



UNIVERSIDAD AUTÓNOMA METROPOLITANA  
IZTAPALAPA

DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA  
DEPARTAMENTO DE QUÍMICA

“MÉTODOS DE LA QUÍMICA CUÁNTICA SOBRE MALLAS”

TESIS PRESENTADA PARA OBTENER EL GRADO DE  
DOCTOR EN CIENCIAS (QUÍMICA)

PRESENTA:

RAYMUNDO HERNÁNDEZ ESPARZA

BAJO LA DIRECCIÓN:

DR. JORGE GARZA OLGUÍN

Ciudad de México

Septiembre 2018



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

# ACTA DE DISERTACIÓN PÚBLICA

No. 00152

Matricula: 2123803076

MÉTODOS DE LA QUÍMICA  
CUÁNTICA SOBRE MALLAS

En la Ciudad de México, se presentaron a las 14:00 horas del día 27 del mes de septiembre del año 2018 en la Unidad Iztapalapa de la Universidad Autónoma Metropolitana, los suscritos miembros del jurado:

DR. ALBERTO MARCIAL VELA AMIEVA  
DR. JORGE GARZA OLGUIN  
DRA. MYRNA HERNANDEZ MATUS  
DR. ROBIN PREENJA SAGAR  
DR. MARCELO ENRIQUE GALVAN ESPINOSA

Bajo la Presidencia del primero y con carácter de Secretario el último, se reunieron a la presentación de la Disertación Pública cuya denominación aparece al margen, para la obtención del grado de:

DOCTOR EN CIENCIAS (QUIMICA)  
DE: RAYMUNDO HERNANDEZ ESPARZA

y de acuerdo con el artículo 78 fracción IV del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana, los miembros del jurado resolvieron:

*Aprobar*

Acto continuo, el presidente del jurado comunicó al interesado el resultado de la evaluación y, en caso aprobatorio, le fue tomada la protesta.



*Raymundo Hernandez Esparza*  
RAYMUNDO HERNANDEZ ESPARZA  
ALUMNO

REVISÓ

*Lic. Julio Cesar de Lara Isassi*  
LIC. JULIO CESAR DE LARA ISASSI  
DIRECTOR DE SISTEMAS ESCOLARES

DIRECTOR DE LA DIVISIÓN DE CBI

*Jesus Alberto Ochoa Tapia*  
DR. JESUS ALBERTO OCHOA TAPIA

PRESIDENTE

*Alberto Marcial Vela Amieva*  
DR. ALBERTO MARCIAL VELA AMIEVA

VOCAL

*Jorge Garza Olguin*  
DR. JORGE GARZA OLGUIN

VOCAL

*Myrna Hernandez Matus*  
CANCELADO

DRA. MYRNA HERNANDEZ MATUS

VOCAL

*Robin Preenja Sagar*  
DR. ROBIN PREENJA SAGAR

SECRETARIO

*Marcelo Enrique Galvan Espinosa*  
DR. MARCELO ENRIQUE GALVAN ESPINOSA



---

*“What I cannot create  
I do not understand”*

*Richard Feynman*

*“De Iztapalapa  
para el mundo”*

*Los Ángeles Azules*

El examen de grado se llevó a cabo el día 27 de  
septiembre de 2018 a las 14:00 horas en la  
Universidad Autónoma Metropolitana - Iztapalapa

El jurado estuvo formado por los siguientes sinodales

Dr. Alberto M. Vela Amieva	Presidente
Dr. Robin Preenja Sagar	Vocal
Dr. Jorge Garza Olguín	Vocal
Dra. Myrna Hernández Matus	Vocal
Dr. Marcelo E. Galván Espinosa	Secretario





---

*Con todo cariño a mis padres:  
Alma y Reyes, por ser parte  
importante de mi vida, por  
todo su apoyo y comprensión*

---



---

El presente trabajo de investigación se llevó a cabo en el área de Físicoquímica Teórica perteneciente al Departamento de Química de la Universidad Autónoma Metropolitana - Iztapalapa (UAM-I), bajo la dirección del Dr. Jorge Garza Olguín, y con el apoyo otorgado por el Consejo Nacional de Ciencia y Tecnología (Número de registro: 283251).

De este trabajo se han generado las siguientes publicaciones:

- R. Hernández-Esparza, S. M. Mejía-Chica, A. D. Zapata-Escobar, A. Guevara-García, A. Martínez-Melchor, J. M. Hernández-Pérez, R. Vargas, J. Garza. **Grid-based algorithm to search critical points, in the electron density, accelerated by graphics processing units.** *Journal of Computational Chemistry*, **35**, 2272-2278, (2014).
- R. Hernández-Esparza, A. D. Zapata-Escobar, A. Martínez-Melchor, M. C. Ortega-Moo, R. Vargas, J. Garza. **Química cuántica, supercómputo y tarjetas gráficas (GPUs).** *ContactoS*, **101**, 17-28, (2016).
- P. López-Albarrán, A. Pizzi, P. Navarro-Santos, R. Hernández-Esparza, J. Garza. **Oligolignols within lignin-adhesive formulations drive their Young's modulus: A ReaxFF-MD study.** *International Journal of Adhesion and Adhesives*, **78**, 227-233, (2017).
- R. Hernández-Esparza, A. Vázquez-Mayagoitia, L. A. Soriano-Agueda, R. Vargas, J. Garza. **GPUs as boosters to analyze scalar and vector fields in quantum chemistry.** *International Journal of Quantum Chemistry*, (2018). En prensa.

Publicaciones generadas en el periodo del doctorado, no incluidas en este trabajo:

- J. Sandoval-Lira, M. Hô, R. Hernández-Esparza, J. C. Ramírez, J. M. Hernández-Pérez. **Characterizing off-diagonal regions of one-electron density matrix.** *Theoretical Chemistry Accounts*, **135**, 1-10, (2016).
- I. P. Zaragoza, L. A. Soriano-Agueda, R. Hernández-Esparza, R. Vargas, J. Garza. **Analyzing ZnO clusters through the density-functional theory.** *Journal of Molecular Modeling*, **24**, 1-13, (2018).

---

Algunos de los resultados fueron difundidos en los siguientes eventos:

- XXV - Escuela Nacional de Optimización y Análisis Numérico, **Cálculo del potencial electrostático molecular sobre unidades de procesamiento gráfico**. Ciudad de México, México, 2015.
- XIV - Reunión Mexicana de Físicoquímica Teórica, **Cálculo del potencial electrostático molecular sobre unidades de procesamiento gráfico**. Tonalá, México, 2015<sup>†</sup>.
- VII - Taller de dinámica y estructura de la materia, **Gpuam: Una herramienta para el análisis de campos escalares y vectoriales derivados de la función de onda**. Ciudad de México, México, 2016.
- XV - Reunión Mexicana de Físicoquímica Teórica, **Estructura de moléculas diatómicas confinadas usando el método de elemento finito**. Mérida, México, 2016<sup>†</sup>.
- II - Physics Postgraduate Students Meeting, **Electronic structure of confined diatomic molecules using the finite element method**. Ciudad de México, México, 2017.

<sup>†</sup> La asistencia a estos eventos fue gracias al apoyo otorgado por la red de Físicoquímica Teórica.



---

# CONTENIDO

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Objetivos . . . . .	6
1.1.1	Objetivo General . . . . .	6
1.1.2	Objetivos Particulares . . . . .	6
<b>2</b>	<b>Marco Teórico</b>	<b>7</b>
2.1	Método de Hartree-Fock . . . . .	8
2.2	Teoría de Funcionales de la Densidad . . . . .	11
2.3	Ecuaciones de Roothaan . . . . .	13
2.4	Métodos Semiempíricos . . . . .	15
2.5	Métodos post Hartree-Fock . . . . .	17
2.5.1	Método de Interacción de Configuraciones . . . . .	17
2.5.2	Método de Perturbaciones . . . . .	18

## *CONTENIDO*

---

2.5.3	Orbitales Naturales . . . . .	20
2.6	Teoría de Átomos en Moléculas . . . . .	21
2.7	Interacciones No Covalentes . . . . .	27
<b>3</b>	<b>MEF, Mallas y GPUs</b>	<b>31</b>
3.1	Método de Elemento Finito . . . . .	31
3.1.1	Formulación del Método de Elemento Finito 1D . . . . .	32
3.1.2	Formulación del Método de Elemento Finito 2D . . . . .	37
3.2	Tipos de Mallas . . . . .	39
3.3	Programación en GPUs . . . . .	41
<b>4</b>	<b>Programas desarrollados</b>	<b>49</b>
4.1	GPUAM . . . . .	50
4.1.1	Tipo de Tareas . . . . .	52
4.1.2	Campos Escalares y Vectoriales . . . . .	53
4.1.3	Análisis de Campos y Búsqueda de Puntos Críticos . . . . .	59
4.1.4	Ejecución de GPUAM . . . . .	63
4.1.5	Implementación y Rendimiento . . . . .	66
4.2	Cube3D . . . . .	69
4.3	Finit1D . . . . .	73
4.3.1	Malla Froese-Fischer . . . . .	76
4.3.2	Malla Chebyshev . . . . .	77

---

4.3.3	Átomos Hidrogenoides y Distintos Confinamientos . . . . .	78
4.3.3.1	Confinamiento Esférico por Paredes Impenetrables . . . . .	79
4.3.3.2	Confinamiento Esférico por Paredes Penetrables . . . . .	80
4.3.3.3	Confinamiento Esférico por un Dieléctrico Continuo . . . . .	81
4.4	Finit2D . . . . .	84
<b>5</b>	<b>Aplicaciones</b>	<b>89</b>
5.1	Oligolinos - celulosa . . . . .	89
5.2	Diamantoide - oro . . . . .	97
5.2.1	Sistema 1: Diamantoide - Guanina . . . . .	99
5.2.2	Sistema 2: Sistema 1 con Oro . . . . .	101
<b>6</b>	<b>Conclusiones</b>	<b>105</b>
<b>A</b>	<b>Armado de Mallas en GPUs</b>	<b>109</b>
<b>B</b>	<b>Matrices Elementales</b>	<b>115</b>
<b>C</b>	<b>Publicaciones</b>	<b>121</b>



---

# ÍNDICE DE TABLAS

2.1	Tipos de puntos críticos no degenerados para un campo escalar en $\mathbb{R}^3$ . . . . .	23
2.2	Clasificación de las interacciones no covalentes con base en el signo del segundo valor propio. . . . .	28
3.1	Variables para poder controlar y manipular los hilos de una malla. . . . .	43
3.2	Funciones implementadas en CUDA para el manejo de memoria. . . . .	46
4.1	Campos y análisis programados en GPUAM, para cada tipo de función de base. . .	51
4.2	Comparación para una serie de polialaninas con $N$ -residuos, evaluando la densidad electrónica sobre distintas arquitecturas. . . . .	68
4.3	Comparación de velocidad entre Multiwfn y Gpuam para distintos sistemas . . . .	69
4.4	Comparación para distintos ordenes de polinomios de interpolación en el MEF. . .	74
4.5	Número de elementos para cada grado del polinomio en la aproximación. . . . .	74

4.6 Comparación de la energía para el estado basal del átomo de hidrógeno, para distintos radios de corte ( $R_c$ ) y un confinamiento de paredes impenetrables. En la segunda columna se muestran resultados usando bases localizadas (tomadas de la referencia [74]) y la tercera columna los resultados de este trabajo. . . . .	79
4.7 Comparación de la energía para el estado basal del átomo de hidrógeno, para distintos radios de corte ( $R_c$ ), y un confinamiento de paredes penetrables ( $U_0 = 0$ ). . .	81
4.8 Comparación energética para el confinamiento de $H_2^+$ , a distintos niveles de $\xi_0$ , la distancia internuclear fue de 2.0 u.a. . . . .	87
5.1 Nomenclatura de este trabajo para los complejos oligolignol-celulosa. . . . .	90
5.2 Caracterización de los enlaces de hidrógeno para el complejo 1. . . . .	93
5.3 Caracterización de los enlaces de hidrógeno para el complejo 2. . . . .	93
5.4 Caracterización de los enlaces de hidrógeno para el complejo 3. . . . .	94
5.5 Caracterización de los enlaces de hidrógeno para el complejo 4. . . . .	94
5.6 Caracterización de los enlaces de hidrógeno para el complejo 5. . . . .	95
5.7 Puntos críticos de enlace en el sistema 1, los datos están en unidades atómicas, la energía (E) en kcal/mol. . . . .	100
5.8 Puntos críticos de enlace en el sistema 2 y la interfaz diamantoide - nucleótido, los datos están en unidades atómicas, la energía (E) en kcal/mol. . . . .	102
5.9 Comparación entre los sistemas 1 y 2, para los puntos críticos encontrados en la interfaz diamantoide-nucleótido, los datos de la columna dos y tres corresponden a densidades electrónicas en el punto crítico, estos datos están en unidades atómicas. . . . .	103
5.10 Puntos críticos de enlace en el sistema 2, y la interfaz nucleótido - oro, los datos están en unidades atómicas, la energía (E) en kcal/mol. . . . .	104

---

# ÍNDICE DE FIGURAS

2.1	<i>Puntos críticos encontrados para la molécula de agua, se muestran los vectores propios, indicando las direcciones de máxima curvatura. . . . .</i>	24
2.2	<i>Elementos topológicos para la molécula de agua, cuencas, trayectorias de enlace, puntos críticos. . . . .</i>	25
2.3	<i>Densidad electrónica y gradiente reducido para interacciones covalentes y no covalentes. . .</i>	27
2.4	<i>Mapas de <math>\rho(\mathbf{r})</math> vs <math>s(\mathbf{r})</math> para dos sistemas modelo. . . . .</i>	28
2.5	<i>Desdoblamiento en las interacciones no covalentes debido, al criterio de <math>\lambda_2</math>. . . . .</i>	28
2.6	<i>Interacciones no covalentes para dos sistemas modelo, en azul las interacciones atractivas, en verde las interacciones de van der Waals y en rojo las interacciones repulsivas. . . . .</i>	29
3.1	<i>Malla unidimensional de 6 nodos (puntos) con 5 elementos <math>\Omega_i</math> y cada uno con tamaño <math>h_i</math>. .</i>	34
3.2	<i>Funciones de forma lineales dentro del elemento <math>\Omega_i</math>. . . . .</i>	34
3.3	<i>Funciones de forma lineales en un dominio discretizado por 5 elementos. Se puede ver una numeración global (roja) y una numeración local (azul). . . . .</i>	34

## ÍNDICE DE FIGURAS

---

3.4	<i>Funciones de forma del elemento <math>\Omega_i</math> y referenciadas al elemento maestro <math>\hat{\Omega}</math>.</i> . . . . .	35
3.5	<i>Solución exacta continua y suave, solución aproximada punteada, con 5 elementos.</i> . . . . .	37
3.6	<i>Elemento triangular <math>\Omega_i</math> en el plano <math>XY</math> y elemento maestro <math>\hat{\Omega}</math> en el plano <math>\xi\eta</math>.</i> . . . . .	39
3.7	<i>Ejemplo de malla no estructurada en un archivo VTK y visualizada en VisIt.</i> . . . . .	41
3.8	<i>Diagrama de una ejecución en un sistema CPU-GPU.</i> . . . . .	42
3.9	<i>Malla para la ejecución de un kernel, se trata de una malla <math>4 \times 2</math> bloques y cada bloque de <math>4 \times 6</math> hilos.</i> . . . . .	44
3.10	<i>Isosuperficie de la función escalar (3.29). Los datos fueron calculados en la GPU.</i> . . . . .	47
4.1	<i>Ejemplo de visualización de campos escalares y vectoriales para la molécula de benceno.</i> . . . . .	52
4.2	<i>Ejemplo de dos campos relacionados con el módulo del gradiente para el sistema <math>8 \cdot (H_2O)^-</math>.</i> . . . . .	54
4.3	<i>Gradiente del orbital HOMO-2, para la molécula de agua, sobre el plano que contiene a los 3 átomos, se grafica también el orbital correspondiente. Visualizado con VisIt.</i> . . . . .	55
4.4	<i>Dos funciones de localización electrónica para el sistema <math>8 \cdot (H_2O)^-</math>.</i> . . . . .	56
4.5	<i>Potencial electrostático para un complejo de ciclodextrina y formamida, <math>\rho(\mathbf{r}) = 0.01</math> u.a.</i> . . . . .	59
4.6	<i>Puntos críticos de la densidad para la molécula de cubano: en amarillo los puntos críticos de enlace (BCP), en verde los puntos críticos de anillo (RCP), en purpura un punto crítico de caja (CCP), las trayectorias de enlace en lila.</i> . . . . .	61
4.7	<i>Puntos críticos del laplaciano para la molécula de agua. Visualizado con el código VMD en amarillo los (3,-1), verde (3,1), morado (3,3), negro (3,-3).</i> . . . . .	62
4.8	<i>Índice de interacciones no covalentes para el dímero de agua.</i> . . . . .	62
4.9	<i>Comparación entre distintas arquitecturas en la determinación de la densidad electrónica.</i> . . . . .	68
4.10	<i>Campos para el dimero de agua determinados usando un esquema numérico.</i> . . . . .	69



