($\langle R \rangle$), el valor esperado del momento ($\langle P \rangle$) y la energía del

sistema. Una vez este archivo se almacene en la carpeta **ref** se puede dar por hecha la actividad de generar la dinámica molecular molecular semi-clásica para el ión molecular H₂⁺, la cual se puede comparar fácilmente con los datos de la función de onda exacta(FOE) de manera gráfica (véase 6).

5.2. Cálculo del orbital molecular $1s\sigma_g$ para cada tiempo t de la dinámica

Figura 5. Diagrama de flujo del cálculo del orbital molecular.



NWCHEM también permite generar las funciones de onda de diferentes moléculas especificando geometría, bases a utilizar y el método para calcular la energía electrónica, en este caso se usó un nivel de teoría SCF y la base aug-cc-pvtz. Para generar estas funciones se requiere entonces suministrar los valores esperados de la posición ($\langle R \rangle$) obtenidos en cada tiempo. Es por esto que el diagrama de flujo de la fig. 5 empieza listando dichos valores que se han almacenado en el archivo `nuclear.dat`. Así mismo, de la misma manera que en la subsección anterior, se toma un script de

referencia de NWCHEM, llamado `h2+.ref`, en el que se reemplaza el valor de $\langle R \rangle$ correspondiente. Este script se encuentra en los anexos. De esta forma NWCHEM nos devuelve un archivo llamado `sigma.cube` que contiene los valores de la función de onda que representa el orbital molecular del sistema simulado, éste se mueve y renombra con el valor de $\langle R \rangle$ correspondiente y se almacena en una carpeta llamada **mo**. Esta serie de pasos se repite a lo largo de toda la trayectoria reemplazando cada uno de los valores de $\langle R \rangle$ de la dinámica anteriormente generada. Son este conjunto de archivos `.cube` los que representarán el comportamiento de los orbitales moleculares del ión molecular H₂⁺ a lo largo de la dinámica y que serán necesarios para la siguiente etapa.

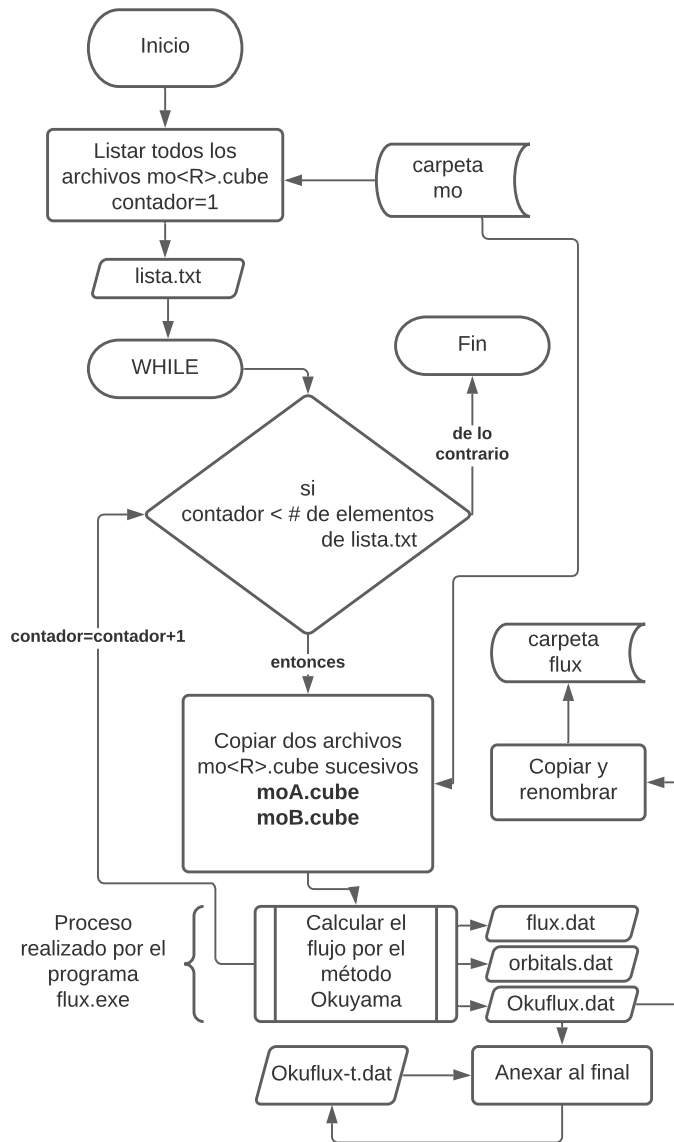
5.3. Cálculo del flujo por el método Okuyama

El cálculo del flujo por el método Okuyama (véase 3.6) lo realiza un programa propio en fortran llamado `flux.exe`. Este programa lee 2 archivos `.cube` sucesivos que previamente se copiaron al directorio de trabajo principal y se renombraron como `moA.cube` y `moB.cube`. Por tal motivo, el diagrama de flujo de la fig. 6 empieza listando todos los archivos presentes en la carpeta **mo** y definiendo un contador que será utilizado en el bucle `while`.

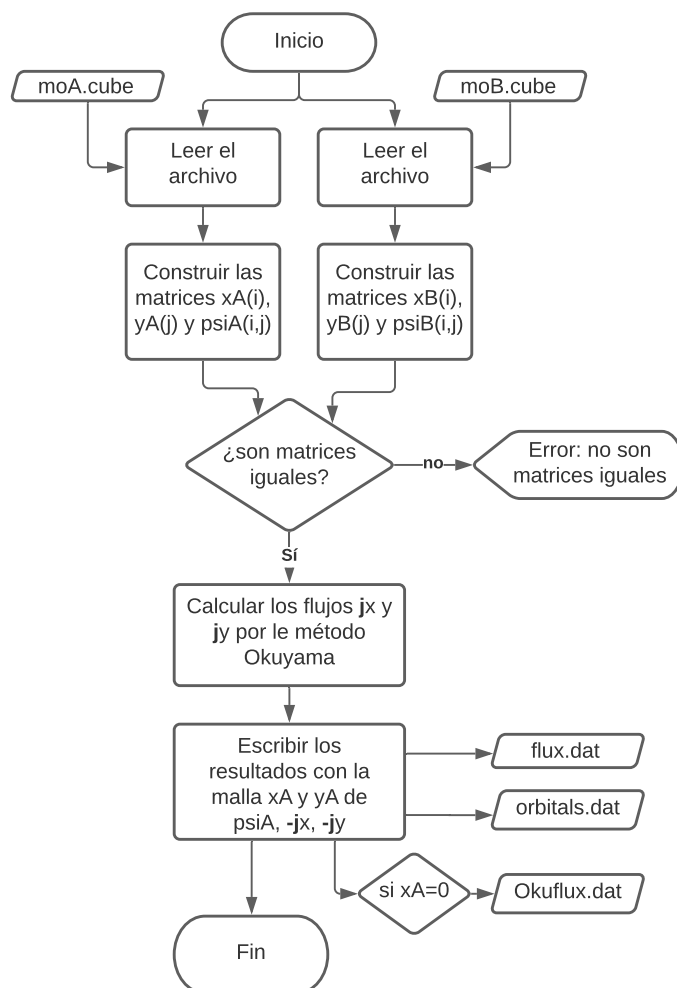
Los archivos elegidos de la carpeta **mo** a utilizar se definen según la lista generada de sus contenidos, asegurándose de esta forma que sean 2 archivos sucesivos en la trayectoria. Esto se repite hasta el penúltimo elemento de la lista, ya que siempre se toma el elemento que señala el contador en la lista y su elemento siguiente, de esta manera se asegura calcular los flujos correspondientes para todos los archivos `mo.cube` posibles.

El funcionamiento del programa `flux.exe` se describe en la fig. 7.

El programa `flux.exe` empieza leyendo los archivos `moA.cube` y `moB.cube` del directorio de trabajo y construyendo las matrices $X(i)$, $Y(j)$ y $\psi(i,j)$ tanto para A como para B. Las matrices X y Y conforman lo que se denomina malla, que dicta la posición en las coordenadas x,y en la que se sitúa el valor de la función de onda ψ . Inmediatamente después de haber leído los archivos A y B se hace una comprobación de igualdad de tamaño de ambas mallas que en dado caso que detecte que no son del mismo tamaño alerta con un error. Una vez hecho esto, el programa aplica la eq. 13

Figura 6. Diagrama de flujo para el calculo del flujo electrónico por el método Okuyama.

donde las matrices ψ_A y ψ_B toman el lugar de $\psi(t + \Delta t/2)$ y $\psi^*(t - \Delta t/2)$ respectivamente. Los cálculos anteriores se realizan por separado para cada componente y se escriben posteriormente en varios archivos según se necesite. Tanto la fig. 6 como la fig. 7 muestran que se escriben 3 archivos diferentes. El primer archivo que se muestra es `flux.dat`, este archivo contiene todos los valores del flujo calculado en toda la malla x,y . El archivo `orbitals.dat` contiene los valores de la función de onda de A en la malla x,y . Por último, el archivo `Okuflux.dat` contiene los valores del flujo

Figura 7. Diagrama de flujo del funcionamiento del programa *flux.exe*.

correspondientes a la sección del eje intermolecular, es decir, cuando $x=0$. Hay que aclarar que estos archivos se generan en cada paso del bucle, por lo que si se quiere obtener una descripción general de toda la trayectoria se tiene que almacenar en cada paso. Tal situación se muestra en la fig. 7 para el archivo *Okuflux.dat*, en el que sus datos son añadidos en cada paso, al final de un archivo mucho más general llamado *Okuflux-t.dat* que describiría el comportamiento del flujo durante toda la dinámica. Además de esto, en cada paso se guardó una copia del archivo *Okuflux.dat* en una carpeta llamada **flux**.