---

4.11 Índice de interacciones no covalentes, puntos críticos de enlace y trayectorias de enlace para un cristal de urea. . . . .	71
4.12 Celda unitaria para un sistema oblicuo, pueden verse los ejes que no son ortogonales y los ángulos que forman entre ellos. . . . .	73
4.13 Décimo estado en energía, determinado con polinomios de orden 6 para el oscilador armónico.	75
4.14 Malla uniformemente espaciada para el átomo de hidrógeno, realizada con 20 elementos y polinomios de grado 3. La energía obtenida es de $-0.496$ u.a. . . . .	75
4.15 Malla de Froese-Fischer para el átomo de hidrógeno, realizada con 20 elementos y polinomio de grado 3. La energía obtenida es de $-0.498$ u.a. . . . .	76
4.16 Malla de Chebyshev para el átomo de hidrógeno, realizada con 20 elementos y polinomio de grado 3. La energía obtenida es de $-0.500$ u.a. . . . .	76
4.17 Puntos dados por Froese-Fisher para distintos $N$ . . . . .	77
4.18 Puntos dados por Chebyshev para distintos $N$ . . . . .	78
4.19 Estado basal para el átomo de hidrógeno confinado por paredes penetrables y potencial $U_0 = 0$ en $R_c = 2.04918$ u.a. . . . .	80
4.20 Energía del estado basal para el átomo de hidrógeno, a distintos radios, los círculos corresponden a una constante dieléctrica de 2.1, mientras los triángulos a la constante dieléctrica de 80.0, las figuras llenas corresponden a datos determinados via MEF mientras las figuras vacías a su contraparte exacta. . . . .	82
4.21 Valor esperado $\langle r \rangle$ para el átomo de hidrógeno a distintos radios, los círculos corresponden a una constante dieléctrica de 2.1, mientras los triángulos a la constante dieléctrica de 80.0, la línea horizontal corresponde al valor esperado para el caso libre. . . . .	83
4.22 Función de onda y densidad electrónica para el estado basal para del átomo de hidrógeno confinado por un dieléctrico, con constante dieléctrica de 80.0. . . . .	83
4.23 Triangulación de Delaunay para 12 puntos. . . . .	84
4.24 Proceso de triangulación de Delaunay. . . . .	86

## ÍNDICE DE FIGURAS

---

4.25	Último paso de la triangulación. . . . .	86
4.26	Densidad para los primeros dos estados de la molécula ion $H_2^+$ , usando dos metodologías distintas, para un confinamiento elipsoidal de $\xi_0 = 1.9934$ u.a. . . . .	87
4.27	Densidad para los primeros dos estados de la molécula ion $H_2^+$ , usando dos metodologías distintas, para un confinamiento elipsoidal de $\xi_0 = 1.3621$ u.a. . . . .	88
5.1	Estructura de dos lignoles: izquierda coniferol, derecha sinapol. . . . .	90
5.2	Trilignoles estudiados. . . . .	91
5.3	Dilignoles estudiados. . . . .	91
5.4	Análisis de interacciones no covalentes para los complejos oligolignol- celulosa, las zonas verdes corresponden a interacciones de van der Waals, los pequeños discos azulados corresponden a interacciones fuerte tipo puente de hidrógeno, mientras que las zonas rojas indican interacciones no enlazantes. Las esferas en color gris sugieren carbono, rojo para oxígeno y blanco para hidrógeno. . . . .	92
5.5	Clasificación de los enlaces de hidrógeno en los complejos oligolignol-celulosa. . . . .	95
5.6	Análisis de átomos en moléculas para las interacciones intermoleculares. . . . .	96
5.7	Estructura Au-Diamantoide ··· Nucleótido de guanina - Au, vista lateral del sistema. Las esferas grises representan átomos de carbono, las azules nitrógeno, rojas oxígeno, blancas hidrógeno, amarilla azufre y naranja oro. . . . .	97
5.8	Estructura Diamantoide ··· Nucleótido de guanina. Las esferas grises representan átomos de carbono, las azules nitrógeno, rojas oxígeno, blancas hidrógeno y amarilla azufre. . . . .	99
5.9	Índice de interacciones no covalentes para diamantoide - nucleótido. . . . .	99
5.10	Puntos críticos y trayectorias de enlace entre el diamantoide y el nucleótido, los puntos críticos se muestran en amarillo, mientras que la trayectoria de enlace en azul, se encuentran ordenados de acuerdo a la densidad en el punto crítico. . . . .	100
5.11	Estructura Au - Diamantoide ··· Nucleótido ··· Au. . . . .	101

- 
- 5.12 *Puntos críticos y trayectorias de enlace entre el diamantoide y el nucleótido, los puntos críticos se muestran en amarillo, mientras que la trayectoria de enlace para la interfaz 1 en azul.* 102
- 5.13 *Puntos críticos y trayectorias de enlace entre el diamantoide y el nucleótido, los puntos críticos se muestran en amarillo, mientras que la trayectoria de enlace para la interfaz 3 en azul.* 103



---

## RESUMEN

El presente trabajo tuvo como base la exploración de algunos métodos de la química cuántica y su implementación computacional sobre mallas. Para ello se exponen algunos detalles que muestran la necesidad de cómputo que procese gran cantidad de datos y haga millones de operaciones en una cantidad de tiempo razonable, para este fin se discute como las tarjetas gráficas (GPU por el término en inglés *Graphics Processing Unit*) han incursionado en el cómputo, como aceleradores de procesos intrínsecamente paralelizables.

Siguiendo esta línea dada por los métodos basados en mallas se han desarrollado tres códigos, dos de ellos para su uso en el análisis de las interacciones químicas, aplicables a átomos, moléculas y sistemas extendidos. Y un tercero código explorando el método de elemento finito (MEF) para su uso en la determinación de la estructura electrónica de sistemas confinados espacialmente. Finalmente se presentan dos aplicaciones del código GPUAM sobre tarjetas gráficas para el análisis de interacciones débiles.



---

---

# CAPÍTULO 1

---

## INTRODUCCIÓN

Desde el desarrollo de las ciencias, hablando en el contexto de las ciencias exactas y naturales, siempre ha existido una dicotomía al momento de estudiarlas y abordarlas. Por un lado, tenemos la parte teórica, cuyo fundamento matemático es la piedra angular y que nos permite estudiar los fenómenos generalmente en la forma de una ecuación diferencial. Complementando esta dicotomía tenemos la parte experimental, donde el investigador trata de tener el mayor control posible sobre el experimento, y de esta forma cuantificar algún observable de interés. Esta íntima relación teoría - experimento ha llevado a muchas de las ciencias exactas y naturales a tener la capacidad de explicar y predecir fenómenos cada vez más complejos.

No fue hasta la mitad del siglo XX, que esta relación teoría - experimento vino a ser complementada desde otro ángulo, un plano que puede pensarse como un intermedio entre el mundo teórico y el mundo experimental: la simulación computacional [1].

Los primeros modelos computacionales surgen con la carrera armamentista, siendo su primer nicho, la criptografía y simulación de trayectorias balísticas. De este periodo datan las máquinas Colossus (1943), ENIAC (1947) y EDVAC(1949). Cabe resaltar que el poder de cómputo en esta época, al menos para la ENIAC (*Electronic Numerical Integrator and Calculator*), era de 5000 sumas o 500 multiplicaciones por segundo [2] y el volumen de muchas de estas

computadoras era comparable al de una casa pequeña.

Hoy en día, la simulación computacional goza de una mejor situación, empleándose en diversos y variados campos, como puede ser la ya mencionada balística computacional, la predicción del clima, la dinámica de fluidos y la química teórica, por mencionar sólo un pequeño grupo del gran abanico existente. Es justo en el marco de la química teórica donde este trabajo se desarrolla, más precisamente en el campo de la estructura electrónica.

La química teórica surge con las ideas de Douglas R. Hartree (1927) y Vladimir A. Fock (1930), tomando la teoría cuántica de Schrödinger para la descripción de los electrones a partir de la función de onda. Sin embargo, es hasta 1950 cuando las computadoras consiguen el poder necesario para efectuar cálculos del tipo Hartree-Fock (HF). Por otro lado, surgía también una teoría basada, no en la función de onda sino en la densidad electrónica; con los trabajos de Llewellyn Thomas (1927) [3] y Enrico Fermi (1928) [4], más adelante sería formalizada por Pierre Hohenberg y Walter Kohn [5], en la teoría de los funcionales de la densidad (TFD), la cual, años después, sería ampliamente usada en la formulación desarrollada por Walter Kohn y Lu J. Sham (KS).

En la actualidad, algunos de los principales enigmas que intenta resolver la química teórica son: entender la naturaleza del enlace químico, tener una comprensión de la relación existente entre la estructura y las propiedades físicas y químicas. La forma actual de abordar estas interrogantes se ha hecho desde la trinchera de la estructura electrónica donde el caballo de batalla es, sin duda, la TFD en la formulación de KS y, de forma ocasional el método de HF. En ambos métodos, los orbitales que construyen la función de onda (o densidad electrónica en KS) se proponen como una combinación de funciones conocidas, si  $N$  es el número de funciones usadas, el método de HF y KS escalan como  $N^4$ . Es decir que si tenemos un sistema, por ejemplo, la molécula de agua, formada por 3 átomos, y usamos sólo una función por cada electrón (10 funciones para la molécula de agua), HF y KS tienen un escalamiento de  $10^4$ . Si crecemos el sistema a uno un poco más grande, por ejemplo benceno (42 electrones) e igual con una sola función de base por cada electrón, éste escalará a  $42^4$ . Haciendo un poco de aritmética, se puede apreciar un costo 311 veces más que una molécula de agua.

Queda claro el esfuerzo computacional que hay que realizar conforme el tamaño del sistema se incrementa. Afortunadamente nos encontramos en una etapa de la historia donde estos cálculos son frecuentemente realizados, ya sea en estaciones de trabajo, o en supercomputadoras armadas a partir de múltiples nodos interconectados. Abriendo un paréntesis para situarnos en el estado actual de los sistemas informáticos, y tomando como referencia las supercomputa-



doras más rápidas (hasta febrero 2018), podemos citar las primeras cinco [6]:

- **Sunway TaihuLight** China, con 10 649 600 núcleos, 1 310 720 GB de memoria RAM, procesadores Sunway SW26010 260C 1.45 GHz, tiene rendimiento de 93 014.6 TFlop/s<sup>1</sup>.
- **Tianhe-2 (MilkyWay-2)** China, con 3 120 000 núcleos, 1 024 000 GB de memoria RAM, procesadores Intel Xeon E5-2692v2 12C 2.2 GHz y coprocesadores Intel Xeon Phi 31S1P, tiene un rendimiento de 33 862.7 TFlop/s.
- **Piz Daint** Suiza, con 361 760 núcleos, 340 480 GB de memoria RAM, procesadores Intel Xeon E5-2690v3 12C 2.6 GHz y tarjetas gráficas Nvidia Tesla P100, tiene un rendimiento de 19 590 TFlop/s.
- **Gyokou** Japón, con 19 860 000 núcleos, 575 552 GB de memoria RAM, procesadores Intel Xeon D-1571 16C 1.3 GHz, y coprocesadores PEZY-SC2 700Mhz, tiene un rendimiento de 19 135.8 TFlop/s.
- **Titan** Estados Unidos, con 560 640 núcleos, 710 144 GB de memoria RAM, procesadores AMD Opteron 6274 16C 2.2 GHz y tarjetas gráficas Nvidia K20x, tiene un rendimiento de 17 590 TFlop/s.

Se observa que salvo la primera, las restantes cuentan con aceleradores en su arquitectura, los cuales son coprocesadores o tarjetas gráficas, destacándose las de la compañía Nvidia con sus tarjetas gráficas Tesla P100 y K20x. Una revisión más detallada sobre los procesadores y aceleradores, nos advierte que el procesador que más se emplea en la implementación de supercomputadoras es Intel Xeon E5 en sus diversas versiones con el 92.4% del total. Mientras que los aceleradores están polarizados hacia los ofrecidos por Nvidia, donde el mayor número es para la familia Pascal con 50 supercomputadoras, y 34 divididas entre la familia Kepler, Fermi y Volta. El otro acelerador que tiene presencia es el ofrecido por Intel, el coprocesador Xeon Phi, con 10 del total.

Toda esta infraestructura computacional, lleva el interés de los científicos a sistemas con una complejidad cada vez mayor, por ejemplo sistemas proteínicos, máquinas moleculares, cadenas de ADN-ARN, estudiados desde la perspectiva de la estructura electrónica, para lo cual se pueden usar métodos híbridos que la mezclan con la mecánica molecular. O basados en

---

<sup>1</sup>Flop/s es un acrónimo para *floating-point operations per second* cuya traducción es Operaciones de punto flotante por segundo, es una medida del rendimiento teórico de una computadora. Se suele usar prefijos de SI, donde TFlop/s es equivalente a  $10^{12}$  Flop/s.

esquemas *divide et impera*, que buscan segmentar el sistema total, tal es el caso del método FMO [7] (por sus siglas en inglés para orbitales moleculares fragmentados), o incluso en rescatar metodologías usadas hace años y mejorarlas, como ha sido el caso de los métodos semiempíricos, que encuentran un resurgimiento en la estructura química de sistemas biológicos [8–11].

Es importante advertir que, con el avance en el hardware (procesadores multinúcleo, tarjetas gráficas y/o coprocesadores) debe de manifestarse también un cambio en el software, ya que poca o nula ventaja se tendrá si se usan los mismos algoritmos en arquitecturas distintas. Por ello, para obtener el mayor beneficio es necesario cambiar los paradigmas de programación, usar programación concurrente, que permita utilizar la mayor cantidad de núcleos de cálculo; en esta lógica, el modelo de programación suele ser uno que involucre gran cantidad de datos. Existen dos técnicas de programación en paralelo, por un lado la de instrucciones simples con datos múltiples (SIMD del inglés *Single Instruction Multiple Data*), que permite paralelizar a nivel de datos, pero con pocas instrucciones; la otra, más general, que involucra paralelismo a nivel de dato y de instrucción es la de instrucciones múltiples con datos múltiples (MIMD del inglés *Multiple Instruction Multiple Data*).

Retornando a los sistemas químicos, una cantidad muy importante en la química es la densidad electrónica, la cual es la base de la TFD y de la teoría de átomos en moléculas (AEM), esta última propuesta por Richard F. W. Bader, a diferencia de la función de onda, la densidad es una cantidad que reside en el mundo tridimensional real y es un observable. Como tal, podemos acceder a ella por medio de experimentos ya sea difracción de rayos X o difracción de neutrones o, en nuestro caso particular, a partir de cálculos de estructura electrónica.

Los cálculos de estructura electrónica nos brindan dos métodos para determinar la densidad electrónica: (1) con base en la expansión orbital de bases conocidas o (2) directamente determinándola en una malla tridimensional. Dentro del último grupo, el método de elemento finito figura como una metodología robusta, pero poco usada, para realizar cálculos de estructura electrónica y obtener la densidad electrónica, sin el uso de funciones orbitales previamente optimizadas.

La expresión para evaluar la densidad electrónica en términos de los orbitales moleculares  $\phi_i$  y sus ocupaciones  $\omega_i$ , es:

$$\rho(\mathbf{r}) = \sum_i \omega_i \phi_i^*(\mathbf{r}) \phi_i(\mathbf{r}), \quad (1.1)$$

donde  $\mathbf{r}$  corresponde a un vector en tres dimensiones  $\mathbf{r} = x\hat{i} + y\hat{j} + z\hat{k}$ . Para evaluarla se genera una malla en tres dimensiones, donde cada vector de posición ( $\mathbf{r}$ ) es independiente del resto,

razón por la cual este problema se puede paralelizar en el modo SIMD.

Ampliando el ejercicio de evaluar la densidad electrónica usando un esquema SIMD sobre unidades aceleradoras tipo GPU, es posible implementar la búsqueda y caracterización de puntos críticos de la densidad, siguiendo las ideas propuestas por Bader y diversos autores que han extendido las ideas de AEM en lo que hoy se conoce como método topológico.

Es importante mencionar que hasta la escritura de esta tesis no se sabe de algún software que haga uso de tarjetas gráficas para la evaluación de campos escalares, vectoriales y búsqueda de puntos críticos, en arquitecturas híbridas CPU-GPU. Así mismo, citar que la implementación y puesta a punto de programas para su uso en sistemas químicos nos confiere cierta autonomía con respecto a otros grupos, creando software propio que responda a los requerimientos particulares de cada investigación.

El presente escrito se encuentra dividido en seis capítulos. En este primero se ha tratado de poner en contexto los métodos que existen en la química teórica, así como también, sobre el uso de herramientas computacionales en el escenario de la química.

En el segundo capítulo se realiza una revisión de los métodos y conceptos usuales en la estructura electrónica. El tercer capítulo acerca al lector a los métodos basados en mallas y, finalmente, se enuncia su implementación en unidades de procesamiento gráfico. En el capítulo cuarto, se exponen los programas desarrollados en este trabajo: GPUAM, CUBE3D y FINIT1-2D. Es en el capítulo quinto donde se mencionan algunas aplicaciones realizadas principalmente con GPUAM con la finalidad de caracterizar interacciones químicas. Por último, en el capítulo sexto, se enuncian las conclusiones de este trabajo.