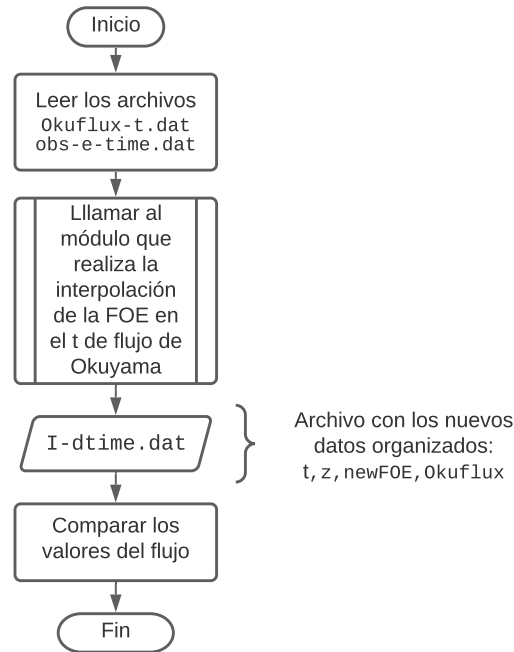
5.4. Calibración del método

Cumplidos los pasos anteriores, queda el objetivo más importante del proyecto: encontrar las condiciones en las que el método de Okuyama y Takatsuka concuerde mejor con el método de la función de onda exacta. Ahora, dado que la densidad de flujo es mayor al lo largo del eje intermolecular es decir, cuando $x = 0$, son éstos los datos de mayor interés, es por esto que en la figura 6 y 7 se muestra el archivo `Okuflux-t.dat` con especial detalle.

El primer inconveniente a resolver en este paso es la diferencia en la malla de tiempo entre los datos de la FOE y los datos del flujo generado a partir de la dinámica simulada y la aproximación de Okuyama. Aunque ambos métodos tratan la misma molécula, la cuestión aquí es que el tiempo entre paso y paso en el que se está simulando la dinámica va a variar según cambien los valores de Δt mientras que los valores de la FOE tienen un intervalo fijo. Lo más acertado a realizar sería generar un nuevo conjunto de valores según la FOE para hacer una comparación equivalente, pero, el proceso para generar un conjunto de valores según la FOE toma aproximadamente 1 semana de tiempo de cómputo cuando menos, y ciertamente esta aproximación busca reducir significativamente los tiempos que demoran dichos cálculos, además de que generar un nuevo conjunto de datos con un intervalo de tiempo mayor malogra la precisión que se tiene con intervalos cortos. Es por esto que la solución que se encontró para mejor es realizar una interpolación de los datos de la FOE con un intervalo de tiempo pequeño (mucho más precisos) sobre la malla temporal que se va generando en cada dinámica, de esta forma se obtienen excelentes valores de densidad de flujo independientemente de la malla de tiempo que se utilice. Para esto se usa un módulo que realiza dicha interpolación basado en bsplines (Carl de Boor, 1978)

Lo anterior se resuelve en un programa en `fortran` al que se llamó `diff.exe` que se explica en la figura 8. Dicho programa lee los valores de los archivos `Okuflux-t.dat` y `obs-e-time.dat`, luego llama al módulo `inter_mod.f90` que realiza la interpolación de los datos de la FOE (`obs-e-time.dat`) en la malla de los datos de `Okuflux-t.dat` y guarda en un archivo llamado `I-dtime.dat` dichos valores. De esta forma se pueden comparar fácilmente ambos flujos.

La optimización del parámetro Δt que interviene en la aproximación de Okuyama se evaluará

Figura 8. Diagrama de flujo para la calibración del método.

minimizando la siguiente integral para cada tiempo t ,

$$I(t) = \min_{\Delta t \in \mathbb{R}} \left\{ \int_{-\infty}^{\infty} \left[j_{e,z}^{\text{exacta}}(z, t) - j_{e,z}^{\text{Okuyama}}(z, t; \Delta t) \right]^2 dz \right\} \quad (14)$$

Para encontrar los Δt que producen magnitudes de flujo similares a los flujos generados por la FOE, el programa `diff.f` utiliza la función `maxval()` sobre los valores de flujo de la FOE y los generados por el método de Okuyama de manera que se pueda obtener un valor representativo para los valores del flujo, de manera análoga a lo que son los valores esperados de otros observables, y así poder escoger un valor de Δt para poder minimizar la anterior integral.

6. Resultados y Análisis

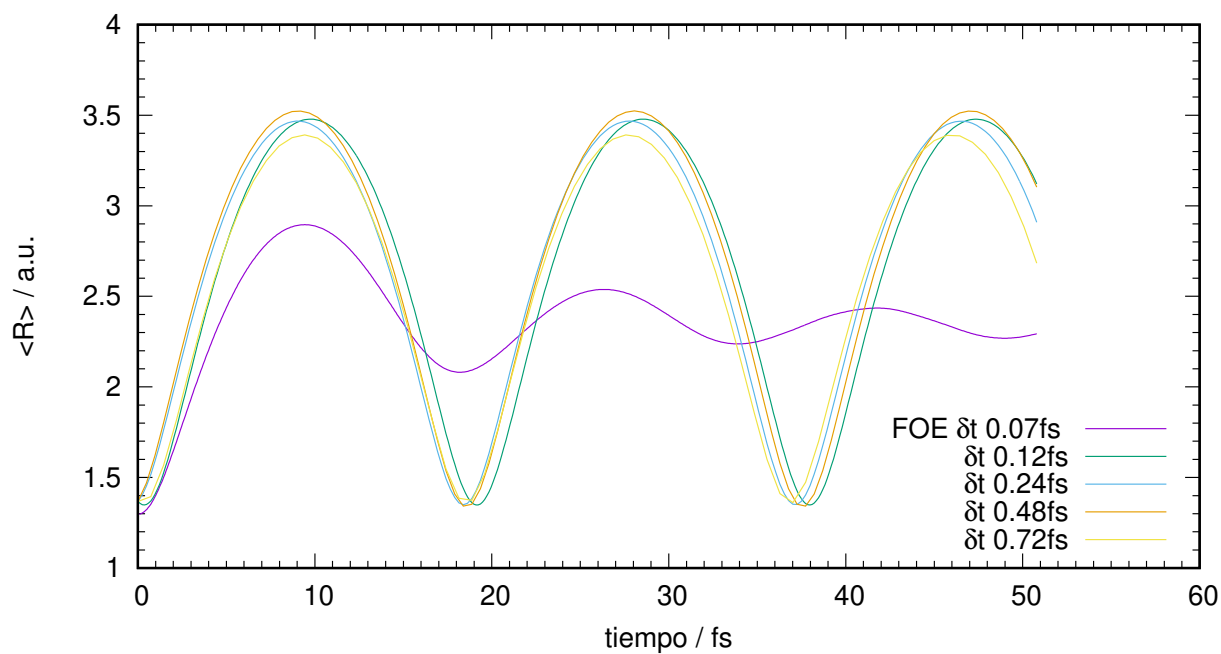
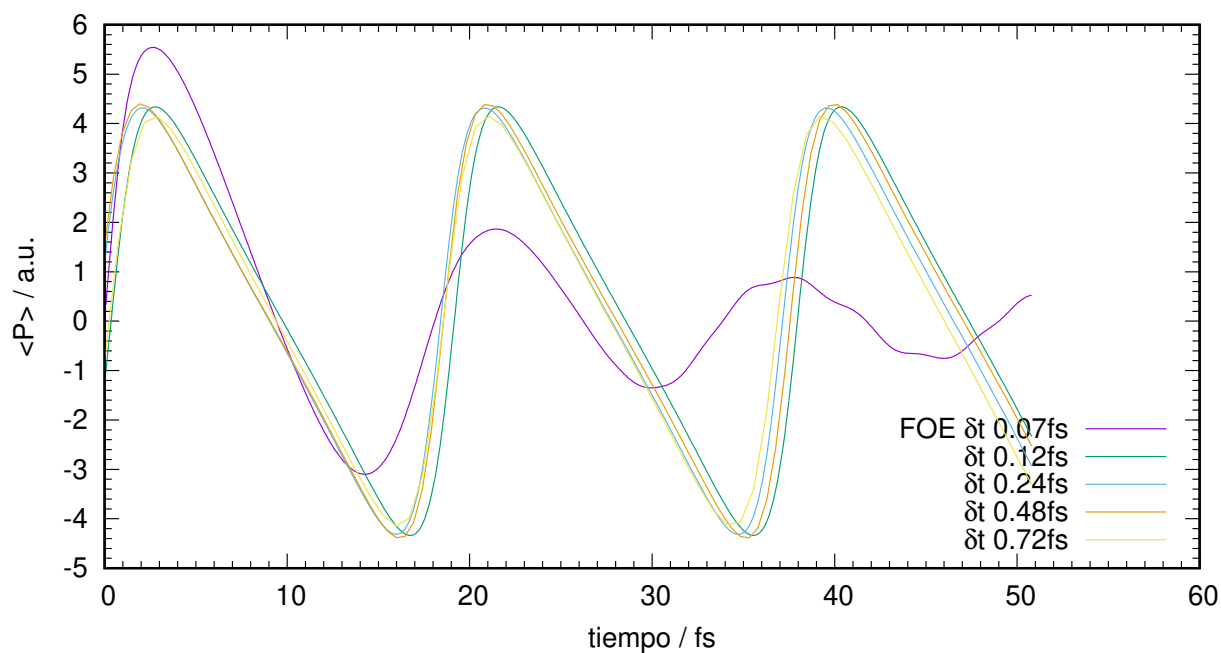
Debido a la forma secuencial en la que se presentó el desarrollo de este proyecto en la sección metodológica, se presentará de la misma forma los resultados y sus respectivos análisis.