## 1.1 Objetivos

Utilizar tarjetas gráficas (GPUS) en la química teórica para analizar sistemas químicos en una escala de tiempo razonable.

### 1.1.1 Objetivo General

Explorar los métodos basados en mallas y su implementación sobre arquitecturas CPU-GPU, para la evaluación de campos escalares, vectoriales y búsqueda de puntos críticos de la densidad electrónica obtenida a partir de cálculos de estructura electrónica.

### 1.1.2 Objetivos Particulares

- Evaluar la densidad electrónica y otros campos escalares derivados a partir de la información de la estructura electrónica.
- Encontrar la forma más adecuada de programar las mallas tridimensionales utilizando las tarjetas gráficas.
- Realizar la búsqueda de puntos críticos de la densidad, dentro del marco de la teoría de átomos en moléculas propuesta por Richard Bader, y caracterizar el enlace químico.
- Aplicar el método de elemento finito para sistemas químicos delimitados espacialmente.

---

---

## CAPÍTULO 2

---

### MARCO TEÓRICO

La mecánica cuántica aplicada a la química es una poderosa herramienta para entender la estructura interna de átomos, moléculas y sólidos. Con ella se puede conocer las propiedades físicas y químicas de compuestos existentes pero que experimentalmente son un reto cuantificar así como también conocer las propiedades de compuestos que aún no se sintetizan. Por esta razón, es un importante nicho de investigación actual.

Hablar de química teórica implica también hablar de química computacional, donde la principal herramienta no son los matrices sino las computadoras, esto es debido a que los sistemas que son de importancia en química, escapan de la habilidad numérica de los seres humanos y resulta un problema intrínsecamente apto para ser resuelto por computadoras. Para profundizar un poco, se hará un pequeño resumen de algunos métodos que se han empleado a lo largo de este trabajo.

En mecánica cuántica, la caracterización de un sistema, puede hacerse al resolver la siguiente ecuación:

$$\hat{\mathcal{H}} \Psi(\mathbf{r}, t) = i\hbar \frac{\partial}{\partial t} \Psi(\mathbf{r}, t), \quad (2.1)$$

propuesta por el físico Erwin Schrödinger y que es conocida como ecuación de onda [12, 13]. En la mayoría de las aplicaciones de química cuántica usamos la ecuación en su forma indepen-

diente del tiempo, esto es así por que la mayoría de los estados que nos interesan son estacionarios. En este caso, la función de onda  $\Psi(\mathbf{r}, t)$  puede separarse en dos funciones, una dependiente sólo del espacio y otra dependiente del tiempo:

$$\Psi(\mathbf{r}, t) = \Psi(\mathbf{r}) e^{-iEt/\hbar}. \quad (2.2)$$

Esto nos conduce a la ecuación de onda independiente del tiempo,

$$\hat{\mathcal{H}} \Psi(\mathbf{r}) = E \Psi(\mathbf{r}). \quad (2.3)$$

Al resolver esta ecuación de valores propios, obtendremos tanto la función de onda  $\Psi$  como la energía  $E$  asociada al sistema. Esta ecuación tiene solución exacta para pocos sistemas, pero en la práctica, los sistemas de interés son más complejos, esto tiene como consecuencia que la función de onda se obtiene tras recurrir a ciertas aproximaciones.

## 2.1 Método de Hartree-Fock

Si las aproximaciones a la ecuación de Schrödinger son generadas sin hacer referencia a datos experimentales, los métodos se denominan *ab initio*. El método Hartree-Fock (HF) es un ejemplo de este tipo y es considerado el punto de partida para tratamientos más precisos (post Hartree-Fock) o para realizar aproximaciones extras (métodos semiempíricos) [14].

El principio de Pauli establece que “la función de onda de un sistema de electrones debe ser antisimétrica con respecto al intercambio de dos electrones” [12]. Una forma de cumplir con este principio es proponer a la función de onda como un determinante de Slater, formado por  $N$  funciones monoeléctricas llamadas espín-orbitales  $\phi_j(\mathbf{x}_i)$ , donde  $\mathbf{x}_i \equiv \mathbf{r}_i \sigma_i$ . La función tiene la siguiente forma:

$$\Psi_{DS}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \phi_1(\mathbf{x}_1) & \phi_2(\mathbf{x}_1) & \cdots & \phi_N(\mathbf{x}_1) \\ \phi_1(\mathbf{x}_2) & \phi_2(\mathbf{x}_2) & \cdots & \phi_N(\mathbf{x}_2) \\ \vdots & \vdots & & \vdots \\ \phi_1(\mathbf{x}_N) & \phi_2(\mathbf{x}_N) & \cdots & \phi_N(\mathbf{x}_N) \end{vmatrix} \quad (2.4)$$

Al expresar la función de onda de esta manera estamos asegurando que sea antisimétrica con respecto al intercambio de las posiciones de dos electrones.

Para evaluar la energía de un sólo determinante de Slater resulta conveniente escribirlo como la suma de permutaciones sobre los elementos de la diagonal,

$$\Psi_{DS}(\{\mathbf{x}_i\}) = \hat{\mathcal{A}}[\phi_1(\mathbf{x}_1)\phi_2(\mathbf{x}_2)\cdots\phi_N(\mathbf{x}_N)] = \hat{\mathcal{A}}\Pi, \quad (2.5)$$

donde  $\Pi$  corresponde al producto de la diagonal del determinante de Slater y el operador  $\hat{\mathcal{A}}$  tiene la forma:

$$\hat{\mathcal{A}} = \frac{1}{\sqrt{N!}} \sum_{p=0}^{N-1} (-1)^p \hat{\mathcal{P}} = \frac{1}{\sqrt{N!}} \left[ 1 - \sum_{ij} \hat{\mathcal{P}}_{ij} + \sum_{ijk} \hat{\mathcal{P}}_{ijk} - \cdots \right]. \quad (2.6)$$

Por otro lado, el hamiltoniano electrónico en la aproximación de Born-Oppenheimer y usando unidades atómicas<sup>1</sup> se expresa como:

$$\hat{\mathcal{H}} = -\frac{1}{2} \sum_i \nabla_i^2 - \sum_i \sum_a \frac{Z_a}{|\mathbf{R}_a - \mathbf{r}_i|} + \sum_i \sum_{j>i} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \quad (2.7)$$

donde el primer término corresponde a la energía cinética de los electrones, el segundo a la interacción núcleo-electrón y el último a la repulsión electrónica. El primer y segundo término, que dependen de las coordenadas de un electrón, se agrupan en el operador monoeléctrico  $\hat{h}_i$ ,

$$\hat{h}_i = -\frac{1}{2} \nabla_i^2 - \sum_a \frac{Z_a}{|\mathbf{R}_a - \mathbf{r}_i|}. \quad (2.8)$$

Mientras que la repulsión entre dos electrones tiene la forma,

$$g_{ij} = \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}. \quad (2.9)$$

Usando estos operadores expresamos el hamiltoniano en términos de  $\hat{h}_i$  y  $g_{ij}$ ,

$$\hat{\mathcal{H}} = \sum_i \hat{h}_i + \sum_i \sum_{j>i} g_{ij}. \quad (2.10)$$

Con estos dos elementos previamente expuestos, la aproximación de la función de onda y la nueva expresión del hamiltoniano, la energía es escrita de la siguiente forma:

$$E = \langle \hat{\mathcal{A}}\Pi | \hat{\mathcal{H}} | \hat{\mathcal{A}}\Pi \rangle, \quad (2.11)$$

<sup>1</sup>Las unidades atómicas son aquellas donde: la constante de Planck reducida ( $\hbar$ ), la masa ( $m_e$ ) del electrón, la carga ( $e$ ) del electrón y la constante de la ley de Coulomb ( $1/(4\pi\epsilon_0)$ ) son iguales a la unidad.

$$E = \sqrt{N!} \langle \Pi | \hat{\mathcal{H}} | \hat{\mathcal{A}} \Pi \rangle, \quad (2.12)$$

$$E = \sum_p (-1)^p \langle \Pi | \hat{\mathcal{H}} | \hat{\mathcal{P}} \Pi \rangle. \quad (2.13)$$

Evaluando para cada elemento del hamiltoniano tenemos:

$$\langle \Pi | \hat{h}_i | \Pi \rangle = \langle \phi_i(1) | \hat{h}_i | \phi_i(1) \rangle = h_i, \quad (2.14)$$

donde  $h_i$  es la llamada integral monoeléctronica. Para el operador  $g_{ij}$  tenemos dos casos, primero cuando el orden de la permutación vale 0,

$$\langle \Pi | g_{12} | \Pi \rangle = \langle \phi_1(1) \phi_2(2) | g_{12} | \phi_1(1) \phi_2(2) \rangle = J_{12}, \quad (2.15)$$

a esta cantidad  $J_{12}$  se le llama integral coulombica y puede ser interpretada como la interacción entre dos distribuciones de carga. El otro caso resulta cuando el orden de la permutación equivale a 1 y se intercambian los electrones 1 y 2:

$$\langle \Pi | g_{12} | \hat{P}_{12} \Pi \rangle = \langle \phi_1(1) \phi_2(2) | g_{12} | \phi_2(1) \phi_1(2) \rangle = K_{12}, \quad (2.16)$$

esta integral es denominada de intercambio y es una consecuencia inmediata del principio de antisimetría de Pauli. Con estos elementos, la energía se obtiene como la suma de dichas integrales:

$$E = \sum_i^N h_i + \sum_i^N \sum_{j>i}^N (J_{ij} - K_{ij}). \quad (2.17)$$

Las últimas dos integrales pueden ser escritas en forma de operador que actuará sobre una función  $|\phi_j(2)\rangle$  quedando expresadas:

$$\hat{\mathcal{J}}_i |\phi_j(2)\rangle = \langle \phi_i(1) | g_{12} | \phi_i(1) \rangle |\phi_j(2)\rangle, \quad (2.18)$$

$$\hat{\mathcal{K}}_i |\phi_j(2)\rangle = \langle \phi_i(1) | g_{12} | \phi_j(2) \rangle |\phi_i(1)\rangle. \quad (2.19)$$

Al combinar estos operadores en uno sólo tenemos el operador de Fock,

$$\hat{\mathcal{F}}_i = \hat{h}_i + \sum_j^N (\hat{\mathcal{J}}_j - \hat{\mathcal{K}}_j). \quad (2.20)$$

Los valores esperados del operador de Fock son las energías de los orbitales, matemáticamente son los multiplicadores de Lagrange asociados a la condición de orto-normalidad de dichos



orbitales, a estos orbitales se les llama orbitales canónicos.

$$\varepsilon_i = \langle \phi_i | \hat{\mathcal{F}}_i | \phi_i \rangle = h_i + \sum_j^N (J_{ij} - K_{ij}). \quad (2.21)$$

La energía para un sistema de  $N$  electrones está determinada a partir de la energía de los orbitales y de las integrales coulombica y de intercambio,

$$E_N = \sum_i^N \varepsilon_i - \frac{1}{2} \sum_{ij}^N (J_{ij} - K_{ij}). \quad (2.22)$$

## 2.2 Teoría de Funcionales de la Densidad

Para un sistema pequeño, por ejemplo el  $H_2O$  tenemos un total de diez electrones, ocho por el átomo de oxígeno y dos más por cada átomo de hidrógeno. Si queremos determinar la función de onda para este sistema necesitamos de tres coordenadas por cada electrón, lo que hace un total de 30 coordenadas, sin tomar en cuenta el espín. Es evidente que para sistemas moleculares de mayor tamaño, la cantidad de variables crece pasmosamente. Es por ello que se han desarrollado alternativas para realizar cálculos mecanocuánticos, uno de ellos es la Teoría de Funcionales de la Densidad (TFD).

El origen de esta teoría puede encontrarse en la teoría estadística de Thomas [3] y Fermi [4] en 1927 y 1928, este trabajo toma en cuenta un gas de electrones homogéneo y cuya densidad es la misma en todos los puntos del espacio.

Fue en 1964 cuando Hohenberg y Kohn demostraron dos teoremas, ubicando al modelo de Thomas y Fermi como una aproximación de una teoría más general, desarrollándose la teoría de funcionales de la densidad (TFD) que toma a la densidad electrónica como variable principal. Los teoremas que propusieron Hohenberg y Kohn [5] son:

**Teorema 1** *Para cualquier sistema de partículas interactuantes en un potencial externo, el potencial externo es determinado unívocamente por la densidad de partículas del estado base.*

**Teorema 2** *Para cualquier potencial externo particular, la energía del estado base del sistema, que es un funcional de la densidad, es el valor mínimo absoluto de este funcional y la densidad que minimiza el funcional es la densidad exacta del estado base.*

Donde el potencial externo es aquel generado por los núcleos del sistema:

$$v(\mathbf{r}_i) = -\sum_a \frac{Z_a}{|\mathbf{R}_a - \mathbf{r}_i|}. \quad (2.23)$$

A partir del primer teorema podemos escribir la energía para un sistema con respecto a la densidad electrónica expresada en la siguiente ecuación:

$$E[\rho] = T[\rho] + V_{ne}[\rho] + V_{ee}[\rho]. \quad (2.24)$$

O bien, a partir de

$$E[\rho] = \int \rho(\mathbf{r}) v(\mathbf{r}) d\mathbf{r} + F_{HK}[\rho], \quad (2.25)$$

donde el último término  $F_{HK}[\rho]$  es llamado funcional de Hohenberg y Kohn, el cual contiene la suma del funcional de energía cinética más el funcional de energía potencial electrón-electrón, siendo éste la suma de los términos coulombico, y de intercambio y correlación de las interacciones electrón-electrón.

$$F_{HK}[\rho] = T[\rho] + V_{ee}[\rho]. \quad (2.26)$$

El funcional de Hohenberg y Kohn (2.26) es un funcional universal que solamente depende de la densidad de un sistema particular, sin embargo, la forma exacta de este funcional es desconocida.

Kohn y Sham recurrieron a un determinante de Slater para modelar el funcional de energía cinética,

$$T_S[\rho] = \langle \Psi_S | \sum_i^N -\frac{1}{2} \nabla_i^2 | \Psi_S \rangle, \quad (2.27)$$

donde  $\Psi_S$  corresponde a la función de onda para un sistema de partículas no interactuantes. Tomando esto en cuenta, la ecuación (2.26) puede reescribirse de la siguiente forma, siendo  $E_{xc}$  la energía de intercambio y correlación:

$$F[\rho] = T_S[\rho] + J[\rho] + E_{xc}[\rho], \quad (2.28)$$

con esta aproximación el funcional universal queda expresado como:

$$F[\rho] = T_S[\rho] + \frac{1}{2} \int \int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} d\mathbf{r}d\mathbf{r}' + E_{xc}[\rho]. \quad (2.29)$$

La energía total del sistema queda especificada por:

$$E[\rho] = T_S[\rho] + \int \rho(\mathbf{r}) v(\mathbf{r}) d\mathbf{r} + \frac{1}{2} \int \int \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}d\mathbf{r}' + E_{xc}[\rho]. \quad (2.30)$$

De la ecuación (2.30) son conocidos todos los términos excepto el último  $E_{xc}[\rho]$ .