6.1. Simulación de la dinámica molecular de H₂⁺

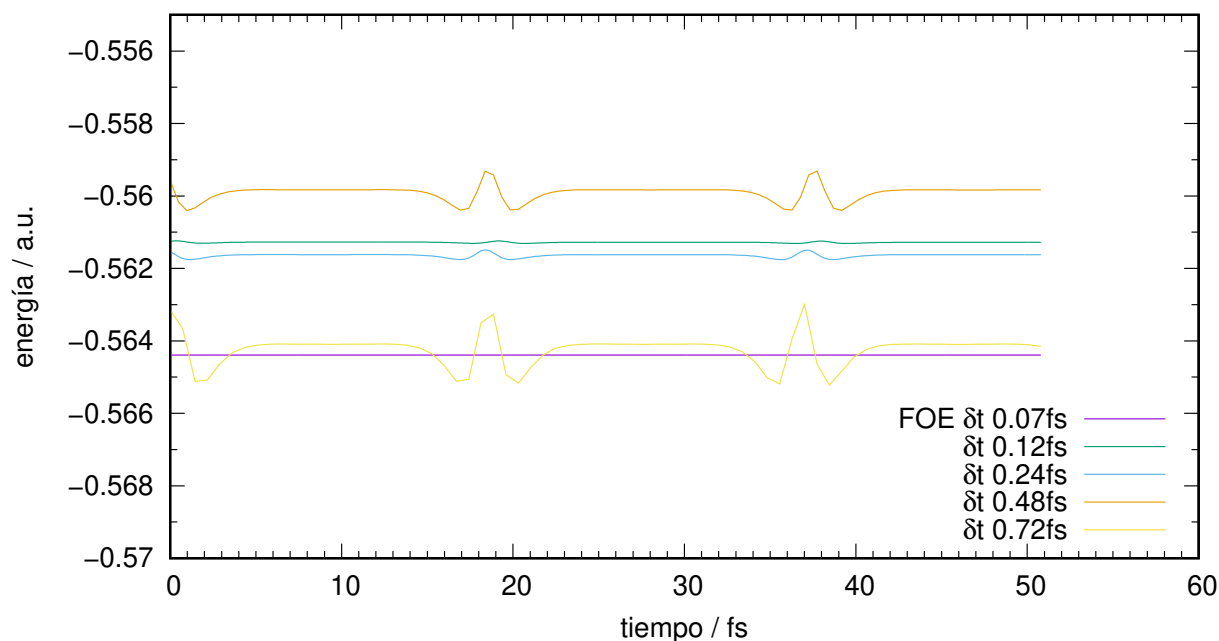
La simulación de la dinámica se empezó utilizando valores δt de 5, 10, 20 y 30 unidades atómicas de tiempo en los scripts de NWCHEM, correspondientes a tiempos de 0.12fs, 0.24fs, 0.48fs y 0.72fs respectivamente. Las dinámicas semiclásicas obtenidas mostraron un comportamiento cercano a los valores de referencia de la FOE como se muestran en la figuras 9 y 10.

En la figura 9a, correspondiente al $\langle R \rangle$ se puede observar cómo los periodos de vibración del ión molecular H₂⁺ simulado mediante NWCHEM casi se comparten con la FOE y se van desfasando ligeramente a medida que transcurre la simulación, principalmente porque en la dinámica de la FOE las amplitudes de las oscilaciones en $\langle R \rangle$ y $\langle P \rangle$ van desapareciendo a medida que pasa el tiempo, mientras que en la dinámica semi-clásica se mantienen constantes con los mismos periodos de vibración y valores esperados máximos y mínimos independientemente del valor de δt . En la figura 9b de $\langle P \rangle$, se puede apreciar que los momentos de la dinámica semi-clásica simulada durante el primer periodo de vibración también tienen un comportamiento muy similar con el momento de la FOE, compartiendo puntos de aceleración (pendiente positiva) y desaceleración (pendiente negativa), que se van desfasando respecto a la FOE, pero que en la dinámica semi-clásica se repiten exactamente igual.

Como era de esperarse, la energía de sistema simulado mostrado en la fig. 10 no es igual pero tampoco se aleja mucho de los valores de referencia de la FOE y se mantienen casi constantes, exceptuando en los puntos de cambio de dirección del movimiento de los átomos donde se observa una pequeña protuberancia en la gráfica. Además, se observa que tal como aumenta el valor de δt se da acentuando el comportamiento irregular en los valores de energía de la dinámica, y una pérdida de nitidez en las gráficas tanto de $\langle R \rangle$ como de $\langle P \rangle$, cuestión atribuible a la pérdida de definición que trae consigo aumentar el valor δt por la disminución de puntos generados en la dinámica. Sin embargo, de manera general se puede afirmar que el cambio de δt no afecta en gran medida el comportamiento de la simulación, desfasándose apenas por valores menores de 1 fs en $\langle R \rangle$ y $\langle P \rangle$. Además, se hace evidente que las trayectorias simuladas con energías similares son casi idénticas, y que el desfase mencionado anteriormente entre ellas depende de sus diferencias de

Figura 9. Valores esperados de la dinámica.**(a)** Gráfica valor esperado de posición ($\langle R \rangle$).**(b)** Gráfica valor esperado de momento ($\langle P \rangle$).

energía. Presentados estos datos se puede dar por cumplida la simulación semi-clásica.

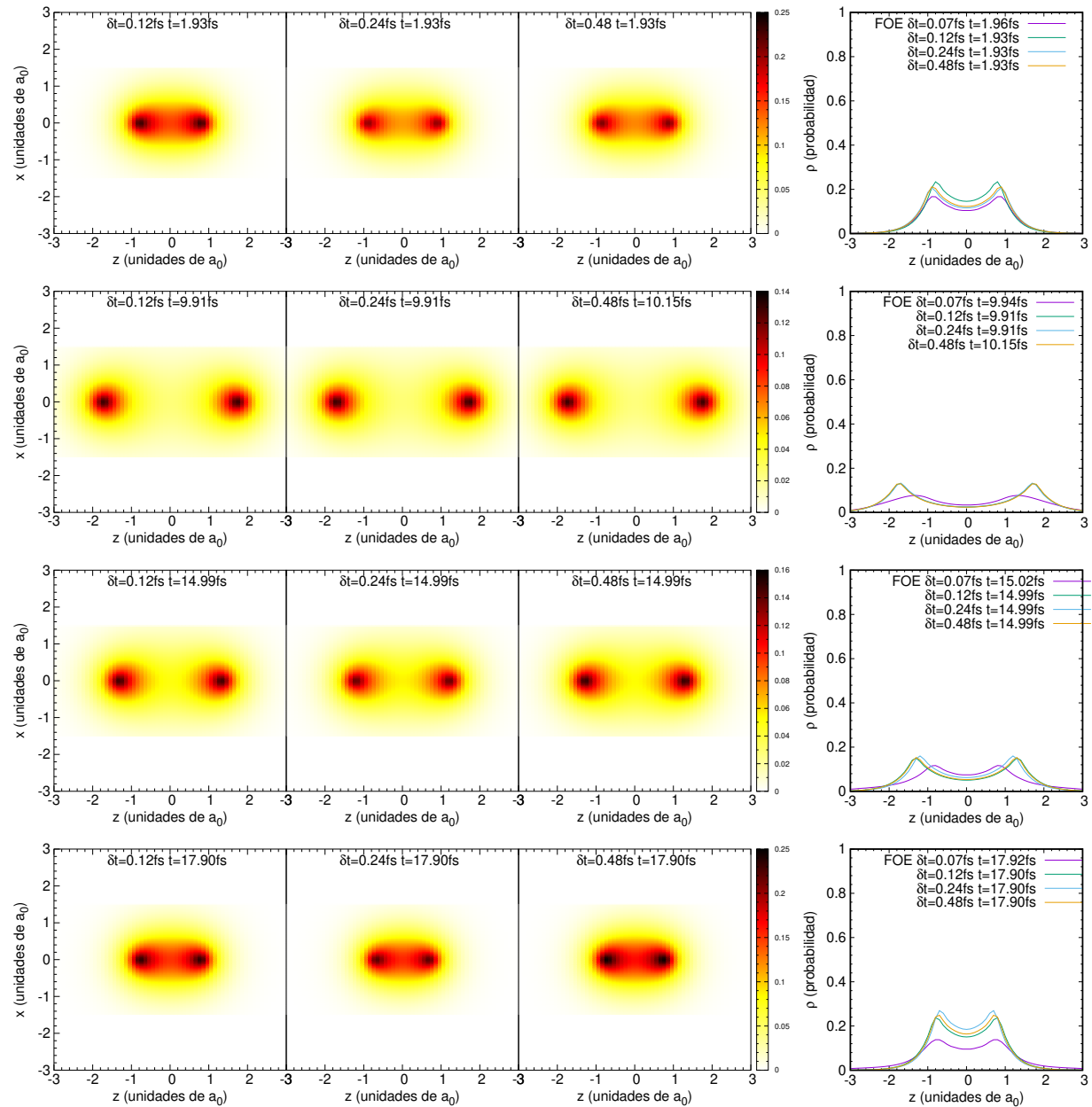
Figura 10. Energías de las dinámicas generadas.

6.2. Cálculo del orbital molecular $1s\sigma_g$ para cada tiempo t de la dinámica

En la fig. 11 se muestran las densidades de probabilidad para las dinámicas generadas en el paso anterior representando en cada fila tiempos clave, como lo son: alrededor de 2fs, en un estiramiento del enlace; alrededor de 10fs, en un estiramiento máximo y cambio de dirección del movimiento; alrededor de 15fs, periodo de compresión del enlace; y alrededor de 18fs, en un nuevo cambio de dirección y mínima compresión. Esta comparación se hace un poco tediosa por las diferencias que existen entre la malla de puntos en el tiempo de las dinámicas, entre ellas mismas y la de la FOE. De manera general, se manifiestan las mismas diferencias que se presentaron en la gráfica 9a en cuanto a los valores de mayor densidad de probabilidad, ya que estas densidades de probabilidad vienen de éstos primeros como se explica en la sección 5.2.