Si bien la variable principal en esta metodología es la densidad electrónica, esta es determinada por la expresión siguiente,

$$\rho(\mathbf{r}) = \sum_{i=1}^{Occ} n_i \phi_i^{KS*}(\mathbf{r}) \phi_i^{KS}(\mathbf{r}), \quad (2.31)$$

donde  $\phi^{KS}$ , es el conjunto de orbitales que se obtienen al minimizar la energía con respecto a los orbitales manteniendo la condición de ortonormalidad de estos [15], lo cual lleva a la ecuación

$$-\frac{1}{2} \nabla^2 \phi_i^{KS}(\mathbf{r}) + v_{eff}(\mathbf{r}) \phi_i^{KS}(\mathbf{r}) = \varepsilon_i \phi_i(\mathbf{r}), \quad (2.32)$$

que son justamente las ecuaciones de Kohn-Sham, donde el potencial efectivo  $v_{eff}$ , esta dado por

$$v_{eff}(\mathbf{r}) = v(\mathbf{r}) + \frac{\delta J[\rho(\mathbf{r})]}{\delta \rho(\mathbf{r})} + \frac{\delta E_{xc}[\rho(\mathbf{r})]}{\delta \rho(\mathbf{r})}. \quad (2.33)$$

En la ecuación anterior el primer término corresponde al potencial núcleo-electrón, el segundo es el potencial coulombico,

$$\frac{\delta J[\rho(\mathbf{r})]}{\delta \rho(\mathbf{r})} = \int \frac{\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r}', \quad (2.34)$$

y el último término, a lo que conocemos como potencial de intercambio y correlación,

$$v_{xc}(\mathbf{r}) = \frac{\delta E_{xc}[\rho(\mathbf{r})]}{\delta \rho(\mathbf{r})}. \quad (2.35)$$

## 2.3 Ecuaciones de Roothaan

Una forma de resolver las ecuaciones de HF y de KS es a través del método de Roothaan. En este caso, se expande el orbital molecular ( $\phi_i$ ) en términos de un conjunto de funciones base conocidas,  $\{\varphi_\nu\}$ ,

$$|\phi_i\rangle = \sum_{\nu}^K c_{\nu}^i |\varphi_{\nu}\rangle. \quad (2.36)$$

Aplicando el operador de Fock sobre este conjunto de funciones tenemos:

$$\hat{F}_i \sum_{\nu}^K c_{\nu}^i |\varphi_{\nu}\rangle = \varepsilon_i \sum_{\nu}^K c_{\nu}^i |\varphi_{\nu}\rangle, \quad (2.37)$$

multiplicando por una función de base específica  $\langle\varphi_{\mu}|$  e integrando en el espacio correspondiente tendremos un conjunto de ecuaciones,

$$\sum_{\nu}^K c_{\nu}^i \langle\varphi_{\mu}|\hat{F}_i|\varphi_{\nu}\rangle = \varepsilon_i \sum_{\nu}^K c_{\nu}^i \langle\varphi_{\mu}|\varphi_{\nu}\rangle, \quad \text{para } \mu = 1, 2, \dots, K. \quad (2.38)$$

Dado un conjunto de funciones base, existen las representaciones matriciales de cada operador, llamamos a  $F_{\mu\nu}$  la matriz de Fock y a  $S_{\mu\nu}$  la matriz de traslape, cuyas formas analíticas son:

$$F_{\mu\nu} = \langle\varphi_{\mu}|\hat{F}|\varphi_{\nu}\rangle = \int d\mathbf{r} \varphi_{\mu}(\mathbf{r})^* \hat{F}(\mathbf{r}) \varphi_{\nu}(\mathbf{r}), \quad (2.39)$$

$$S_{\mu\nu} = \langle\varphi_{\mu}|\varphi_{\nu}\rangle = \int d\mathbf{r} \varphi_{\mu}(\mathbf{r})^* \varphi_{\nu}(\mathbf{r}). \quad (2.40)$$

La matriz de Fock consta de dos términos, uno que involucra los términos monoeléctricos y otro, los bielectrónicos.

$$F_{\mu\nu} = H_{\mu\nu}^{core} + G_{\mu\nu}. \quad (2.41)$$

Los monoeléctricos corresponden a los términos asociados a la energía cinética y a la interacción núcleo-electrón,  $H_{\mu\nu}^{core} = T_{\mu\nu} + V_{\mu\nu}^{ne}$ .

$$T_{\mu\nu} = \langle\varphi_{\mu}|\frac{1}{2}\nabla^2|\varphi_{\nu}\rangle, \quad (2.42)$$

$$V_{\mu\nu}^{ne} = \langle\varphi_{\mu}|\sum_A \frac{Z_A}{|\mathbf{r}-\mathbf{R}_A|}|\varphi_{\nu}\rangle. \quad (2.43)$$

Mientras los términos bielectrónicos se construyen dependiendo de la aproximación. En el método de Hartree-Fock (HF), tenemos

$$G_{\mu\nu} = \sum_{\lambda\sigma} \left( 2 \sum_i^{N/2} c_{\lambda}^i c_{\sigma}^{i*} \right) \left[ (\mu\nu|\sigma\lambda) - \frac{1}{2}(\mu\lambda|\sigma\nu) \right]. \quad (2.44)$$

y, por otro lado, en el método de Kohn y Sham (KS),

$$G_{\mu\nu} = \sum_{\lambda\sigma} \left( 2 \sum_i^{N/2} c_{\lambda}^i c_{\sigma}^{i*} \right) (\mu\nu|\sigma\lambda) + \int d\mathbf{r} \varphi_{\mu}^*(\mathbf{r}) v_{xc}(\mathbf{r}) \varphi_{\nu}(\mathbf{r}), \quad (2.45)$$

donde la última integral se evalúa de forma numérica, y el término  $v_{xc}(\mathbf{r})$  dependerá del tipo de funcional empleado, resulta importante mencionar que el término bielectrónico ( $G_{\mu\nu}$ ) presentado anteriormente es aplicable a sistemas de capa cerrada, existiendo las ecuaciones análogas para aquellos de capa abierta.

En ambos métodos las contribuciones bielectrónicas necesitan de los coeficientes ( $\{c_\nu^i\}$ ), por lo que se resuelve de forma iterativa, en un ciclo llamado ciclo autoconsistente.

Finalmente, con esta representación, dada por la base, el problema integrodiferencial se transforma en un problema de valores propios generalizado,

$$FC = SCE. \quad (2.46)$$

La elección de una base  $\varphi_\alpha$  adecuada es un requerimiento para el éxito del cálculo, una forma es utilizar *funciones tipo Slater* (STO),

$$\varphi_{\zeta,n,l,m}(\mathbf{r}) = Y_l^m(\theta, \phi) r^{n-1} e^{-\zeta r}, \quad (2.47)$$

o bien *funciones tipo gaussianas* (GTO),

$$\varphi_{\alpha,l_x,l_y,l_z}(\mathbf{r}) = x^{l_x} y^{l_y} z^{l_z} e^{-\alpha r^2}, \quad (2.48)$$

donde la suma de los términos  $l_x, l_y$  y  $l_z$  caracteriza al tipo de orbital; de tal manera que aquellos con el mismo valor en la suma,  $l_x + l_y + l_z$ , se le denomina capa.

## 2.4 Métodos Semiempíricos

Los métodos semiempíricos surgieron como una alternativa para abordar sistemas de tamaño medio, con la infraestructura computacional de las décadas de los 70 y 80 del siglo pasado. Hoy en día los métodos semiempíricos tienen un nuevo nicho de aplicación en sistemas biológicos; estos sistemas, por el tamaño, son difícilmente abordables con metodologías como Hartree-Fock o Kohn-Sham. Sin embargo, son fácilmente tratables con esta familia de métodos.

Los métodos semiempíricos toman como base el método de Hartree-Fock e incorporan más aproximaciones, siendo tal vez la más importante la que se conoce como, traslape diferen-

cial nulo (ZDO del inglés *Zero differential overlap*), el cual ignora los productos de las funciones base ubicados sobre distintos átomos. Si  $\varphi_{\mu}^A(\mathbf{r})$  es la función  $\mu$  localizada en el átomo  $A$  y  $\varphi_{\nu}^B(\mathbf{r})$  es la función  $\nu$  localizada en  $B$ , esto se ejemplifica como

$$\varphi_{\mu}^A(\mathbf{r})\varphi_{\nu}^B(\mathbf{r}) = \delta_{\mu\nu}\delta_{AB}. \quad (2.49)$$

Esto trae como consecuencia directa que la matriz de traslape en las ecuaciones de Roothaan sea la matriz identidad ( $\mathbb{S} = \mathbb{1}$ ) y que las integrales correspondientes a cuatro centros se vean reducidas a dos. Así el costo computacional del método Hartree-Fock, aproximadamente  $K^4$ , se ve disminuido en los métodos semiempíricos a  $K^2$ , con  $K$  como el número de funciones base.

Otra de las aproximaciones en este método es la descripción exclusivamente de los electrones de valencia, los cuales se representan con una base mínima de funciones tipo Slater. Así para los átomos del segundo período de la tabla periódica tenemos un total de 4 funciones base, una tipo "s" y tres "p".

Es natural pensar que con las aproximaciones hechas, los resultados no serían los adecuados para tener una metodología de amplio uso. Esto fue remediado al construir toda una familia de métodos semiempíricos cuya principal meta es la correcta descripción de propiedades químicas. Este trabajo fue originalmente desarrollado por Dewar y Thiel [8] presentando el método MNDO (*Modified Neglect of Diatomic Overlap*); este método, así como a sus sucesores AM1 (*Austin Model 1*) [9] y PM3 (*Parametric Method number 3*) [10], se parametrizaron para minimizar las sumas ponderadas de los cuadrados de los errores en propiedades moleculares.

Hoy en día el método más reciente es el denominado PM7 (*Parametric Method number 7*) [11], el cual ha sido desarrollado por Stewart y contiene parámetros que permiten describir de manera correcta interacciones no covalentes como lo son puentes de hidrógeno, las cuales son interacciones que aparecen de forma ubicua en los sistemas biológicos.

La densidad electrónica proveniente de los métodos semiempíricos puede evaluarse conociendo los vectores propios de las ecuaciones de Roothaan, siendo las funciones de base escritas de la forma siguiente,

$$\varphi(\mathbf{r}) \equiv \varphi(\mathbf{r}; \zeta, n, l, m, \mathbf{R}) = N |\mathbf{r} - \mathbf{R}|^{n-1} e^{-\zeta|\mathbf{r}-\mathbf{R}|} Y_l^m(\theta, \phi), \quad (2.50)$$

donde  $\zeta$  es el exponente de la primitiva,  $n, l$  y  $m$  son los números cuánticos primario, secundario y magnético,  $\mathbf{R}$  es la posición donde está centrada la primitiva,  $N$  es la constante de normalización de la primitiva y  $Y_l^m(\theta, \phi)$  es el armónico esférico asociado a esta.

Mientras que la densidad electrónica es determinada por medio de la siguiente expresión:

$$\rho(\mathbf{r}) = \sum_i^{Occ} \omega_i \left[ \sum_{\mu}^K c_{\mu i} \varphi_{\mu}(\mathbf{r}) \right]^2 = \sum_i^{Occ} \omega_i \sum_{\mu}^K \left[ c_{\mu i}^2 \varphi_{\mu}^2(\mathbf{r}) \right], \quad (2.51)$$

donde en esta última expresión se utilizó que el traslape diferencial es nulo entre dos funciones de base.

## 2.5 Métodos post Hartree-Fock

El método de Hartree-Fock es una buena descripción de la estructura electrónica de átomos y moléculas. Sin embargo, hay distintos casos donde la descripción a nivel de HF no es lo suficientemente precisa.

A la diferencia entre la energía no relativista del sistema y la energía, en el límite de base infinita de Hartree-Fock, se le denomina energía de correlación [13], cantidad que siempre es negativa.

$$E_{corr} = E_{exacta} - E_{HF}. \quad (2.52)$$

Existen diversos métodos que van más allá de HF, entre estos podemos citar [13, 14].

- Interacción de configuraciones (CI).
- Teoría de perturbaciones Møller - Plesset (MP).
- Clúmulos acoplados (CC).
- Método de multiconfiguraciones (MCSCF).

### 2.5.1 Método de Interacción de Configuraciones

Recordamos que la función de onda, en el método de HF corresponde a un sólo determinante de Slater formado por el producto de  $N$  orbitales moleculares ocupados,

$$\Psi_0 = |\phi_1, \phi_2, \dots, \phi_N\rangle. \quad (2.53)$$

Si tenemos  $M$  orbitales moleculares, pero sólo  $N$  ocupados, tendremos  $M - N$  orbitales virtuales, en donde podemos crear diferentes configuraciones por medio de excitaciones de uno o más electrones, desde los orbitales ocupados a los orbitales virtuales, por ejemplo:

$$\begin{aligned}\Psi_0 &= |\phi_1, \phi_2, \dots, \phi_a, \phi_b, \dots, \phi_N\rangle \\ \Psi_a^r &= |\phi_1, \phi_2, \dots, \phi_r, \phi_b, \dots, \phi_N\rangle \\ \Psi_{ab}^{rs} &= |\phi_1, \phi_2, \dots, \phi_r, \phi_s, \dots, \phi_N\rangle\end{aligned}\quad (2.54)$$

Estas configuraciones pueden mezclarse para obtener una mejor aproximación a la función de onda,

$$\Phi = c_0 \Psi_0 + \sum_{ar} c_a^r \Psi_a^r + \sum_{\substack{a < b \\ r < s}} c_{ab}^{rs} \Psi_{ab}^{rs} + \dots \quad (2.55)$$

sin embargo, el número de configuraciones crece de manera exponencial haciendo este método computacionalmente costoso y sólo aplicable a sistemas pequeños.

## 2.5.2 Método de Perturbaciones

Por otro lado, en metodologías como teoría de perturbaciones, expresamos el hamiltoniano en términos de un hamiltoniano conocido ( $\mathcal{H}_0$ ) y una pequeña variación  $\mathcal{V}$ , teniendo la siguiente ecuación,

$$\mathcal{H}|\Phi_i\rangle = (\mathcal{H}_0 + \mathcal{V})|\Phi_i\rangle = \mathcal{E}_i|\Phi_i\rangle \quad (2.56)$$

de la cual conocemos las eigenfunciones y los eigenvectores de  $\mathcal{H}_0$ ,

$$\mathcal{H}_0|\Psi_i^{(0)}\rangle = E_i^{(0)}|\Psi_i^{(0)}\rangle. \quad (2.57)$$

Si la perturbación es pequeña  $\mathcal{V}$ , como sucede en la energía de correlación, podemos suponer que la función de onda es muy parecida a la función del sistema sin perturbar, de tal forma que podemos introducir un parámetro  $\lambda$  que sintoniza la perturbación,

$$\mathcal{H}_\lambda = \mathcal{H}_0 + \lambda \mathcal{V}, \quad (2.58)$$

y realizar una expansión en series de Taylor en  $\lambda$ ,

$$\mathcal{E}_i = E_i^{(0)} + \lambda E_i^{(1)} + \lambda^2 E_i^{(2)} + \dots, \quad (2.59)$$



$$|\Phi_i\rangle = |\Psi_i^{(0)}\rangle + \lambda|\Psi_i^{(1)}\rangle + \lambda^2|\Psi_i^{(2)}\rangle + \dots \quad (2.60)$$

Aplicando la forma de la función de onda (2.60) sobre el hamiltoniano (2.58), y agrupando las potencias de  $\lambda$ , llegamos a las ecuaciones siguientes:

$$\begin{aligned} \mathcal{H}_0|\Psi_i^{(0)}\rangle &= E_i^{(0)}|\Psi_i^{(0)}\rangle & n=0 \\ \mathcal{H}_0|\Psi_i^{(1)}\rangle + \mathcal{V}|\Psi_i^{(0)}\rangle &= E_i^{(0)}|\Psi_i^{(1)}\rangle + E_i^{(1)}|\Psi_i^{(0)}\rangle & n=1 \\ \mathcal{H}_0|\Psi_i^{(2)}\rangle + \mathcal{V}|\Psi_i^{(1)}\rangle &= E_i^{(0)}|\Psi_i^{(2)}\rangle + E_i^{(1)}|\Psi_i^{(1)}\rangle + E_i^{(2)}|\Psi_i^{(0)}\rangle & n=2, \text{ etc.} \end{aligned} \quad (2.61)$$

Estas relaciones nos llevan a las siguientes expresiones para las energías:

$$\begin{aligned} E_i^{(0)} &= \langle\Psi_i^{(0)}|\mathcal{H}_0|\Psi_i^{(0)}\rangle & n=0 \\ E_i^{(1)} &= \langle\Psi_i^{(0)}|\mathcal{V}|\Psi_i^{(0)}\rangle & n=1 \\ E_i^{(2)} &= \langle\Psi_i^{(0)}|\mathcal{V}|\Psi_i^{(1)}\rangle & n=2 \\ E_i^{(3)} &= \langle\Psi_i^{(0)}|\mathcal{V}|\Psi_i^{(2)}\rangle & n=3, \text{ etc.} \end{aligned} \quad (2.62)$$

El esquema de trabajo es usar la ecuación (2.61) para obtener la función de onda y, con ello, obtener el orden deseado en la energía de la ecuación (2.62).