6.3. Densidades de flujo electrónico

Las densidades de flujo electrónico y densidades de probabilidad para los mismos tiempos seleccionados en el paso anterior se muestran en la fig. 12. La primera fila corresponde a los datos

Figura 11. Comparación de densidades de probabilidad para algunos tiempos.

obtenidos de la FOE, las filas 2, 3 y 4 a las densidades de probabilidad obtenidas por NMCHEM y los flujos obtenidos por el método de Okuyama con un δt de 0.12fs, 0.24fs, 0.72fs respectivamente. Los tiempos de la dinámica que se trataron de representar en las gráfica son 2fs, 10fs, 15fs y 19fs. Hay que aclarar que en la fila 1, las gráficas de la FOE contienen un par de curvas en azul, la curva azul superior corresponde a la densidad de probabilidad nuclear, la curva azul inferior a la densidad de flujo nuclear. Estas aparecen junto a los datos de la FOE porque también se obtuvieron de manera

mecano-cuántica, pero en este trabajo los núcleos se simularon de manera semi-clásica basándose en el teorema de Ehrenfest y por eso no aparecen en las demás.

Lo primero que salta a la vista de las gráficas de la figura 12 es cómo el flujo obtenido por el método de Okuyama cambia su magnitud respecto al valor de δt , a mayor δt mayor magnitud. Esto se puede explicar al observar la ecuación 13, y al revisar su implementación en el programa `flux.exe`. Como el método de Okuyama utiliza dos funciones de onda ligeramente diferentes, entendiéndose a 2 tiempos consecutivos en la dinámica simulada, a mayor sea la separación temporal de estas dos funciones de onda, mayor va a ser el flujo obtenido por este método, pero obviamente si las comparaciones se hacen a tiempos lo suficientemente separados se pierde la noción de integral de superposición que proponen Okuyama y Takatsuka para fundamentar su método. Por el contrario, si disminuye el valor de δt la diferencia de las funciones de onda va a ser tan pequeña que termina aproximándose a cero, como termina ocurriendo dentro de la aproximación de Born-Oppenheimer. Esto se había mencionado en su publicación (Okuyama y Takatsuka, 2009), por lo que al final, se tiene que encontrar un balance entre la pérdida de sensibilidad y definición que trae consigo aumentar el δt para obtener un flujo representativo, lo cual ha sido el objetivo principal desde el principio.

Sobre los tiempos representados en las gráficas se puede decir que tanto en la FOE como por el método de Okuyama es en los periodos de compresión y estiramiento cuando se obtienen mayores valores de flujo electrónico y cuando se dan los cambios de dirección de movimiento cuando se obtienen los valores más pequeños de flujo. Esto es apenas lógico si se entiende el concepto de flujo como la corriente electrónica que representa, siendo entonces cuando hay menor cambio de la densidad de probabilidad electrónica cuando menor flujo va surgir.

6.4. Resultados de la calibración

Los pasos descritos en la metodología se repitieron iterando para distintos δt con los que obtuvieron los resultados que se mostrarán a continuación.

La primera gráfica de la figura 13 muestra en escala logarítmica los valores máximos para cada

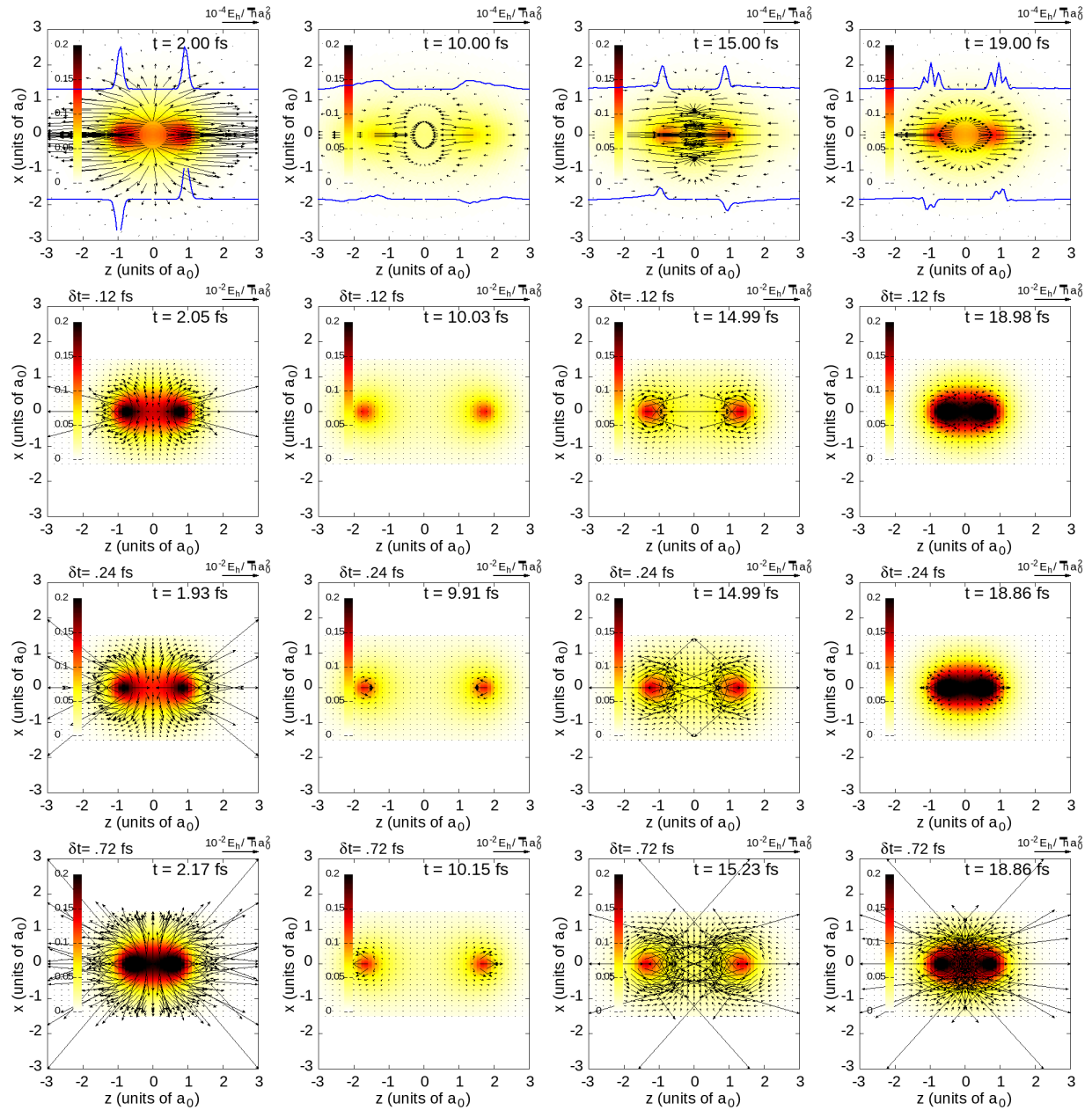
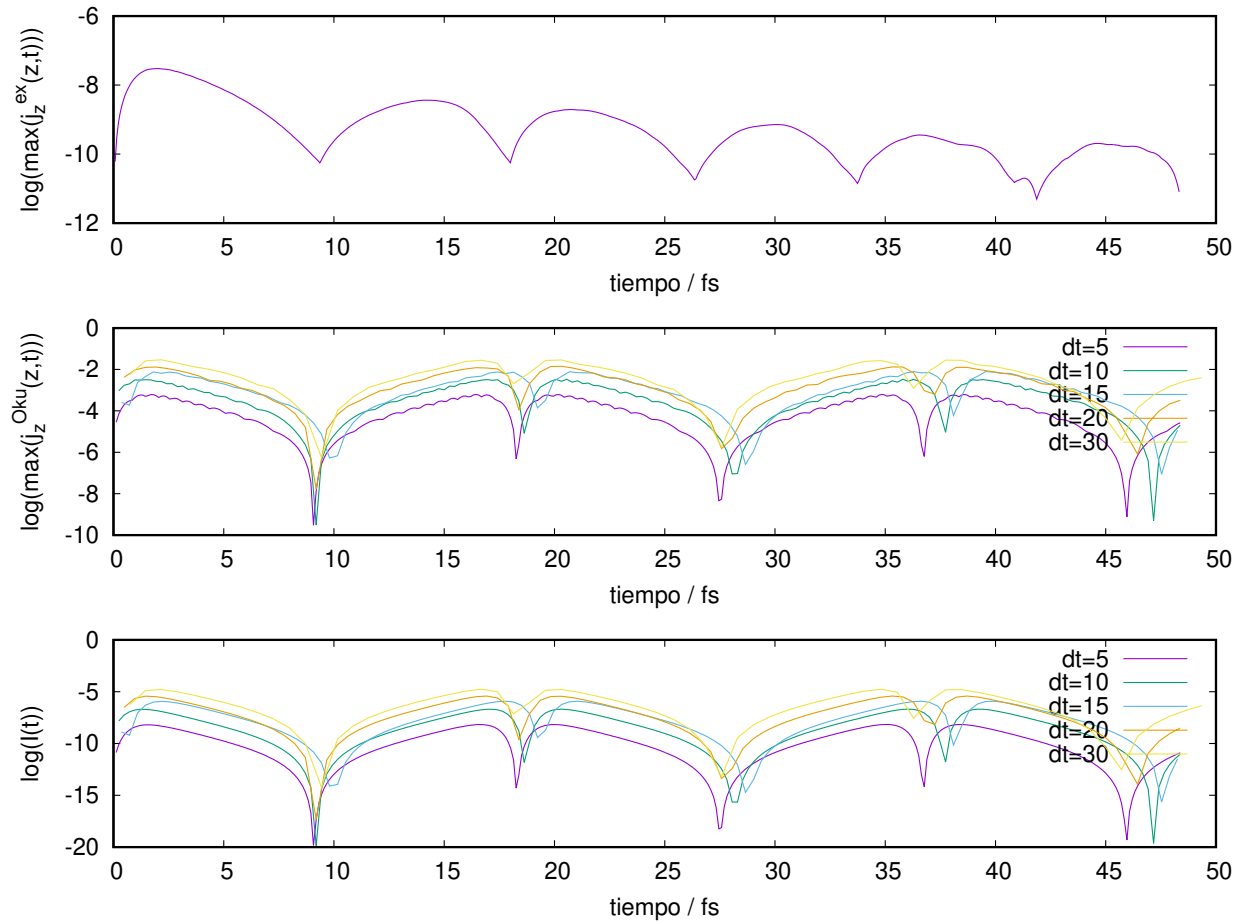
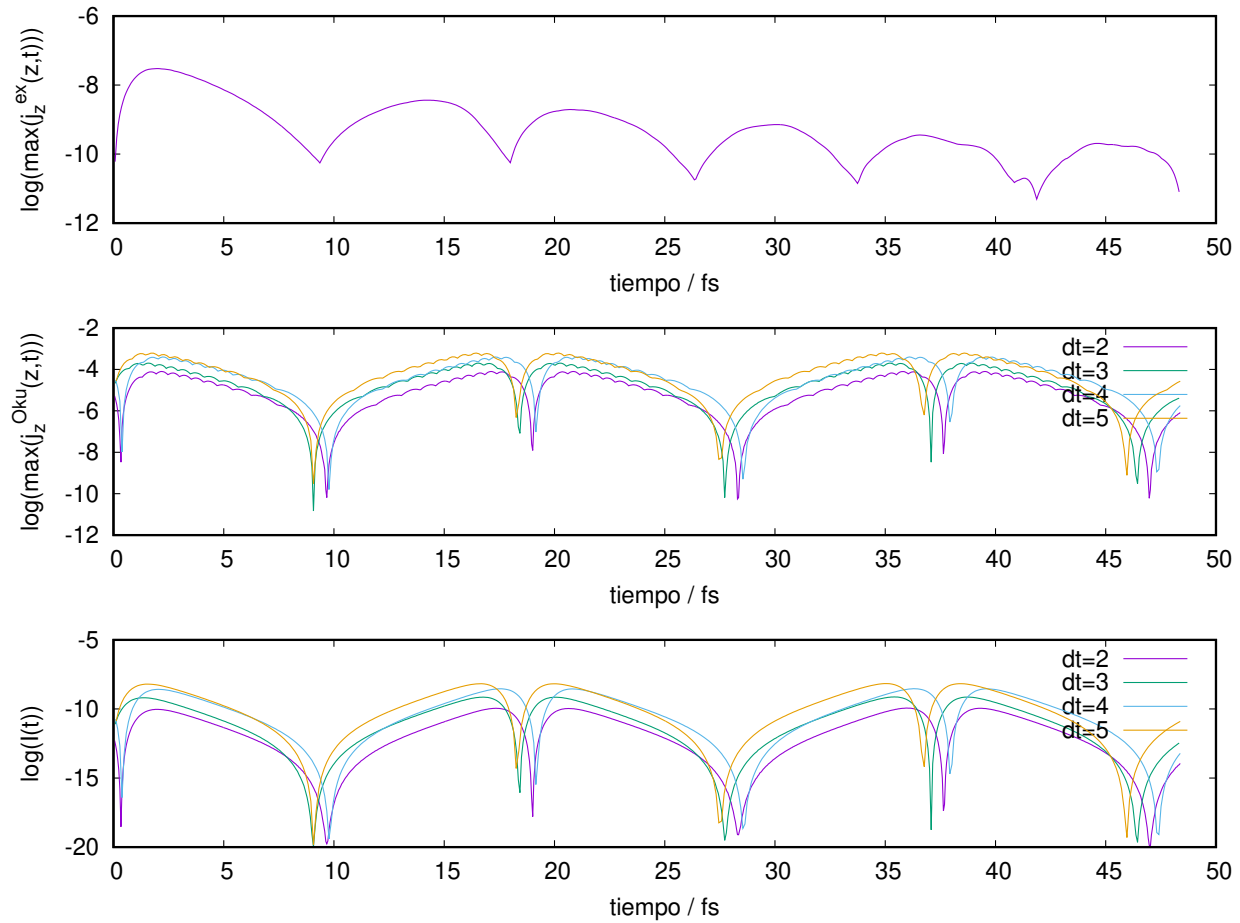
Figura 12. Densidades y flujos: FOE vs time-shift flux.

Figura 13. Resultados integral $I(t)$: δt general.

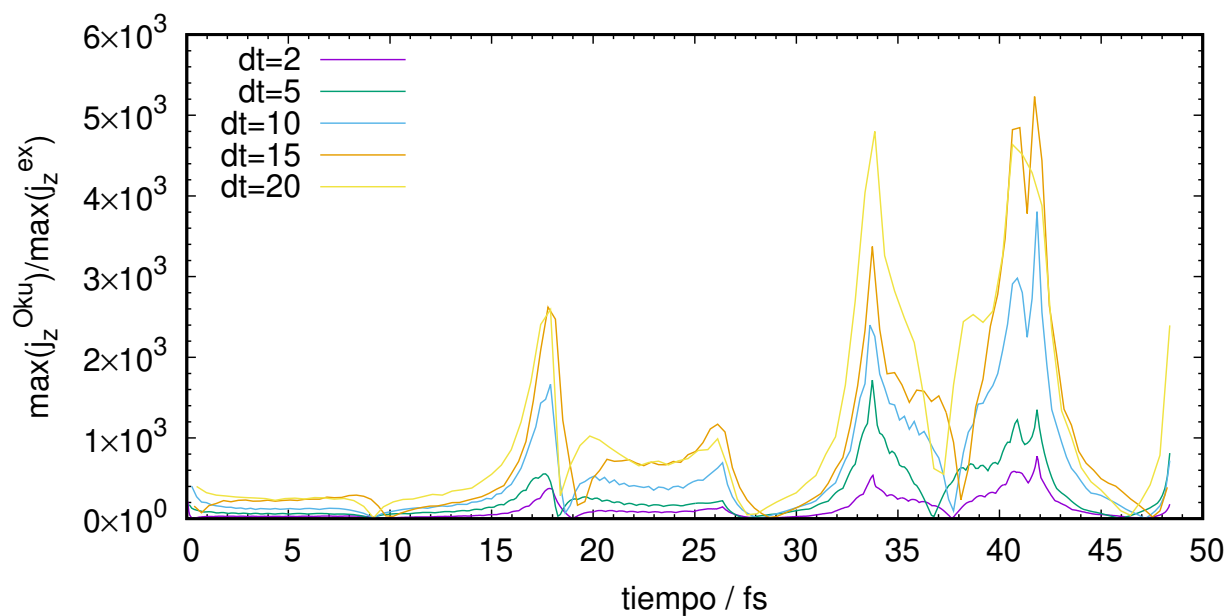
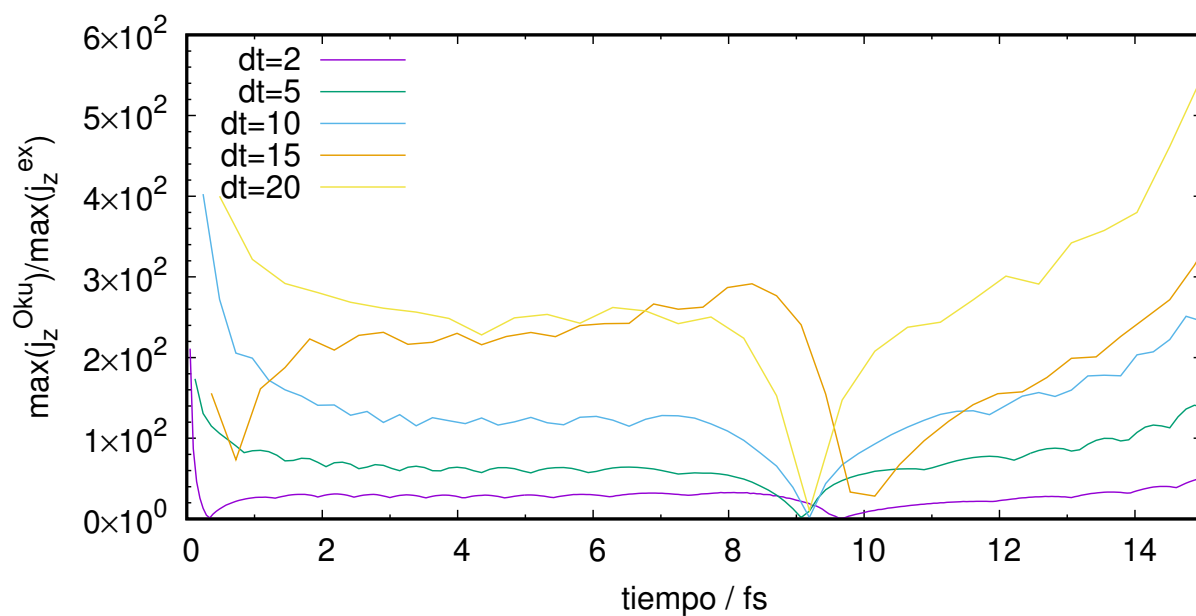
tiempo del flujo de la FOE obtenidos mediante la interpolación. La segunda gráfica de la figura 13 muestra de igual forma, en escala logarítmica, los valores máximos del flujo generado por el método de Okuyama. En ésta se puede observar oscilaciones en los valores además de valles bastante pronunciados cada 8 fs, esto últimos también están presentes en la gráfica de los valores máximos de la FOE que coinciden con los puntos de mayor estiramiento del enlace en la dinámica. La tercera gráfica de la figura 13 muestra, también en escala logarítmica los resultados de la integral $I(t)$ descrita en la sección 5.4. En todas las gráficas de la figura 13 es bastante notable que a medida que menor es el δt utilizado más se acerca a los valores de la FOE, sin embargo incluso con un $\delta t=0.12\text{fs}$, los valores máximos de flujo de Okuyama están a varios órdenes de magnitud por encima de los valores de la FOE.