Por ejemplo, tomemos la expresión (2.61) para  $n=1$  y factorizamos la función de onda de cada lado. De la ecuación (2.62) con  $n=1$  sustituimos el valor de  $E_i^{(1)}$ :

$$\left(E_i^{(0)} - \mathcal{H}_0\right)|\Psi_i^{(1)}\rangle = \left(\mathcal{V} + \langle\Psi_i^{(0)}|\mathcal{V}|\Psi_i^{(0)}\rangle\right)|\Psi_i^{(0)}\rangle. \quad (2.63)$$

Recurriendo al hecho de que las funciones propias de  $\mathcal{H}_0$  generan un conjunto ortogonal que podemos usar como base, expandimos en términos de estas funciones propias a  $|\Psi_i^{(1)}\rangle$ :

$$|\Psi_i^{(1)}\rangle = \sum_m c_m^{(1)}|\Psi_m^{(0)}\rangle \quad \text{con} \quad c_m^{(1)} = \langle\Psi_m^{(0)}|\Psi_i^{(1)}\rangle. \quad (2.64)$$

Multiplicando la ecuación (2.63) por  $\langle\Psi_m^{(0)}|$  y tomando en consideración que estas funciones son ortogonales,

$$\left(E_i^{(0)} - E_m^{(0)}\right)\langle\Psi_m^{(0)}|\Psi_i^{(1)}\rangle = \langle\Psi_m^{(0)}|\mathcal{V}|\Psi_i^{(0)}\rangle, \quad (2.65)$$

o de otra forma

$$\langle\Psi_m^{(0)}|\Psi_i^{(1)}\rangle = \frac{\langle\Psi_m^{(0)}|\mathcal{V}|\Psi_i^{(0)}\rangle}{E_i^{(0)} - E_m^{(0)}}. \quad (2.66)$$

Por otro lado, tenemos una expresión para la función a primer orden,

$$|\Psi_i^{(1)}\rangle = \sum_m |\Psi_m^{(0)}\rangle \langle \Psi_m^{(0)} | \Psi_i^{(1)} \rangle \quad \text{para } m \neq i. \quad (2.67)$$

La cual podemos sustituir en la expresión (2.62) para  $n = 2$ ,

$$E_i^{(2)} = \sum_m \langle \Psi_i^{(0)} | \mathcal{V} | \Psi_m^{(0)} \rangle \langle \Psi_m^{(0)} | \Psi_i^{(1)} \rangle \quad \text{para } m \neq i. \quad (2.68)$$

En esta última expresión, sustituyendo la dada por (2.66),

$$E_i^{(2)} = \sum_m \frac{\langle \Psi_i^{(0)} | \mathcal{V} | \Psi_m^{(0)} \rangle \langle \Psi_m^{(0)} | \mathcal{V} | \Psi_i^{(0)} \rangle}{E_i^{(0)} - E_m^{(0)}} \quad \text{para } m \neq i. \quad (2.69)$$

Finalmente, puede manipularse la ecuación anterior, para obtener una forma explícita para la corrección a la energía de Møller-Plesset de segundo orden dada por la ecuación (2.70) y que recupera gran parte de la correlación electrónica (80% - 90%).

$$E^{\text{MP2}} = \sum_{i < j} \sum_{a < b} \frac{\text{occ vir} (\langle \phi_i \bar{\phi}_j | \phi_a \phi_b \rangle - \langle \phi_i \phi_j | \phi_b \phi_a \rangle)}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \quad (2.70)$$

### 2.5.3 Orbitales Naturales

Es obvio que en los métodos post-HF expresar a la densidad electrónica no parece ser tan transparente, como sucede en el método de HF o de KS. Obtener la densidad electrónica, así como sus primeras y segundas derivadas, resulta vital para análisis posteriores; y una cantidad que nos ayuda a expresarla es la matriz de densidad reducida de orden uno, la cual está definida como

$$\gamma(\mathbf{x}_1, \mathbf{x}'_1) = N \int d\mathbf{x}_2 \cdots \mathbf{x}_N \Phi(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N) \Phi^*(\mathbf{x}'_1, \mathbf{x}_2, \cdots, \mathbf{x}_N). \quad (2.71)$$

La matriz de densidad de orden uno puede expresarse en términos de los espín orbitales de HF  $\{\chi_i\}$ , como

$$\gamma(\mathbf{x}_1, \mathbf{x}'_1) = \sum_{ij} \chi_i(\mathbf{x}_1) \gamma_{ij} \chi_j^*(\mathbf{x}'_1), \quad (2.72)$$

donde  $\gamma_{ij}$ , es la representación matricial de la matriz de densidad en la base ortonormal  $\{\chi_i\}$ .

En un caso general,  $\Phi$  puede provenir de una expansión CI (2.55), CI-truncada, de cúmu-

los acoplados o de métodos perturbativos(2.60).

Aprovechamos el hecho de que  $\gamma_{ij}$  es hermítica y, por lo tanto, existe una transformación  $\mathbb{U}$  que la hace diagonal; a esos elementos los llamamos orbitales naturales ( $\{\eta_i\}$ )

$$\chi_i = \sum_k \eta_k U_{ki}^*, \quad \eta_i = \sum_k \chi_k U_{ki}, \quad (2.73)$$

a los elementos diagonales de la matriz de densidad, les llamamos números de ocupación  $\lambda_i$ , y se encuentran acotados entre cero y uno,

$$\lambda_i = \sum_{k,l} U_{ik}^* \gamma_{kl} U_{li}. \quad (2.74)$$

De forma general, la densidad electrónica expresada en términos de los orbitales naturales, toma una forma relativamente sencilla

$$\rho(\mathbf{r}) = \int d\sigma \gamma(\mathbf{r}\sigma, \mathbf{r}\sigma), \quad \text{con} \quad \mathbf{x} \equiv \mathbf{r}\sigma. \quad (2.75)$$

donde se ha hecho la integración sobre la variable de espín ( $\sigma$ ) y la matriz de densidad en la base de orbitales naturales tiene la forma,

$$\gamma(\mathbf{x}_1, \mathbf{x}'_1) = \sum_i \lambda_i \eta_i(\mathbf{x}_1) \eta_i^*(\mathbf{x}'_1). \quad (2.76)$$

## 2.6 Teoría de Átomos en Moléculas

La teoría de átomos en moléculas (AEM) que ha sido ampliamente utilizada como una herramienta para poder caracterizar las interacciones existentes en los sistemas químicos, tiene como base fundamental el análisis de la densidad electrónica y de su campo gradiente [16–19].

Cabe destacar que esta teoría trata de vincular el lenguaje químico, como es el concepto de grupos funcionales, a partir de analizar la densidad electrónica, pudiendo ser esta última obtenida de cálculos de estructura electrónica o de información experimental, así pues, es una teoría de espacio real.

La descripción topológica de la densidad electrónica se basa en localizar y caracterizar aquellos puntos donde su campo gradiente se anula. Al estar definida la densidad electrónica

en el espacio euclídeo  $\mathbb{R}^3$ , un punto crítico será aquel que cumpla con la relación (2.77),

$$\nabla\rho(\mathbf{r}_c) = \hat{i}\frac{\partial}{\partial x}\rho(\mathbf{r}_c) + \hat{j}\frac{\partial}{\partial y}\rho(\mathbf{r}_c) + \hat{k}\frac{\partial}{\partial z}\rho(\mathbf{r}_c) = \mathbf{0}. \quad (2.77)$$

Una vez localizado el punto ( $\mathbf{r}_c$ ) donde el gradiente se anula, el siguiente paso es caracterizarlo, es decir, indicar si corresponde a un máximo, un mínimo o un punto de silla de la densidad electrónica, la herramienta que nos ofrece esa información es la matriz hessiana,

$$\mathbb{H} = \begin{pmatrix} \frac{\partial^2}{\partial x^2}\rho(\mathbf{r}) & \frac{\partial^2}{\partial x\partial y}\rho(\mathbf{r}) & \frac{\partial^2}{\partial x\partial z}\rho(\mathbf{r}) \\ \frac{\partial^2}{\partial y\partial x}\rho(\mathbf{r}) & \frac{\partial^2}{\partial y^2}\rho(\mathbf{r}) & \frac{\partial^2}{\partial y\partial z}\rho(\mathbf{r}) \\ \frac{\partial^2}{\partial z\partial x}\rho(\mathbf{r}) & \frac{\partial^2}{\partial z\partial y}\rho(\mathbf{r}) & \frac{\partial^2}{\partial z^2}\rho(\mathbf{r}) \end{pmatrix}_{\mathbf{r}_c}. \quad (2.78)$$

La matriz hessiana está asociada a un problema de valores propios con la siguiente forma:

$$\mathbb{H}\vec{u} = \lambda\vec{u}, \quad (2.79)$$

donde  $\vec{u}$ , corresponde a un sistema de vectores ortogonales y cuya dirección corresponde a las direcciones de máxima curvatura. Complementando la ecuación (2.79), cada uno de los vectores propios tiene asociado un valor propio  $\lambda_i$  para  $i \in \{1, 2, 3\}$  y que caracteriza si es máximo ( $\lambda_i < 0$ ) o mínimo ( $\lambda_i > 0$ ) en esa dirección.

Los puntos críticos se caracterizan a partir de los valores propios, si los tres son distintos de cero se nombran como puntos críticos no degenerados, por el contrario, si algún valor propio es cero, entonces, el punto crítico es llamado degenerado. Los puntos críticos no degenerados, en el espacio euclídeo ( $\mathbb{R}^3$ ), son cuatro y están caracterizados por dos valores, uno al que llamamos rango ( $\omega$ ) y otro firma ( $\sigma$ ). El rango es la suma de aquellos valores propios distintos de cero, mientras que la firma es la suma de los signos de cada valor propio.

Los cuatro puntos críticos que podemos encontrar para la densidad electrónica son:

- Atractores nucleares y no nucleares (**NCP**). Los cuales son máximos en la densidad electrónica, los atractores nucleares aparecen generalmente en las posiciones de los núcleos. Los no - nucleares (**NNACP**), aparecen donde no hay núcleos, sin embargo, hay una acumulación de densidad electrónica.
- Puntos críticos de enlace (**BCP**). Son puntos críticos que corresponden a puntos de silla de primer orden, es decir, aquellos puntos que son máximos en dos direcciones y míni-

mos en una. Por estas características aparecen entre dos máximos (NCP), existiendo una trayectoria asociada a la dirección donde es mínimo que conecta con dichos máximos.

- Puntos críticos de anillo (**RCP**). Son puntos críticos que corresponden a puntos de silla de segundo orden. Aparecen generalmente en el centro de anillos formados por NCP.
- Puntos críticos de caja (**CCP**). Finalmente, estos puntos críticos son aquellos donde la densidad electrónica es mínima comparada con los alrededores. Generalmente esta rodeado por atractores nucleares y puntos críticos de anillo.

Un resumen de estas características para cada punto crítico puede verse en la tabla 2.1, en la cual se ha seguido la convención para reportar los valores propios que es:  $\lambda_1 \leq \lambda_2 \leq \lambda_3$ .

**Tabla 2.1:** Tipos de puntos críticos no degenerados para un campo escalar en  $\mathbb{R}^3$ .

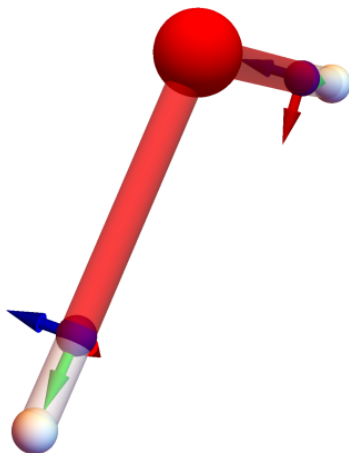
Tipo de punto crítico	$\lambda_1$	$\lambda_2$	$\lambda_3$	$(\omega, \sigma)$
Atractor (no)nuclear ( <b>NCP</b> )	-	-	-	(3, -3)
Punto crítico de enlace ( <b>BCP</b> )	-	-	+	(3, -1)
Punto crítico de anillo ( <b>RCP</b> )	-	+	+	(3, +1)
Punto crítico de caja ( <b>CCP</b> )	+	+	+	(3, +3)

Al conjunto  $\{n_{NCP}, n_{BCP}, n_{RCP}, n_{CCP}\}$ , donde  $n$  es la cantidad de puntos críticos de determinado tipo, se le conoce como conjunto característico y satisface la siguiente relación topológica, la cual es conocida como relación de Poincaré-Hopf.

$$n_{NCP} - n_{BCP} + n_{RCP} - n_{CCP} = \begin{cases} 1 & \text{moléculas aisladas,} \\ 0 & \text{sistemas periódicos.} \end{cases} \quad (2.80)$$

En la figura 2.1, se muestran los puntos críticos y vectores propios para la molécula de agua; debido a la simetría de la molécula ( $C_{2v}$ ), los dos puntos críticos tienen el mismo conjunto de valores propios siendo estos:  $\lambda_1 = -1.43884$  asociado al vector rojo,  $\lambda_2 = -1.3887$  para el vector azul y  $\lambda_3 = 1.38058$  para el vector verde. A partir de la información de la tabla 2.1 se caracteriza a los puntos críticos como puntos críticos de enlace, correspondiendo éstos a máximos en las direcciones roja y azul, y a un mínimo en la dirección verde.

Continuando con la descripción topológica de la densidad, dado un punto crítico de enlace (3, -1), existe una trayectoria que une cada **BCP** con los máximos locales **NCP**. Esta trayectoria máxima en la densidad es conocida como trayectoria de enlace (*bond path* en inglés), la



**Figura 2.1:** Puntos críticos encontrados para la molécula de agua, se muestran los vectores propios, indicando las direcciones de máxima curvatura.

cual es una curva paramétrica que depende del gradiente de la densidad electrónica:

$$\frac{d\mathbf{r}(s)}{ds} = \frac{\nabla\rho(\mathbf{r}(s))}{\|\nabla\rho(\mathbf{r}(s))\|}, \quad (2.81)$$

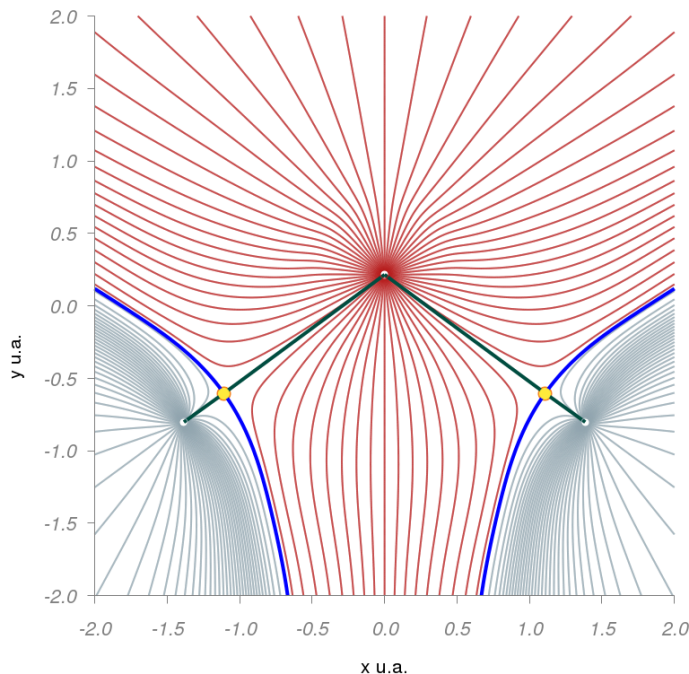
esta ecuación corresponde a una ecuación diferencial ordinaria, con valor de frontera  $\mathbf{r}_{BCP}$ , para resolverla usualmente se usan métodos numéricos como es el de Runge-Kutta.

Por otro lado, el análisis exhaustivo del gradiente de la densidad permite realizar una partición del volumen total en lo que se conoce como cuencas (*basin*), que están delimitadas por superficies interatómicas también llamadas superficies de flujo cero, estas superficies quedan definidas matemáticamente por la siguiente relación:

$$\nabla\rho(\mathbf{r}_S) \cdot \hat{\mathbf{n}}(\mathbf{r}_S) = 0 \quad \forall \mathbf{r}_S \in S(\Omega), \quad (2.82)$$

donde  $\hat{\mathbf{n}}(\mathbf{r})$  son vectores normales a las superficies interatómicas. De esta forma, un átomo topológico es la unión de un atractor, máximo en la densidad (**NCP**) y la cuenca ( $\Omega$ ).

Una imagen en dos dimensiones de los elementos topológicos de la molécula de agua puede observarse en la figura 2.2. Las líneas rojas son aquellos caminos de gradiente que comienzan en el infinito y terminan en el oxígeno, por otro lado, las líneas grises corresponden a los caminos gradiente de las cuencas asociadas a cada uno de los hidrógenos. Se puede apreciar como el espacio queda dividido por las líneas azules, siendo estas la proyección de la superficie de interacción atómica sobre el plano que contiene a la molécula de agua. Complementando la topología, las líneas verdes corresponden a los caminos de enlace que unen a los atractores vía un punto crítico de enlace, el cual se encuentra marcado en la figura por los círculos amarillos.



**Figura 2.2:** Elementos topológicos para la molécula de agua, cuencas, trayectorias de enlace, puntos críticos.

La partición del espacio total, en cuencas atómicas, lleva a la partición de las propiedades electrónicas en contribuciones. Entre las propiedades más estudiadas se encuentran: las cargas atómicas, los volúmenes atómicos y las energías. Así, el valor esperado para una propiedad mecano-cuántica sobre la molécula es equivalente a la suma de cada contribución atómica,

$$N(\Omega) = \int_{\Omega} dV \rho(\mathbf{r}), \quad (2.83)$$

y de esta forma podemos determinar la población para el átomo dentro de la cuenca ( $\Omega$ ). Naturalmente, la suma sobre todas las cuencas deberá de ser el número total de electrones en el sistema. De esta forma, pueden obtenerse cargas, llamadas cargas de Bader, a partir de la diferencia entre la población de la cuenca  $N(\Omega)$  y la carga nuclear  $Z_{\Omega}$ ,

$$q(\Omega) = Z_{\Omega} - N(\Omega). \quad (2.84)$$

Es esta idea de particionar las contribuciones electrónicas y la definición de los átomos topológicos es lo que lleva a la idea natural de grupo funcional, es decir, que las cantidades son aditivas y que las propiedades pueden ser transferibles de un sistema a otro, en la medida que la densidad electrónica permanezca inalterada.