La figura 14, repite los resultados mostrados en la figura 13 para δt más pequeños, en estas

Figura 14. Resultados integral $I(t)$: δt valores pequeños.

gráficas se muestran los mismos valles cada 8 fs y se alcanza a apreciar las oscilaciones presentes en la gráfica de los valores máximos del flujo de Okuyama. De manera general se puede afirmar que a medida que δt disminuye más se acercan los valores de flujo Okuyama a los valores de la FOE. Esto se hace más evidente si graficamos los valores máximos del flujo de Okuyama sobre los valores máximos del flujo de la FOE como se muestra en la figura 15

En ésta gráfica se puede apreciar que a medida que transcurre el tiempo y las amplitudes de la dinámica de la FOE van disminuyendo, y con estas también el valor del flujo, la diferencia entre el flujo de Okuyama y el flujo de la FOE se hacen cada vez mayores. También se puede apreciar con más detalle las oscilaciones mencionadas en las figuras 13 y 14. Si nos enfocamos en los primeros 15 fs, como se muestra en la figura 16 podemos ver que mientras con un δt de 2 u.a. (0.05fs) el flujo de Okuyama es 10 veces mayor, con δt de 5 u.a. (0.12fs) ya toma valores 100 veces mayores. Ya con

Figura 15. *Flujo Okuyama / Flujo FOE. 0-50 fs.***Figura 16.** *Flujo Okuyama / Flujo FOE. 0-15 fs.*

esto se puede afirmar que a medida que disminuye el δt con el que se genera la dinámica, los valores del flujo simulado se acercan cada vez más a los valores de la FOE, sin embargo, así mismo se hace mayor el tiempo de cómputo de la dinámica y la cantidad de cálculos necesarios para generar el flujo de Okuyama.

Aún así, el tiempo de cómputo necesario para aplicar este método es significativamente menor comparado con la aplicación de la FOE, tomando hasta 20 min con un δt de 2 u.a. (0.05 fs) y va disminuyendo a medida que δt se hace más grande, hasta 5 min o menos, todo esto teniendo en cuenta generar la dinámica, obtener los orbitales para cada punto de la dinámica y calcular el flujo por el método de Okuyama. Hay que tener en cuenta que este proyecto trabajó con una de las moléculas más sencillas, si no la más, y estos tiempos pueden verse incrementados en gran medida si la geometría y estructura de la molécula simulada es más compleja.

7. Conclusiones

Se presentó la aplicación del método “time-shift flux” de Okuyama y Takatsuka variando el término Δt y relacionándolo con una dinámica semi-clásica en base al Teorema de Ehrenfest para compararlo con el método de la función de onda exacta que sigue una dinámica meramente mecano-cuántica, la cuál es la única referente para el sistema H₂⁺, en un marco temporal de 2100 u.a. de tiempo (aprox. 50fs). Se encontró que a medida que los valores de δt disminuyen, más se acerca el flujo generado por el método de Okuyama al generado por la FOE, encontrándose que un $\delta t = 2$ u.a. (0.05 fs) durante los primeros 15 fs de la dinámica daba una buena aproximación a los valores de la FOE. Valores menores a 2 u.a. (0.05 fs) aumentan significativamente los tiempos de cómputo y se empiezan a notar errores en la aproximación como oscilaciones en los valores máximos de flujo generados por la cada vez menor diferencia entre los orbitales simulados. Se encontró bastante útil el flujo simulado por el método de Okuyama para representar procesos sencillos como lo es la vibración del ión molecular H₂⁺ ya que describe de manera cualitativa como la densidad de probabilidad electrónica se va desplazando a lo largo de tiempo, siendo esto mucho más educativo que observar una densidad de probabilidad fija. El método como tal es muy sencillo de implementar

si se tiene una buena dinámica de la cual partir y generar las funciones de onda electrónicas, y así mismo su densidad de probabilidad. Sería muy interesante comparar el desempeño del método de Okuyama frente a otros métodos aproximativos similares que buscan obtener una densidad de flujo electrónico como lo hace Denis Diestler en su método de Coupled-Channels. Así mismo se espera poder verificar en un futuro los resultados con datos experimentales de resolución temporal de attosegundos para este tipo de procesos, o bien utilizar este método para estudiar otros procesos similares.

Referencias bibliográficas

- Andrae, D., Barth, I., Bredtman, T., Hege, H.-C., Manz, J., Marquardt, F. & Paulus, B. (2011). electronic quantum fluxes during pericyclic reactions exemplified for the cope rearrangement of semibullvalene. *The Journal of Physical Chemistry B*, 115(18), 5476-5483. <https://doi.org/10.1021/jp110365g>
- Barth, I., Hege, H.-C., Ikeda, H., Kenfack, A., Koppitz, M., Manz, J., Marquardt, F. & Paramonov, G. K. (2009). Concerted quantum effects of electronic and nuclear fluxes in molecules. *Chemical Physics Letters*, 481(1-3), 118-123. <https://doi.org/10.1016/j.cplett.2009.09.011>
- Barth, I. & Jörn Manz, K. Y., Yasuteru Shigeta. (2006). Unidirectional electronic ring current driven by a few cycle circularly polarized laser pulse: Quantum model simulations for Mg-Porphyrin. *Journal of the American Chemical Society*, 128, 7043-7049. <https://doi.org/10.1021/ja057197l>
- Born, M. & Oppenheimer, R. (1927). Zur quantentheorie der molekeln. *Annalen der Physik*, 389(20), 457-484. <https://doi.org/10.1002/andp.19273892002>
- Bratos, S., Mirloup, F., Vuilleumier, R., Wulff, M. & Plech, A. (2004). X-ray “filming” of atomic motions in chemical reactions. *Chemical Physics*, 304(1-3), 245-251. <https://doi.org/10.1016/j.chemphys.2004.03.065>
- Cramer, C. J. (2004). *Essentials of Computational Chemistry: Theories and Models*. John Wiley & Sons, Ltd.
- de Boor, C. (1978). *A Practical Guide to Splines*. Springer-Verlag New York.
- Diestler, D. J. (2012). Coupled-Channels quantum theory of electronic flux density in electronically adiabatic processes: Fundamentals. *The Journal of Physical Chemistry A*, 116(11), 2728-2735. <https://doi.org/10.1021/jp2078437>
- Diestler, D. J., Kenfack, A., Manz, J. & Paulus, B. (2012). Coupled-Channels quantum theory of electronic flux density in electronically adiabatic processes: Application to the hydrogen

- molecule ion. *The Journal of Physical Chemistry A*, 116(11), 2736-2742. <https://doi.org/10.1021/jp207844w>
- Ehrenfest, P. (1927). Bemerkung über die angenäherte gültigkeit der klassischen mechanik innerhalb der quantenmechanik. *Zeitschrift für Physik*, 45(7-8), 455-457. <https://doi.org/10.1007/bf01329203>
- Hermann, G., Pohl, V., Dixit, G. & Tremblay, J. C. (2020). Probing electronic fluxes via time-resolved x-ray scattering. *Physical Review Letters*, 124(013002). <https://doi.org/10.1103/PhysRevLett.124.013002>
- Levine, I. N. (2001). *Química Cuántica*. Pearson Educación.
- Nafie, L. A. (1997). Electron transition current density in molecules: 1. Non-Born-Oppenheimer theory of vibronic and vibrational transitions. *The Journal of Physical Chemistry A*, 101(42), 7826-7833. <https://doi.org/10.1021/jp9706137>
- Okuyama, M. & Takatsuka, K. (2009). Electron flux in molecules induced by nuclear motion. *Chemical Physical Letters*, 476(1-3), 109-115. <https://doi.org/10.1016/j.cplett.2009.05.077>
- Schiff, L. I. (1986). *Quantum Mechanics*. McGraw-Hill.
- Schrödinger, E. (1926a). Quantisierung als Eigenwertproblem. *Annalen der physik*, 384(4), 361-376. <https://doi.org/10.1002/andp.19263840404>
- Schrödinger, E. (1926b). Quantisierung als Eigenwertproblem. *Annalen der physik*, 384(6), 489-527. <https://doi.org/10.1002/andp.19263840602>
- Schrödinger, E. (1926c). Quantisierung als Eigenwertproblem. *Annalen der physik*, 385(13), 437-490. <https://doi.org/10.1002/andp.19263851302>
- Schrödinger, E. (1926d). Quantisierung als Eigenwertproblem. *Annalen der physik*, 386(18), 109-139. <https://doi.org/10.1002/andp.19263861802>
- Steiner, E. & Fowler, P. W. (2001). Patterns of ring currents in conjugated molecules: A few-electron model based on orbital contributions. *The Journal of Physical Chemistry A*, 105, 9553-9562. <https://doi.org/10.1021/jp011955m>

Apéndices

Apéndice A Código del script general

```
#!/bin/bash
```

```
#Compilacion de programas
```

```
gfortran nwchem/union.f -o nwchem/union.exe
```

```
gfortran flux.f -o flux.exe
```

```
gfortran -c inter_mod.f90 #Para generar inter_mod.o
```

```
gfortran diff.f inter_mod.o -o diff.exe
```

```
#OBTENCIÓN DE LA DINÁMICA MOLECULAR R(T)
```

```
timegap=2000
```

```
tstep=(20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2)
```

```
rm Okuflux-t.dat I-*.dat
```

```
for i in "${tstep[@]}"
```

```
do
```

```
    dtime="$i"
```

```
    nstep='echo "scale=0;□$timegap/$dtime" | bc '
```

```
    cd ./nwchem
```

```
    rm *.qmdrst
```

```
    sed s/numero_pasos/"$nstep"/g ../ref/dynH2+.ref > dynH2+.temp
```

```
    sed s/intervalo_tiempo/"$dtime"/g dynH2+.temp > dynH2+.nw
```

```
    echo "starting"
```

```
    nwchem dynH2+.nw > dynH2+.log
```

```
    rm union.inp
```

```
    echo "$nstep□$dtime" > union.inp
```

```
    ./union.exe < union.inp #aisla los valores de t, R, P y E en un archivo
```

```
# cp nuclear.dat ../ref
```

```
    cp nuclear.dat ../ref/nuclear-"$dtime".dat
```

```
    rm nuclear.dat
```

```
    cd ..
```

```
#OBTENCION DE ORBITALES PARA CADA R(t) DE LA DINÁMICA
```

```
    rm ./mo/*.cube
```

```
    rm ./orbitals/*
```

```

rm ./listaMO.txt
tiempo=$(awk '{print $1}' ./ref/nuclear-"$dtime".dat)
distancia=$(awk '{print $2}' ./ref/nuclear-"$dtime".dat)
cp ./ref/h2+.ref ./nwchem
cd ./nwchem

for i in "${distancia[@]}" #genera un rho para cada <R> listado en nuclea
do
    RA='echo "scale=17;_ $i/2" | bc'; RB='echo "scale=17;_-$i/2" | bc'
    sed s/RA/"$RA"/g h2+.ref > h2+.temp; sed s/RB/"$RB"/g h2+.temp > h2+.nw
    nwchem h2+.nw > h2+.log
    mv sigmag.cube ../mo/mo$i.cube
done

cd ..

#OBTENCION DEL FLUJO Y GRÁFICAS
# rm ./graficas/png/*.png
# rm ./listaPNG.txt
rm ./flux/*.dat
ls -tr ./mo > ./listaMO.txt #

trayectoria=('cat listaMO.txt')
ntrayec=${#trayectoria[@]}
contador=1

rm list.txt Okuflux-t.dat
while [ $contador -lt $ntrayec ]
do
    cp ./mo/${trayectoria[$(($contador-1))]} ./moA.cube
    cp ./mo/${trayectoria[$contador]} ./moB.cube
    ./flux.exe
#    gnuplot flux.gnu
    echo "${tiempo[$contador]}" | bc >> Okuflux-t.dat
    cat Okuflux.dat >> Okuflux-t.dat
    cp orbitals.dat ./orbitals/$(($dtime))orbital-$(($contador*$dtime)).dat
#    mv flux.png ./graficas/png/flux-$(($contador-1))-$contador.png
#    cp Okuflux.dat ./flux/Okuflux-$(($contador*$dtime)).dat
    echo "${tiempo[$contador]}_ ${trayectoria[$contador]}"
    contador=$((contador+1))
    dt='echo "scale=2;_ $dtime/41.341" | bc -l'
    tfs='echo "scale=2;_ ${tiempo[$contador]}/41.341" | bc -l'
    sed s/tiempo/$tfs/ Okuobs.ref > Okuobs.temp

```



```

    sed s/dtime/$dt/ Okuobs.temp > Okuobs.gnu
    gnuplot Okuobs.gnu; mv ./Okuobs.png ./Okuobs"$tfs"$fs-"$dtime".au.png
    echo "Okuobs"$tfs"$fs-"$dtime".au.png" >> list.txt
done

cp ./orbitals/$(($dtime))orbital-80.dat ./prueba1/
cp ./orbitals/$(($dtime))orbital-410.dat ./prueba1/
cp ./orbitals/$(($dtime))orbital-620.dat ./prueba1/
cp ./orbitals/$(($dtime))orbital-740.dat ./prueba1/
cp ./orbitals/$(($dtime))orbital-420.dat ./prueba1/
./copygraphs.sh
duration=30 #duración del clip de video
framerate='echo "scale=2;_$_nstep/$duration" | bc -l'

mencoder mf://@list.txt -mf w=800:h=600:fps=$framerate:type=png -ovc copy
rm ./Okuobs*.png

#EJECUCIÓN DE LA INTEGRAL DE DIFFERENCIA  $I(t)$ 

    echo "201_$_nstep_$_dtime_0.1d0" > diff.inp
# echo "# $dtime" >> I-$dtime.dat
./diff.exe < diff.inp
cp ./I-dt.dat ./I-$dtime.dat
cp ./diff-dt.dat ./diff-$dtime.dat
done

./graficador.sh

```

Apéndice B Código de los programas en fortran utilizados**union.f**

```

program union
implicit none
real*8 :: void
real*8, allocatable :: t(:), R1(:), R2(:), P(:), E(:)
integer :: i,j,k,natom,nt,dt
character :: atom

natom=2
read(5,*)nt,dt
allocate (t(nt),R1(nt),R2(nt),P(nt),E(nt))

open(1,file='dynH2+.xyz')
do i=1,nt+1
  read(1,*)
  read(1,*)t(i), E(i)
  read(1,*)atom, void, R1(i), void, void,P(i)
  read(1,*)atom, void, R2(i)
enddo

close(1)

open(2,file='nuclear.dat')
do i=1,nt+1
  write(2,*)t(i)*dt,(ABS(R1(i))+ABS(R2(i))),P(i),E(i)
enddo
close(2)

stop 'ok'
end program

```

flux.f

```

program flux
implicit none
integer :: i,j,k,nxA,nyA,nz1,natom1
real*8 :: dxA,xmin1,dyA,ymin1,z1,dz1,zmin1
real*8, allocatable :: xA(:),yA(:)
real*8 :: val1
real*8, allocatable :: psiA(:,:),root1(:,:)
integer :: l,m,n,nxB,nyB,nz2,natom2
real*8 :: dxB,xmin2,dyB,ymin2,z2,dz2,zmin2
real*8, allocatable :: xB(:),yB(:)

```

```

real*8 :: val2
real*8, allocatable :: psiB (:,:), root2 (:,:)
real*8, allocatable :: psi0 (:,:)
real*8, allocatable :: jx (:,:), jy (:,:)
real*8 :: a,b,c,d

```

!Notice that array xA should be equal to array xB, then both orbitals are d

```

open(1, file='moA.cube')
read(1,*)
read(1,*)
read(1,*)natom1, xmin1, ymin1, zmin1
read(1,*)nxA, dxA, val1, val1
read(1,*)nyA, val1, dyA, val1
read(1,*)nz1, val1, val1, dz1

do i=1, natom1
    read(1,*)
enddo

allocate(psiA(nxA, nyA), xA(nxA), yA(nyA))
do i=1, nxA
    xA(i)=xmin1+(i-1)*dxA
    do j=1, nyA
        yA(j)=ymin1+(j-1)*dyA
        read(1,*)psiA(i, j) !here we have psiA for the grid points (x,y)
    enddo
enddo
close(1)

open(2, file='moB.cube')
read(2,*)
read(2,*)
read(2,*)natom2, xmin2, ymin2, zmin2
read(2,*)nxB, dxB, val2, val2
read(2,*)nyB, val2, dyB, val2
read(2,*)nz2, val2, val2, dz2
do l=1, natom2
    read(2,*)
enddo
allocate(psiB(nxB, nyB), xB(nxB), yB(nyB))
do l=1, nxB
    xB(l)=xmin2+(l-1)*dxB
    do m=1, nyB
        yB(m)=ymin2+(m-1)*dyB
    enddo
enddo

```

```

        read(2,*)psiB(1,m) !here we have psiB for the grid points (x,y)
    enddo
enddo
close(2)

if((nxA.ne.nxB).or.(nyA.ne.nyB)) then
    write(6,*)'Problem: grid for psiA is not the same for psiB'
    stop
endif

!Now we attemp to calculate jx=psiB*d_x(psiA)-psiA*d_x(psiB) where d_x mean
allocate(jx(nxA,nyA))
jx=0.d0
do j=1,nyA
    do i=2,nxA-1
        jx(i,j)=psiB(i,j)*(psiA(i+1,j)-psiA(i-1,j))/(xA(i+1)-xA(i-1))
$        -psiA(i,j)*(psiB(i+1,j)-psiB(i-1,j))/(xA(i+1)-xA(i-1))
    enddo
enddo

!Now we attemp to calculata jy...
allocate(jy(nxA,nyA))
jy=0.d0
do i=1,nxA
    do j=2,nyB-1
        jy(i,j)=psiB(i,j)*(psiA(i,j+1)-psiA(i,j-1))/(yA(j+1)-yA(j-1))
$        -psiA(i,j)*(psiB(i,j+1)-psiB(i,j-1))/(yA(j+1)-yA(j-1))
    enddo
enddo

!Lets write down some points of jx,jy

open(1,file='flux.dat')
write(1,*)'#',nxA/10,nyA/2
do i=1,nxA,10
    do j=1,nyA,2
        write(1,*)xA(i),yA(j),-jx(i,j),-jy(i,j)
        enddo
        write(1,*)
    enddo
close(1)

open(3,file='orbitals.dat')
open(2,file='Okuflux.dat')
! write(1,*)'#',nxA,nyA

```

```

do i=1,nxA
  do j=1,nyA
!      write(1,*)xA(i),yA(j),-jx(i,j),-jy(i,j)
    write(3,*)xA(i),yA(j),psiA(i,j)
    if(xA(i).eq.0) then
      write(2,*)xA(i),yA(j),-jx(i,j),-jy(i,j)
    endif
  enddo
!      write(1,*)
  write(3,*)
enddo
!      close(1)
close(2)
close(3)
!      open(1,file='orbitals.dat')
!      do i=1,nxA,1
!        do j=1,nyA,1
!          write(1,*)xA(i),yA(j),psiA(i,j)
!        enddo
!        write(1,*)
!      enddo
!      close(1)

stop 'ok'
end program flux

```

diff.f

```

program diff
  use inter !Utiliza el modulo inter_mod.f90 para interpolar
! Las variables "xtmp(:)", "ytmp(:)" y "finter(:)" estan declaradas
! dentro del módulo inter_mod.f90. xtmp(:) e ytmp(:) representan "x" y
! "f(x)" respectivamente. precubsp() prepara la interpoación. Luego la
! función finter(:) calcula la función f(new_x) en el nuevo grid new_x(:)
! En este caso queremos interpolar el dominio temporal de FOE en el grid
! de dominio temporal de Okuyama, así, se puede calcular Okuyama con
! cualquier paso de tiempo y luego se ajusta una única FOE al grid
! temporal de Okuyama.
  implicit none
  integer :: i,j
  real*8, allocatable :: Oku(:, :), phi(:, :), D(:, :), N1(:, :), N2(:, :),
  real*8, allocatable :: zA(:, :), zB(:, :), tA(:, :), tB(:, :), val1(:, :), val2(:, :),
  real*8, allocatable :: FOE(:, :), newFOE(:, :),
  integer :: nz2, nz1, nstep2, nstep1
  real*8 :: x0, jx, dz, dt