Otra cantidad importante en AEM es el laplaciano de la densidad electrónica, el cual es la divergencia del vector gradiente,

$$\nabla \cdot \nabla \rho(\mathbf{r}) = \nabla^2 \rho(\mathbf{r}) = \frac{\partial^2}{\partial x^2} \rho(\mathbf{r}) + \frac{\partial^2}{\partial y^2} \rho(\mathbf{r}) + \frac{\partial^2}{\partial z^2} \rho(\mathbf{r}). \quad (2.85)$$

El signo del laplaciano es un indicativo acerca de las regiones en donde la densidad electrónica se concentra o disgrega con respecto al entorno,

$$\nabla^2 \rho(\mathbf{r}) \begin{cases} > 0 & \text{densidad disgregada,} \\ < 0 & \text{densidad acumulada.} \end{cases} \quad (2.86)$$

Este valor también nos permite clasificar los puntos críticos de enlace (**BCP**) para interacciones de capa compartida ( $\nabla^2 \rho(\mathbf{r}) < 0$ ), covalentes, e interacciones de capa cerrada ( $\nabla^2 \rho(\mathbf{r}) > 0$ ) entre los que podemos encontrar: interacciones iónicas, van der Waals y enlaces de hidrógeno.

Existen importantes relaciones entre las densidades de energía cinética, potencial y total basadas en la forma local del teorema del virial, las cuales son las siguientes [17]:

$$\frac{1}{4} \nabla^2 \rho(\mathbf{r}_c) = 2G(\mathbf{r}_c) + V(\mathbf{r}_c), \quad (2.87)$$

$$H(\mathbf{r}_c) = G(\mathbf{r}_c) + V(\mathbf{r}_c). \quad (2.88)$$

En donde  $G(\mathbf{r}_c)$  es la densidad de energía cinética,  $V(\mathbf{r}_c)$  es la densidad de energía potencial o también llamada campo virial, y  $H(\mathbf{r}_c)$  es la densidad de energía total. La densidad de energía cinética ( $G$ ) es una cantidad siempre positiva, mientras que ( $V$ ) siempre es negativa, por lo que otra forma de caracterizar las interacciones químicas puede ser vía estas cantidades [20, 21].

Una búsqueda de puntos críticos, con el laplaciano como función escalar, permite localizar aquellos puntos donde la concentración de la densidad es máxima, estos puntos críticos están en concordancia con el modelo de Lewis sobre los pares de electrones. El reto, en el caso del laplaciano, es la necesidad de derivadas de orden mayor para efectuar la búsqueda de puntos críticos, así como también las oscilaciones que presenta este campo y la complejidad en la interpretación de los diversos puntos críticos.

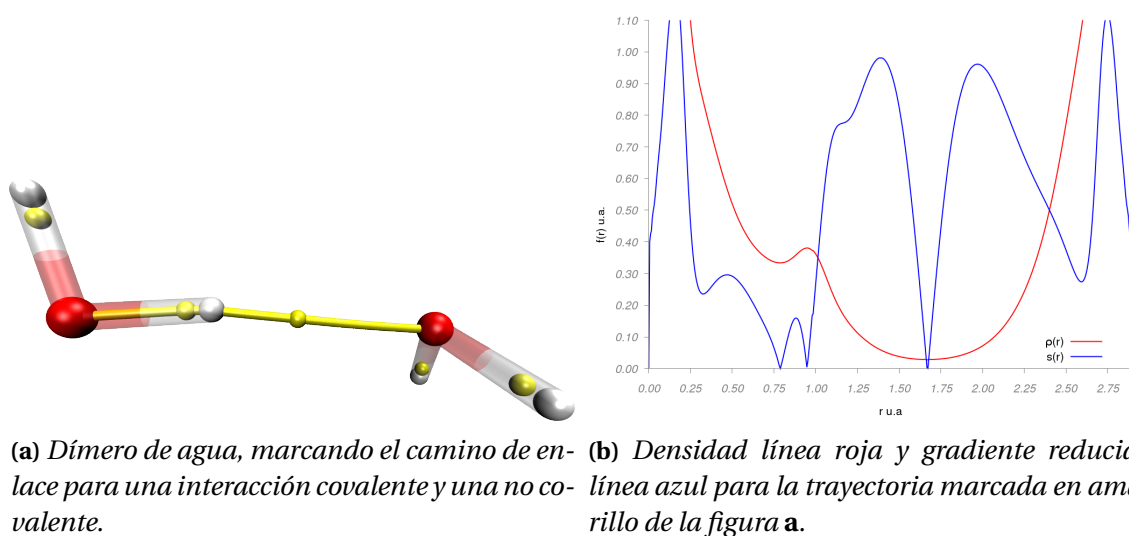


## 2.7 Interacciones No Covalentes

Un análisis que resulta complementario al ofrecido por AEM es la determinación del índice de interacciones no covalentes (INC) [22, 23]; este índice corresponde también a un análisis de espacio real. La cantidad principal en este estudio es el gradiente reducido de la densidad, también llamado gradiente adimensional, que goza de un importante rol en el diseño de funcionales como el funcional de Perdew, Burke y Ernzerhof [24] dentro del formalismo de KS. Esta cantidad se define como:

$$s(\mathbf{r}) = \frac{|\nabla\rho(\mathbf{r})|}{2(3\pi^2)^{1/3}\rho^{4/3}(\mathbf{r})}. \quad (2.89)$$

Las interacciones no covalentes, al contrario de las covalentes, se ubican en zonas de baja densidad electrónica, por ejemplo: analizando la densidad electrónica y el gradiente reducido en interacciones covalentes y no covalentes del dímero de agua, podemos ver en qué régimen de densidad se encuentran.

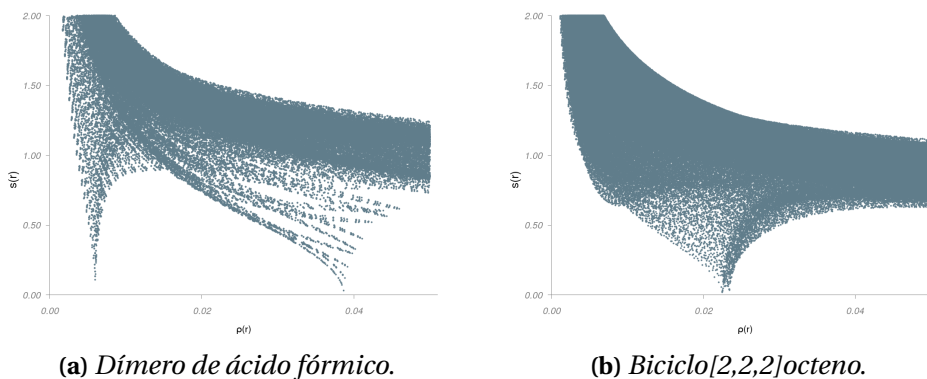


(a) Dímero de agua, marcando el camino de enlace para una interacción covalente y una no covalente. (b) Densidad línea roja y gradiente reducido línea azul para la trayectoria marcada en amarillo de la figura a.

**Figura 2.3:** Densidad electrónica y gradiente reducido para interacciones covalentes y no covalentes.

Se puede observar en la figura 2.3 que las interacciones no covalentes aparecen en zonas de baja densidad electrónica ( $\rho(\mathbf{r}) \approx 0.1$ ) y las covalentes son en general un orden de magnitud mayor.

Analizando los mapas de  $\rho(\mathbf{r})$  contra  $s(\mathbf{r})$  para dos sistemas, uno con interacciones atractivas (puente de hidrógeno), dímero de ácido fórmico figura 2.4a, y otro para interacciones repulsivas (interacciones estéricas), biciclo[2,2,2]octeno figura 2.4b, se observa que ambas inte-



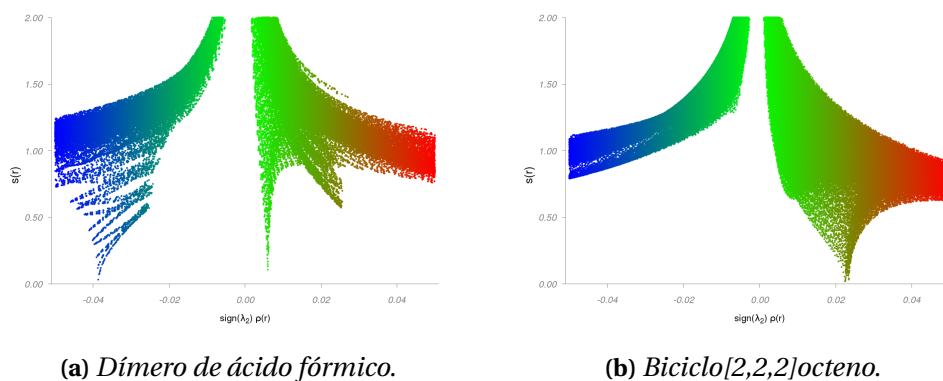
**Figura 2.4:** Mapas de  $\rho(\mathbf{r})$  vs  $s(\mathbf{r})$  para dos sistemas modelo.

**Tabla 2.2:** Clasificación de las interacciones no covalentes con base en el signo del segundo valor propio.

Tipo de interacción	$\Lambda(\mathbf{r}) = \text{sign}(\lambda_2)\rho(\mathbf{r})$
Fuerte y atractiva	$\Lambda(\mathbf{r}) < 0$
Débil	$\Lambda(\mathbf{r}) \approx 0$
Fuerte y repulsiva	$\Lambda(\mathbf{r}) > 0$

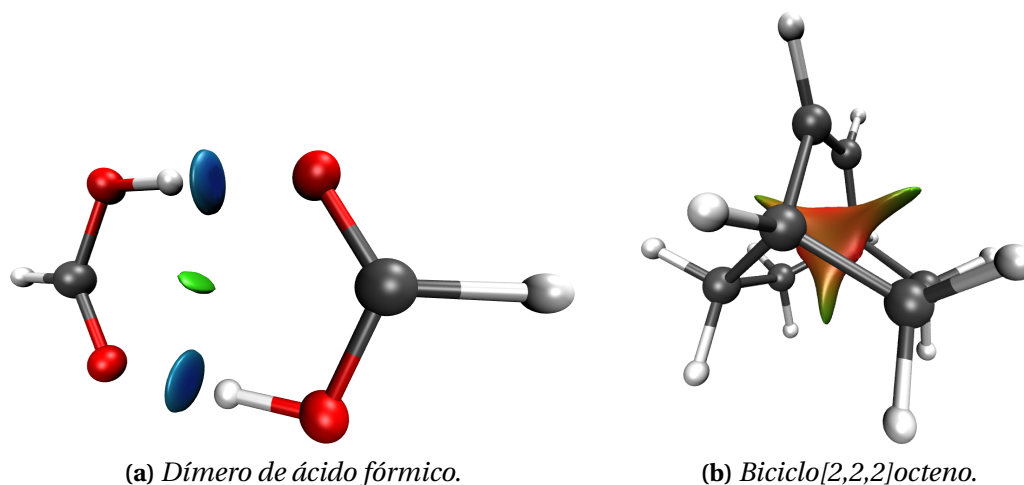
racciones aparecen en las mismas zonas, por lo tanto, para poder discernir si se trata de una interacción repulsiva o atractiva, se ha usado el criterio del segundo valor propio ( $\lambda_2$ ) de la matriz hessiana [17].

Añadiendo el criterio del segundo valor del hessiano, para los sistemas descritos anteriormente, podemos observar una separación en las interacciones; para hacer más notoria esta separación se ha coloreado la figura 2.5, siguiendo la convención en [22, 23], azul para las interacciones fuertes y atractivas, verde para las débiles y rojo para las repulsivas.



**Figura 2.5:** Desdoblamiento en las interacciones no covalentes debido, al criterio de  $\lambda_2$ .

Usando este sistema de colores, anteriormente descrito, sobre isosuperficies de gradiente reducido, obtenemos los mapas de interacciones no covalentes, que pueden verse en la figura 2.6, para los dos sistemas modelo.



**Figura 2.6:** Interacciones no covalentes para dos sistemas modelo, en azul las interacciones atractivas, en verde las interacciones de van der Waals y en rojo las interacciones repulsivas.



---

---

## CAPÍTULO 3

---

# MÉTODO DE ELEMENTO FINITO, MALLAS Y PROGRAMACIÓN EN GPUS

### 3.1 Método de Elemento Finito

El método de elemento finito (MEF) es una metodología general para la resolución de ecuaciones diferenciales. Se usa ampliamente en el diseño de estructuras mecánicas, en dinámica de fluidos y, en general, en sistemas con diferentes geometrías [25, 26].

Algunas características importantes del MEF son las siguientes:

- Es una formulación variacional.
- El problema que se resuelve se formula en su forma débil.
- El dominio de interés se subdivide en subdominios simples.
- Las funciones solución son polinomiales sobre los subdominios.

Estas propiedades permiten resolver ecuaciones diferenciales de segundo orden.

Este método presenta la ventaja de que es posible resolver ecuaciones diferenciales sobre dominios complicados, donde otros métodos como diferencias finitas resultaría deficiente. Otra ventaja que resulta importante es la relativa facilidad con la que se manipulan las condiciones de frontera. Así, este método permite resolver la ecuación de Schrödinger para distintos tipos de potenciales.

### 3.1.1 Formulación del Método de Elemento Finito 1D

Reformulemos la ecuación de Schrödinger unidimensional en términos del método de elemento finito. Para ello, partimos de la ecuación independiente del tiempo,

$$-\frac{1}{2} \frac{d^2}{dx^2} u + Vu - Eu = 0, \quad \text{para } 0 \leq x \leq L, \quad (3.1)$$

con un potencial unidimensional  $V = V(x)$ , definido en el intervalo  $[0, L]$ , siendo  $u = u(x)$  la solución y con las condiciones de frontera tipo Dirichlet siguientes:

$$u(0) = u(L) = 0. \quad (3.2)$$

Para realizar la formulación débil escogemos una función de prueba  $v$ , también conocida como función de desplazamiento virtual. Multiplicamos la ecuación diferencial original por esta función e integramos en el dominio de interés,

$$-\frac{1}{2} \int_0^L dx \left( \frac{d^2}{dx^2} u \right) v + \int_0^L dx Vuv - E \int_0^L dx uv = 0, \quad (3.3)$$

utilizando una integración por partes para el primer término podemos expresar la ecuación (3.3) como

$$-\frac{1}{2} \left( \frac{d}{dx} u \right) v \Big|_0^L + \frac{1}{2} \int_0^L dx \left( \frac{d}{dx} u \right) \left( \frac{d}{dx} v \right) + \int_0^L dx Vuv - E \int_0^L dx uv = 0. \quad (3.4)$$

Dado que la función de prueba  $v$  debe de pertenecer al mismo grupo que la función  $u$ , ambas deben de tener las mismas condiciones de frontera ( $v(0) = v(L) = 0$ ), lo que hace que el primer término se anule quedando sólo expresada como,

$$\frac{1}{2} \int_0^L dx \left( \frac{d}{dx} u \right) \left( \frac{d}{dx} v \right) + \int_0^L dx Vuv - E \int_0^L dx uv = 0. \quad (3.5)$$

Es importante notar que, en la formulación original del problema,  $u$  debe de tener al menos una segunda derivada (clase  $C^2$ ), pero en la formulación débil buscamos la solución en un espacio mayor, debido a que imponemos que tenga al menos una derivada (clase  $C^1$ ).

El espacio donde podemos encontrar la solución  $u$  del problema débil está dentro de un espacio vectorial, que además es un espacio de dimensión infinita. Como computacionalmente no es posible manipular espacios infinitos, entonces se acota la dimensión del espacio donde existe la solución del problema y hablamos de un espacio de dimensión  $N$ . En este espacio finito, la solución puede ser escrita como una combinación lineal de las funciones en dicho espacio,

$$u(x) \approx u_N(x) = \sum_i^N c_i \varphi_i(x), \quad (3.6)$$

mientras,

$$v(x) \approx v_N(x) = \sum_j^N b_j \varphi_j(x), \quad (3.7)$$

a este proceso de generar una solución aproximada ( $u_N$ ) en un espacio finito se le denomina método de Galerkin [25].

Finalmente, la formulación débil, y en un espacio de dimensión finita, toma la siguiente forma.

$$\sum_{i=1}^N \sum_{j=1}^N \left( \int_0^L dx \left( \frac{d}{dx} \varphi_i \right) \left( \frac{d}{dx} \varphi_j \right) + \int_0^L dx V \varphi_i \varphi_j \right) c_i b_j = \sum_{i=1}^N \sum_{j=1}^N E \left( \int_0^L dx \varphi_i \varphi_j \right) c_i b_j. \quad (3.8)$$

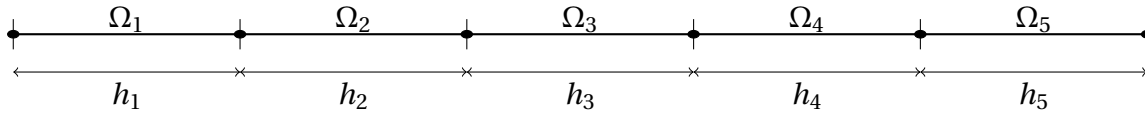
Este problema genera un sistema de ecuaciones que podemos escribir de forma matricial, similar a las ecuaciones de Fock,

$$(\mathbb{T} + \mathbb{V})\mathbb{C} = E\mathbb{S}\mathbb{C}. \quad (3.9)$$

El método de elemento finito nos proporciona una forma particular para construir las funciones  $\{\varphi_i\}_{i=1}^N$ , dentro de la aproximación de Galerkin. Para ello definimos a las funciones  $\varphi_i$  como funciones polinomiales por tramos sobre cada uno de los subdominios.

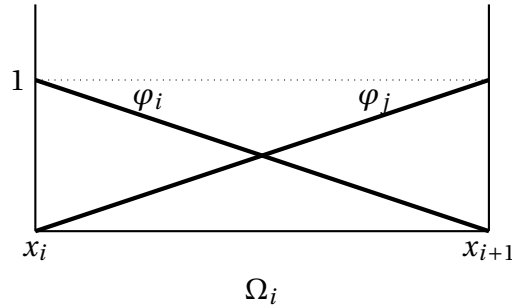
Al conjunto de nodos y elementos se le denomina malla y es de vital importancia tener una buena calidad en la malla dependiendo de la aplicación a resolver, ver figura 3.1.

Por otro lado, dentro de cada elemento generamos las funciones de forma, la aproximación más sencilla es la función de interpolación lineal. Para el caso del elemento  $\Omega_i$  tenemos



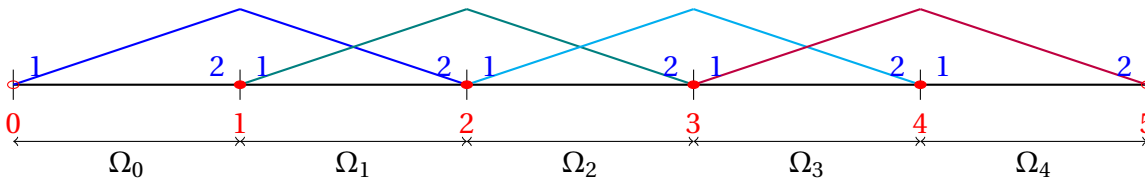
**Figura 3.1:** Malla unidimensional de 6 nodos (puntos) con 5 elementos  $\Omega_i$  y cada uno con tamaño  $h_i$ .

dos funciones de forma, la izquierda que denominamos  $\varphi_i$  y la derecha que denominamos  $\varphi_j$ , como se puede ver en la figura 3.2.



**Figura 3.2:** Funciones de forma lineales dentro del elemento  $\Omega_i$ .

La figura completa se puede ver en la figura 3.3 donde tenemos la malla (nodos y elementos) y las funciones de forma para cada elemento.



**Figura 3.3:** Funciones de forma lineales en un dominio discretizado por 5 elementos. Se puede ver una numeración global (roja) y una numeración local (azul).

Es importante mencionar que las funciones de forma  $\{\varphi_i\}$  tienen características importantes, una de las principales:

$$\varphi_i(x_j) = \delta_{ij}. \tag{3.10}$$

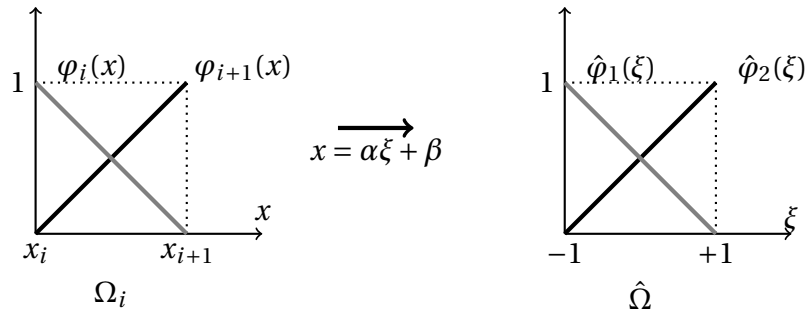
Las funciones que cumplen con esta propiedad las generamos a partir de los polinomios interpoladores de Lagrange, los cuales los formamos a partir de la siguiente expresión:

$$\varphi_j(x) = \prod_{\substack{i=0 \\ i \neq j}}^k \frac{x - x_i}{x_j - x_i}. \tag{3.11}$$

En la figura 3.3 es posible observar como hay seis nodos y cinco elementos, para poder rea-



lizar las operaciones de forma sistemática, se toma un elemento al que nombramos elemento maestro ( $\hat{\Omega}$ ) y con respecto a él referimos las integrales que aparecen. Este procedimiento permite generalizar las operaciones necesarias para cada elemento  $e$  de la malla, para ello basta aplicar una transformación lineal del espacio real  $x$  al espacio del nodo maestro  $\xi$ , como se ilustra en la figura 3.4.



**Figura 3.4:** Funciones de forma del elemento  $\Omega_i$  y referenciadas al elemento maestro  $\hat{\Omega}$ .

De esta forma, podemos mapear las integrales sobre un elemento  $\Omega_i$ , al elemento maestro  $\hat{\Omega}$ .

$$\int_{x_i}^{x_{i+1}} dx \varphi_1(x) \varphi_2(x) = \alpha \int_{-1}^{+1} d\xi \varphi_1(\alpha\xi + \beta) \varphi_2(\alpha\xi + \beta) \quad (3.12)$$

$$= \alpha \int_{-1}^{+1} d\xi \hat{\varphi}_1(\xi) \hat{\varphi}_2(\xi) \quad (3.13)$$

con

$$\alpha = \frac{x_{i+1} - x_i}{2} \quad \text{y} \quad \beta = \frac{x_{i+1} + x_i}{2}. \quad (3.14)$$

Para funciones de forma lineales tendremos matrices elementales de tamaño  $2 \times 2$ , las cuales estarán determinadas de la siguiente forma:

$$\hat{\mathbb{S}} = \begin{pmatrix} \int_{-1}^{+1} d\xi \hat{\varphi}_1(\xi) \hat{\varphi}_1(\xi) & \int_{-1}^{+1} d\xi \hat{\varphi}_1(\xi) \hat{\varphi}_2(\xi) \\ \int_{-1}^{+1} d\xi \hat{\varphi}_2(\xi) \hat{\varphi}_1(\xi) & \int_{-1}^{+1} d\xi \hat{\varphi}_2(\xi) \hat{\varphi}_2(\xi) \end{pmatrix} \quad (3.15)$$

Lo mismo se aplica para la matriz elemental de energía cinética. Por otro lado, la matriz elemental para el término del potencial, se maneja no con una forma analítica del potencial  $V(x)$ , sino que se utiliza la misma familia de polinomios interpoladores que las funciones de forma. De esta manera, la matriz del potencial está en términos de valores discretos del potencial, valores que se ubican en los nodos de la malla. Para el elemento maestro con funciones de

forma lineal, la expresión de la matriz está dada por:

$$\hat{\mathbb{V}} = \begin{pmatrix} \frac{\nu_0}{2} + \frac{\nu_1}{6} & \frac{\nu_0}{6} + \frac{\nu_1}{6} \\ \frac{\nu_0}{6} + \frac{\nu_1}{6} & \frac{\nu_0}{6} + \frac{\nu_1}{2} \end{pmatrix}. \quad (3.16)$$

Dependiendo del orden del polinomio ( $p$ ) tendremos matrices de tamaño  $(p+1) \times (p+1)$ .

Con estos elementos expuestos y usando la propiedad de sumabilidad, las matrices del problema débil, pueden ser escritas en términos de matrices elementales, es decir, las matrices  $\mathbb{T}$ ,  $\mathbb{V}$  y  $\mathbb{S}$  se reestructuran como la suma de matrices para cada elemento  $\Omega^e$ .

$$\mathbb{S} = \sum_e \mathbb{S}^e \quad (3.17)$$

$$\mathbb{S} = \begin{pmatrix} S_{11}^0 & S_{12}^0 & 0 & 0 & 0 & 0 \\ S_{21}^0 & S_{22}^0 + S_{11}^1 & S_{12}^1 & 0 & 0 & 0 \\ 0 & S_{21}^1 & S_{22}^1 + S_{11}^2 & S_{12}^2 & 0 & 0 \\ 0 & 0 & S_{21}^2 & S_{22}^2 + S_{11}^3 & S_{12}^3 & 0 \\ 0 & 0 & 0 & S_{21}^3 & S_{22}^3 + S_{11}^4 & S_{12}^4 \\ 0 & 0 & 0 & 0 & S_{21}^4 & S_{22}^4 \end{pmatrix} \quad (3.18)$$

A este proceso se le conoce como armado de las matrices para el caso de elementos de orden uno, resulta sencillo, pero conforme se incrementa el orden, la matriz comienza a tener más complejidad. Así, es importante contar con la matriz de ligadura, que vincula los nodos locales, con los nodos globales de la matriz.

Cuando se resuelven las ecuaciones en el método de elemento finito, lo que obtenemos son los valores de la función de onda en cada uno de los nodos:

$$\mathbb{FC} = \mathbb{E}\mathbb{S}\mathbb{C}. \quad (3.19)$$

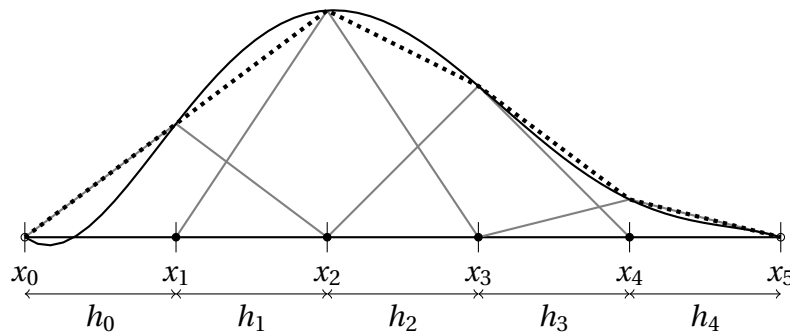
Para resolverla se busca una matriz de transformación unitaria que haga  $\mathbb{S} = \mathbb{I}$ , para ello podemos optar por una ortogonalización simétrica,

$$\mathbb{X} = \mathbb{U}\mathbb{Z}^{-1/2}\mathbb{U}^\dagger. \quad (3.20)$$

o una ortogonalización canónica,

$$\mathbb{X} = \cup \mathbb{Z}^{-1/2}. \quad (3.21)$$

Al resolver por medio de alguna de estas transformaciones se encuentra el conjunto de valores (C) que constituyen la solución de nuestra ecuación diferencial, es decir el valor que toma la solución en el nodo  $x_j$ . Como puede verse en la figura 3.5, la solución exacta  $u(x)$  esta dada por la línea suave y continua, mientras, la solución aproximada  $u_N(x)$  está descrita por la línea a trozos y punteada, y corresponde a la suma de las funciones de base (funciones sobrero).



**Figura 3.5:** Solución exacta continua y suave, solución aproximada punteada, con 5 elementos.

### 3.1.2 Formulación del Método de Elemento Finito 2D

Para sistemas multielectrónicos resulta ventajoso resolver primero las contribuciones mono-electrónicas. Para ello, podemos dividir el hamiltoniano en contribuciones de un sólo electrón y de dos núcleos,

$$\hat{h}u(\mathbf{r}) = -\frac{1}{2}\nabla^2 u(\mathbf{r}) + V^{NE}(\mathbf{r}) u(\mathbf{r}) = \varepsilon u(\mathbf{r}), \quad (3.22)$$

con

$$V^{NE}(\mathbf{r}) = -\frac{Z_A}{\|\mathbf{R}_A - \mathbf{r}\|} - \frac{Z_B}{\|\mathbf{R}_B - \mathbf{r}\|}. \quad (3.23)$$

para una molécula diatómica.

La ecuación mono-electrónica en su forma débil puede ser escrita como:

$$\int_{\Omega} \nabla v \cdot \nabla u \, d\mathbf{r} - \int_{\Gamma} \frac{\partial u}{\partial n} v \, dS + 2 \int_{\Omega} V^{NE} v u \, d\mathbf{r} = 2\varepsilon \int_{\Omega} u v \, d\mathbf{r}. \quad (3.24)$$

Imponiendo simetría azimutal la ecuación diferencial es separable para la parte azimutal y por el MEF se resolverá la parte no separable. Por lo anterior, las integrales que aparecen en la

ecuación (3.24) corresponden a integrales sobre dominios bidimensionales, es decir,  $\Omega \in \mathbb{R}^2$ .

En este caso, la condición que imponemos a la función de onda es anularse en las fronteras, por lo tanto, la contribución del término de frontera en la formulación débil será nula,

$$\int_{\Gamma} \frac{\partial u}{\partial n} v \, dS = 0. \quad (3.25)$$

Por otro lado, las integrales se resuelven sobre cada subdominio (elemento). Las condiciones sobre los elementos serán las siguientes:

- La unión de los elementos forman el dominio:

$$\bigcup_{e=1}^{N_e} \Omega^e = \Omega.$$

- Dos elementos no se intersectan:

$$\Omega^e \cap \Omega^{e'} = \emptyset.$$

Con las propiedades anteriores se tiene:

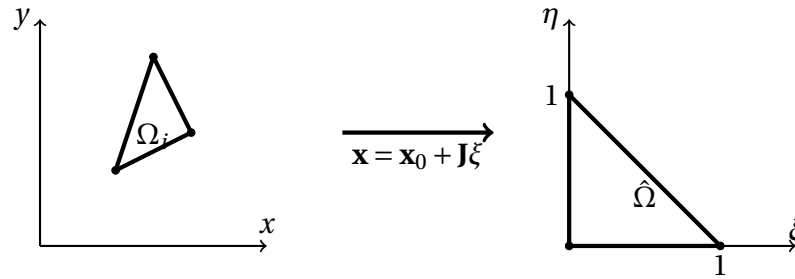
$$\int_{\Omega} \nabla u \cdot \nabla v \, d\mathbf{r} = \sum_e \int_{\Omega^e} \nabla \varphi_i^e \cdot \nabla \varphi_j^e \, d\mathbf{r}, \quad (3.26)$$

donde cada  $\Omega^e$  es el subdominio de un elemento en la malla.

En el caso unidimensional, las funciones de forma más sencilla corresponden a polinomios lineales, también llamadas funciones sombrero. En el caso bidimensional, las funciones de forma más sencilla corresponden a planos que se intersectan en lo que llamamos funciones piramidales.

De igual forma, resulta conveniente referenciar cualquier elemento  $\Omega_i$  en el plano  $XY$  a un elemento maestro  $\hat{\Omega}$  en el plano  $\xi\eta$ , como se muestra en la figura 3.6.

La malla en el método de elemento finito corresponde a los nodos y a la conectividad existente entre ellos. Es importante clasificar los nodos en nodos externos y nodos internos, debido a que es en los primeros donde asumiremos que la función de onda es cero, a estos nodos también suele llamarse nodos tipo Dirichlet. Por su parte, para el ensamblado de las matrices será necesario conocer la correcta conectividad entre los puntos.



**Figura 3.6:** Elemento triangular  $\Omega_i$  en el plano  $XY$  y elemento maestro  $\hat{\Omega}$  en el plano  $\xi\eta$ .

El método que resulta ideal para realizar el mallado, (la conectividad de puntos) es la triangulación de Delaunay [27, 28].

## 3.2 Mallas Estructuradas y No Estructuradas

Podemos definir como mallas estructuradas a aquellas que con pocos elementos, es posible generar todos los nodos y las conectividades. En una dimensión, el ejemplo claro es una malla uniformemente espaciada, dado que sólo necesitamos el punto inicial  $x_0$ , el espaciamiento  $h_x$  y el número total de nodos  $i_{max}$ ,

$$x = x_0 + i \times h_x. \quad (3.27)$$

Esto mismo se puede aplicar en dos y tres dimensiones,

$$\begin{aligned} x &= x_0 + i \times h_x, \\ y &= y_0 + j \times h_y, \\ z &= z_0 + k \times h_z. \end{aligned} \quad (3.28)$$

Un ejemplo de este tipo de mallas, en química, es la utilizada en los archivos cube.

Archivo .cube

```

1 Test density
2 lambda_2 density for NCI
3 10 -9.664389 -6.206247 -4.000126
4 194 0.100000 0.000000 0.000000
5 125 0.000000 0.100000 0.000000
6 81 0.000000 0.000000 0.100000
7 1 1.000000 5.664384 -0.485365 -0.000126
8 6 6.000000 3.607304 -0.332536 0.000051
9 8 8.000000 2.876871 2.052382 -0.000057
10 1 1.000000 0.999245 2.130679 0.000069
11 8 8.000000 -2.233315 2.206249 0.000035
12 6 6.000000 -3.607307 0.332533 -0.000020
13 8 8.000000 -2.876865 -2.052380 -0.000054
14 8 8.000000 2.233310 -2.206247 0.000037
15 1 1.000000 -0.999237 -2.130667 0.000031
16 1 1.000000 -5.664389 0.485346 0.000150

```

```

17  1.000000E+02  1.000000E+02  1.000000E+02  1.000000E+02  1.000000E+02  1.000000E+02
18  1.000000E+02  1.000000E+02  1.000000E+02  1.000000E+02  1.000000E+02  1.000000E+02
19  ...