```

*!La idea es leer los datos en grid iguales para poder
!compararlos*

```

read(5,*)nz1,nstep1,dt,dz

allocate(Oku(nstep1,nz1),tA(nstep1),zA(nz1))
open(1,file='Okuflux-t.dat')
do i=1,nstep1
    read(1,*)tA(i)
    do j=1,nz1
        read(1,*)x0,zA(j),jx,Oku(i,j)
    enddo
enddo
close(1)

```

```

!do i=1,nstep
!do j=1,nz
!write(6,*)tA(i),zA(j),Oku(i,j)
!enddo
!enddo

```

```

nz2=201
nstep2=700 !700
allocate(FOE(nstep2,nz2),zB(nz2),tB(nstep2),phi(nstep2,nz2))

```

```

!Lee FOE original
open(1,file='obs-e-time.dat')
do i=1,nstep2
do j=1,nz2
    read(1,*)zB(j),phi(i,j),FOE(i,j),tB(i)
enddo
enddo
close(1)

```

```

!Parte que revisa si las cosas van bien
write(6,*)'nz1:',nz1,'nz2:',nz2
write(6,*)'nstep1:',nstep1,'nstep2:',nstep2
write(6,*)'size_tA:',size(tA),'size_tB:',size(tB)
write(6,*)'tA:␣(',tA(1),'-',tA(size(tA)),")"
write(6,*)'tB:␣(',tB(1),'-',tB(size(tB)),")"

```

```

!Se lleva a cabo la interpolación
nstep1=nstep1
allocate(xtmp(nstep2),ytmp(nstep2))

```

```

allocate (newFOE( nstep1 , nz2 ))
do j=1, nz2
    do i=1, nstep2
        xtmp( i )=tB( i )
        ytmp( i )=FOE( i , j )
    enddo
    call precubsp( nstep2 )
    do i=1, nstep1
        newFOE( i , j )= finter( tA( i ))
    enddo
enddo
!      open( 2 , file = 'I-dt. dat '
!      write( 2 , *) '##' , dt
!      do i=1, nstep1
!      do j=1, nz2
!          write( 2 , *) tA( i ) , zB( j ) , newFOE( i , j ) , Oku( i , j )
!      enddo
!      write( 2 , *)
!      enddo
!      close( 2 )

!Se lleva a cabo la integración para calcular I(t)
allocate (D( nstep1 ) , N1( nstep1 ) , N2( nstep1 ))
D=0. d0
do i=1, nstep1
    do j=1, nz2/2
        D( i ) = D( i ) + dz* ( newFOE( i , j )- Oku( i , j ) )**2
    enddo
enddo

open( 1 , file = 'I-dt. dat ' )
write( 1 , *) '##' , dt

allocate ( val1 ( nz2 / 2 ) , val2 ( nz2 / 2 ))
do i=1, nstep1
    do j=1, nz2/2
        val1 ( j )=Oku( i , j )
        val2 ( j )=newFOE( i , j )
    enddo
    write( 1 , *) tA( i )/41.341 d0
$          , maxval( dabs( val2 ))
$          , maxval( dabs( val1 ))
$          , D( i )
enddo
write( 1 , *)

```

```

    write(1,*)
close(1)
stop 'ok'
end program diff

```

inter_mod.f90

```

module inter
  real*8, dimension(:), allocatable :: xtmp, ytmp
  contains
! %%%%%%%%%%
  real*8 function finter(x)
    implicit none
    integer nmax, nf
    parameter (nmax=200000)
    real*8 :: x, y
    real*8 :: xx, yy, yd2, aa
    common/ splines / xx(nmax), yy(nmax), yd2(nmax), aa(2, nmax), nf

    if ((x.gt.xx(nf)).or.(x.lt.xx(1))) then
      write(6,*) 'x,xx(1),xx(nf)',x,xx(1),xx(nf)
      stop 'Prob0:␣coef'
    endif
    call splint(xx,yy,yd2,nf,x,y)
    finter=y

    return
  end function finter
! %%%%%%%%%%
  real*8 function dfinter(x,dx)
    implicit none
    integer nmax, nf
    parameter (nmax=200000)
    real*8 :: x, y2, y1, dx
    real*8 :: xx, yy, yd2, aa
    common/ splines / xx(nmax), yy(nmax), yd2(nmax), aa(2, nmax), nf

    if ((x.gt.xx(nf)).or.(x.lt.xx(1))) then
      write(6,*) 'x,xx(1),xx(nf)',x,xx(1),xx(nf)
      stop 'Prob0:␣coef'
    endif
    call splint(xx,yy,yd2,nf,x+dx/2.d0,y2)
    call splint(xx,yy,yd2,nf,x-dx/2.d0,y1)
    dfinter=(y2-y1)/dx

```



```

return
end function dfinter
! %%%%%%%%%%%
subroutine precubsp( ntmp)
implicit real*8 (a-h,o-z)
integer nmax , ntmp , nf
parameter( nmax=200000)
real*8 :: xx , yy , yd2 , aa
common/ splines /xx(nmax) , yy(nmax) , yd2(nmax) , aa(2 ,nmax) , nf

nf=ntmp
do i=1,nf
    xx(i)=xtmp(i)
    yy(i)=ytmp(i)
enddo

!      ordena

vmin=xx(1)
do i=2,nf
    if (xx(i).lt.vmin) vmin=xx(i)
enddo
vmax=xx(1)
do i=2,nf
    if (xx(i).gt.vmax) vmax=xx(i)
enddo
vmin=vmin-2
ncot=0
1 continue
ncot=ncot+1
rant=vmax+1.d0
do i=1,nf
    if ((xx(i).lt.rant).and.(xx(i).gt.vmin)) then
        rant=xx(i)
        nbu=i
    endif
enddo
aa(1,ncot)=xx(nbu)
aa(2,ncot)=yy(nbu)
xx(nbu)=vmin-3
if (ncot.lt.nf) go to 1
do i=1,nf
    xx(i)=aa(1,i)
    yy(i)=aa(2,i)
    if (xx(i).eq.aa(1,i+1)) then

```

```

        write(6,*) 'Warning_'
        write(6,fmt='(i5,2e16.8)') i,xx(i),yy(i)
        stop 'Hay puntos repetidos'
    endif
enddo

!      Calcula la primera derivada en el primer y en el ultimo punto

ydi=(yy(2)-yy(1))/(xx(2)-xx(1))
ydf=(yy(nf)-yy(nf-1))/(xx(nf)-xx(nf-1))

!      Calcula las segundas derivadas

call spline(xx,yy,nf,ydi,ydf,yd2)

return
end subroutine precubsp
! %%%%%%%%%%%
subroutine spline(x,y,n,yp1,ypn,y2)
integer n,nmax
real*8 yp1,ypn,x(n),y(n),y2(n)
parameter (nmax=200000)
integer i,k
real*8 p,qn,sig,un,u(nmax)
if (yp1.gt..99d30) then
    y2(1)=0.d0
    u(1)=0.d0
else
    y2(1)=-0.5d0
    u(1)=(3.d0/(x(2)-x(1)))*((y(2)-y(1))/(x(2)-x(1))-yp1)
endif
do i=2,n-1
    sig=(x(i)-x(i-1))/(x(i+1)-x(i-1))
    p=sig*y2(i-1)+2.d0
    y2(i)=(sig-1.d0)/p
    u(i)=(6.d0*((y(i+1)-y(i))/(x(i+1)-x(i))-(y(i)-y(i-1))&
        /(x(i)-x(i-1)))/(x(i+1)-x(i-1))-sig*u(i-1))/p
enddo
if (ypn.gt..99d30) then
    qn=0.d0
    un=0.d0
else
    qn=0.5d0
    un=(3.d0/(x(n)-x(n-1)))*(ypn-(y(n)-y(n-1))/(x(n)-x(n-1)))
endif
endif

```

```

y2(n)=(un-qn*u(n-1))/(qn*y2(n-1)+1.d0)
do k=n-1,1,-1
    y2(k)=y2(k)*y2(k+1)+u(k)
enddo
return
end subroutine spline
! %%%%%%%%%%
subroutine splint(xa,ya,y2a,n,x,y)
integer n
real*8 x,y,xa(n),y2a(n),ya(n)
integer k,khi,klo
real*8 a,b,h
klo=1
khi=n
1 if (khi-klo.gt.1) then
    k=(khi+klo)/2
    if (xa(k).gt.x) then
        khi=k
    else
        klo=k
    endif
go to 1
endif
h=xa(khi)-xa(klo)
if (h.eq.0.d0) stop 'bad_xa_input_in_splint'
a=(xa(khi)-x)/h
b=(x-xa(klo))/h
y=a*ya(klo)+b*ya(khi)+&
((a**3-a)*y2a(klo)+(b**3-b)*y2a(khi))*(h**2)/6.d0
return
end subroutine splint
! %%%%%%%%%%
end module inter
```

Apéndice C Scripts de referencia para NWCHEM**dynH2+.ref**

```
start
echo
charge 1
geometry units au noautosym noautoz
  H  0.0  0.684  0.0  0.0  0.0001813  0.0
  H  0.0 -0.684  0.0  0.0 -0.0001813  0.0
  symmetry C1
end
basis
  * library aug-cc-pvtz
end
scf
  doublet
end
qmd
  nstep_nucl numero_pasos
  dt_nucl intervalo_tiempo
  com_step 200
  thermostat none
  linear
  print_xyz 1
end
task scf qmd
```

h2+.ref

```
echo
start molecule
charge 1
geometry units anstrong print xyz autosym
H  0.0000000  RA  0.0000000
H  0.0000000  RB  0.0000000
symmetry c1
end
basis
  * library aug-cc-pvtz
end
scf
  doublet; uhf
  vectors output h2+.movecs
end
task scf energy
```

```
dplot
  TITLE MolOrb
  vectors h2+.movecs
  LimitXYZ units au
  -1.5 1.5 200
  -10 10 200
  0.0 0.0 0
  spin total
  gaussian
  orbitals view; 1;01
  output sigmag.cube
end
task dplot
```