```

Los parámetros necesarios para generar la malla son los tres últimos valores de la línea 3, que indican el origen de la malla,  $(x_0, y_0, z_0)$ . Los siguientes valores a destacar son los encontrados en la línea 4, 5 y 6, donde el primer dato corresponde al número de puntos en esa dirección  $(i_{max}, j_{max}, k_{max})$  y el siguiente es el espaciamiento  $(h_x, h_y, h_z)$ .

Los demás datos en grupos de 5 (línea 7 a 16) al número atómico, carga y coordenadas  $(x, y, z)$  para cada uno de los núcleos en el archivo. El siguiente grupo de datos pertenecen al campo escalar en cada uno de los nodos (líneas 17 hasta el fin del archivo), es posible visualizar los datos usando distintos tipos de software en química (VMD [29], Jmol [30], Pymol [31], Molden [32], etc.).

Por otro lado, existen mallas no estructuradas, donde es necesario escribir explícitamente cada coordenada de los nodos, así como también la conectividad en ellos; un formato bastante extendido es el formato dado por la librería VTK [33].

#### Archivo.vtk

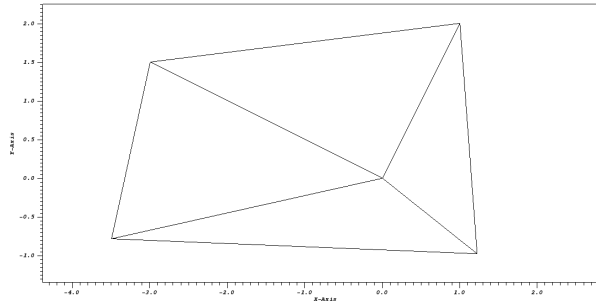
```

1 Prueba para MEF
2 ASCII
3 DATASET UNSTRUCTURED_GRID
4 POINTS 5 double
5 0.000000 0.000000 0.000000
6 1.000000 2.000000 0.000000
7 -3.000000 1.500000 0.000000
8 -3.500000 -0.781831 0.000000
9 1.222521 -0.974928 0.000000
10 CELLS 4 16
11 3 0 1 2
12 3 0 2 3
13 3 0 3 4
14 3 0 4 1
15 CELL_TYPES 4
16 5
17 5
18 5
19 5
20 POINT_DATA 5
21 SCALARS material double
22 LOOKUP_TABLE default
23 0.000000
24 1.000000
25 1.500000
26 1.250000
27 1.750000

```

Para el caso de mallas no estructuradas es importante indicar las coordenadas de cada punto en la malla, así como la información del tipo de elemento y la conectividad. Esos datos podemos verlos en el archivo VTK listado y cuya representación se puede ver en la figura 3.7. Las mallas de este tipo no son tan ampliamente usadas en química, pero si en otras ramas de

la ciencia, como meteorología, en dinámica de fluidos, diseño de estructuras metálicas y en la elaboración de imágenes por resonancia magnética nuclear. En este proyecto, los archivos de la triangulación han sido elaborados en un formato VTK y graficados por medio del paquete computacional VisIt [34].



**Figura 3.7:** Ejemplo de malla no estructurada en un archivo VTK y visualizada en VisIt.

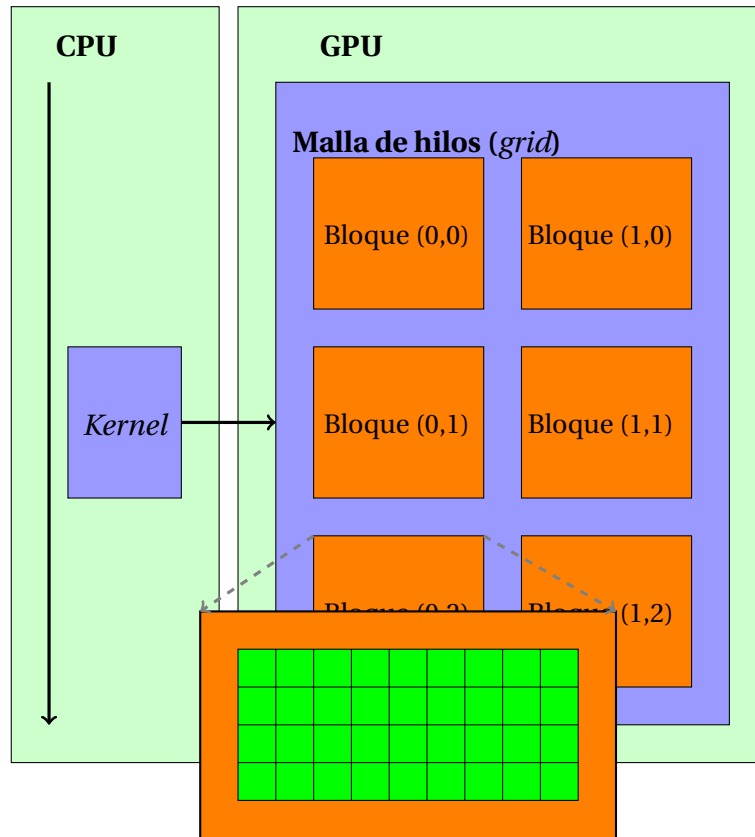
### 3.3 Programación en GPUs

Las GPUs son unidades de procesamiento gráfico que hasta hace poco sólo eran usadas para el procesamiento de videojuegos y la composición de imágenes 3D. Actualmente, es posible hacer uso de ellas para cómputo general. Como tal, una de las áreas donde más se ha explotado es en el cómputo científico, evidencia de ello es que la mayoría de las supercomputadoras [6] corresponde a sistemas que integran GPUs con CPUs. La aplicación de las GPUs fuera del ámbito gráfico [35] van desde dinámica computacional de fluidos, predicción del clima, aprendizaje automático (*machine learning*), bioinformática y en química computacional [36–44]. En esta última rama se han hecho grandes esfuerzos en dinámica de proteínas, acoplamiento molecular (*docking*), en acelerar cálculos semiempíricos, acelerar cálculos de estructura electrónica y en la evaluación rápida y eficiente de las integrales de repulsión electrónica.

Al programar sobre unidades de procesamiento gráfico es necesaria una modificación en el paradigma de programación, dado que se efectúan operaciones en dos unidades distintas. La unidad de procesamiento central (CPU por sus siglas en inglés) y la de procesamiento gráfico (GPU de *graphics processing units*). A este modelo de programación se le denomina heterogéneo y, para ello es necesario contar con dos tipos de compiladores, uno para el CPU y otro para el GPU.

El diagrama general de ejecución se puede ver en la figura 3.8, el control principal lo

mantiene el procesador central (CPU) y, en las partes a paralelizar, se hacen las llamadas necesarias a la tarjeta gráfica (GPU) esto por medio de funciones llamadas *kernel*. Cada ejecución del *kernel* genera una malla compuesta por bloques de hilos, siendo cada hilo la entidad mínima de procesamiento que ejecuta el *kernel*.



**Figura 3.8:** Diagrama de una ejecución en un sistema CPU-GPU.

En este proyecto usamos como entorno de desarrollo el ofrecido por NVIDIA, ya que cuenta con diferentes herramientas y con el compilador propio para GPUs. Por lo que a partir de aquí, lo referente a la programación en GPUs será exclusivo en el entorno CUDA [36, 37, 45], es importante mencionar que existen marcos de trabajo diferentes al ofrecido por NVIDIA. En 2011, diversas empresas en el ramo tecnológico se reunieron para generar un nuevo estándar en la computación heterogénea (CPU-GPU o CPU-Coprocesadores), es el llamado OPEN CL [46–48].

Para ejemplificar los elementos básicos de la programación usando CUDA, se presenta un pequeño código que evaluó una función  $f(\mathbf{r})$  sobre una malla en tres dimensiones, el programa completo puede revisarse en el apéndice A. La función a evaluar será la siguiente,

$$f(\mathbf{r}) = f(x, y, z) = (3z^2 - r^2)e^{-r} \quad \text{con} \quad r = \sqrt{x^2 + y^2 + z^2}, \quad (3.29)$$



que salvo una constante de normalización, corresponde a la forma analítica de un orbital  $d_{z^2}$ , localizado en el origen.

Como se mencionó líneas arriba, el *kernel* es aquella función que se ejecuta de forma paralela en cada hilo de ejecución. Esta función tiene una particularidad, y es que dentro del estándar CUDA, la función *kernel* es del tipo **void**, además de tener la palabra reservada **\_\_global\_\_** precedida en su definición, esta palabra indica que se trata de una función que es llamada por el Cpu pero ejecutada en la tarjeta gráfica, es decir esta función conecta directamente al CPU con la GPU.

**Código 3.1:** Definición del *kernel*

```

22 __global__ void kernel (int , int , int , int ,
23                        double , double , double ,
24                        double , double , double ,
25                        double* ) ;

```

Una parte importante en programación sobre unidades de procesamiento gráfico, es la manipulación de índices sobre los cuales se trabaja y opera. Así, resulta conveniente asignar a cada elemento de trabajo (hilo) un identificador; los hilos pueden estar acomodados en una o dos dimensiones formando bloques, los cuales de igual forma pueden encontrarse también en una o dos dimensiones. De acuerdo al problema a tratar se realizará el armado de la malla.

CUDA permite identificar cada hilo y bloque que forman la malla. Para el manejo de cada uno de ellos se hace uso de variables propias del lenguaje, estas se resumen en la tabla 3.1.

**Tabla 3.1:** Variables para poder controlar y manipular los hilos de una malla.

Identificador	Descripción
threadIdx	Identificador para los hilos dentro de un bloque.
blockIdx	Identificador para los bloques.
blockDim	Cantidad de hilos por bloque.
gridDim	Cantidad de bloques por malla.

A partir de la figura 3.9, es posible notar que los índices o localizadores para cada hilo son únicos dentro de cada bloque, pero no dentro de la malla, es decir, existen distintos hilos cuyo identificador será (2,4). La forma de poder acceder a los hilos de forma unívoca consiste en manipular el índice del hilo y del bloque para generar un índice único. A este conjunto de índices podemos llamarle índices-gpu, dado que cada uno tendrá una relación con cada hilo de



**Figura 3.9:** Malla para la ejecución de un kernel, se trata de una malla  $4 \times 2$  bloques y cada bloque de  $4 \times 6$  hilos.

procesamiento. Existe otra clase de índices, los cuales son los propios de nuestra aplicación, en el caso de la evaluación de un campo escalar en un espacio tridimensional, estos índices son los que están dados por las ecuaciones (3.28) y los podemos denominar índices-problema.

### Código 3.2: Relación entre índices

```

290 // Kernel que es llamado desde el host
291 // Cambia de la malla real (fisica ) a la malla virtual
292 // para generar cada x,y,z
293 __global__ void kernel (int nx,int ny, int nz, int npt,
294                         double x0, double y0, double z0,
295                         double hx, double hy, double hz,
296                         double *valor) {
297
298 // índices-problema
299 unsigned int idx ,idy ,idz ;
300 // índices-gpu
301 unsigned int indiceHilo = threadIdx.y*blockDim.x + threadIdx.x ;
302 unsigned int indiceBloq = blockIdx.y*gridDim.x + blockIdx.x ;
303
304 unsigned int indice = (blockDim.x*blockDim.y)*indiceBloq + indiceHilo ;
305 double x,y,z ;
306
307 if (indice < npt) {

```

```

308 // Apartir de los indices-gpu genero los indices-problema
309 // Una malla 2D genera la malla 3D
310 idx = (unsigned int) (indice/nz)/ny;
311 idy = (unsigned int) (indice/nz)%ny;
312 idz = (unsigned int) (indice%nz);
313
314 x = x0 + idx*hx;
315 y = y0 + idy*hy;
316 z = z0 + idz*hz;
317
318 valor[indice] = funcionDev(x,y,z);
319
320 } else
321 valor[indice] = 0.;

```

En el código 3.2 se puede ver, en la línea 301 y 302 cómo se usan las variables `threadIdx`, `blockIdx`, `blockDim` y `gridDim` para poder generar el índice único para cada hilo. Más adelante en las líneas 310-312, se observa como este genera tres índices más, con los cuales es posible determinar cada valor de  $x$ ,  $y$  y  $z$  y poder evaluar la `funcionDev(x,y,z)`.

El armado de la malla es una operación que se realiza en el CPU instantes antes de llamar al *kernel*. Para ello es necesario saber cuál será el índice máximo que necesitamos y, en función de los hilos por bloque determinar los bloques a usar. Resulta indispensable garantizar que se lanzan los hilos necesarios para efectuar las operaciones. Para generar la malla nuevamente el marco de trabajo, CUDA, ofrece un tipo de variable específico para esta tarea, el cual es **dim3**. En el siguiente extracto del código podemos ver cómo para un número de puntos a calcular (nodos en la malla del problema), se determinan la cantidad de bloques, código 3.3 y como se asigna usando las variables **dim3** en el código 3.4.

### Código 3.3: Cantidad de hilos y bloques lanzados en el kernel

```

110 // Variables para generar la malla
111 bloques2D = posibles / (HILOS2D);
112 if( posibles%HILOS2D != 0)
113     bloques2D++;
114
115 bloquesX = (int) sqrt(bloques2D);
116 bloquesY = (int) bloques2D/bloquesX;
117
118 if( bloquesX*bloquesY < bloques2D)
119     bloquesX++;
120 if( bloquesX*bloquesY < bloques2D)
121     bloquesY++;

```

**Código 3.4:** Uso de variables `dim3` al generar la malla

```

141 // Genero mi malla
142 dim3 blocks (bloquesX, bloquesY, 1) ;
143 dim3 threads (HILOS1D, HILOS1D, 1) ;

```

Habiendo generado la malla, es importante tener comunicación entre la CPU y la GPU, esto se lleva a cabo con funciones que permitan direccionar los datos de la memoria RAM del CPU a la memoria interna de la tarjeta gráfica, para este proceso existen funciones parecidas al estándar C como son la función `malloc` y la función `free`, la primera permite reservar memoria RAM en el CPU y la segunda la libera, este proceso necesariamente se lleva a cabo con la utilización de punteros. Por otro lado, en CUDA las funciones para el manejo y liberación de memoria están resumidas en la tabla 3.2.

**Tabla 3.2:** Funciones implementadas en CUDA para el manejo de memoria.

Función	Uso
<code>cudaMalloc</code>	Permite reservar cierta cantidad de memoria en la tarjeta gráfica.
<code>cudaMemcpy</code>	Copia información de arreglos de CPU a GPU, y viceversa.
<code>cudaFree</code>	Libera memoria previamente reservada por <code>cudaMalloc</code> .

Por último, es posible hacer la llamada al *kernel* y lanzar los hilos necesarios para el problema en particular, para ello basta con realizar una llamada a la función, tal y como se realiza en C, pero añadiendo nuevos parámetros propios para la generación de la malla, estos se añaden dentro de paréntesis angulares, `<<< , >>>`.

**Código 3.5:** Llamada a la función *kernel*

```

138 // Reservo memoria para dev_valor en la GPU
139 cudaMalloc ((void**)&dev_valorGpu , tamaño) ;
140
141 // Genero mi malla
142 dim3 blocks (bloquesX, bloquesY, 1) ;
143 dim3 threads (HILOS1D, HILOS1D, 1) ;
144
145 // llamo al kernel
146 kernel <<<blocks , threads>>> (nx, ny, nz, npt, x0, y0, z0, hx, hy, hz, dev_valorGpu) ;
147
148
149 // Copio del device al host
150 cudaMemcpy(valorGpu, dev_valorGpu, tamaño, cudaMemcpyDeviceToHost) ;
151
152 gettimeofday(&tfin, NULL) ;
153

```

```

154 // Libero memoria en el device
155   cudaFree (dev_valorGpu);

```

En la línea 146, se observa la forma de llamar al *kernel*, el cual es ejecutado en una malla compuesta por bloques e hilos, las dimensiones de éstos están guardadas en `blocks` y `threads`.

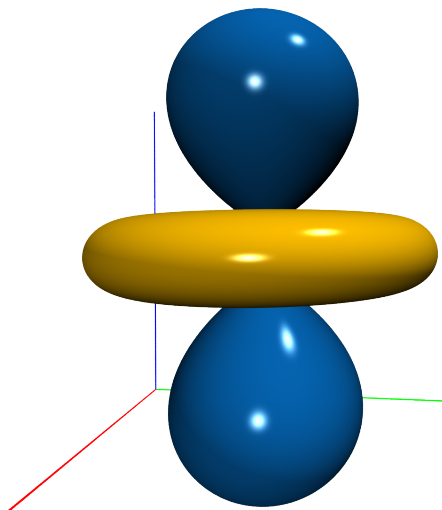
Al ejecutar el programa, este nos da la información de la malla: cuántos bloques la conforman y de cuántos hilos. También reporta la cantidad de hilos lanzados y los hilos excedentes. Los datos calculados en dicha aplicación se exportaron con formato `cube` y visualizado en `VMD`, siendo este resultado presentado en la figura 3.10, este es el proceso general para la evaluación de campos escalares.

### Ejecución

```

[ray@fedora-gpu ~] ./ejemplo.x
Numero de puntos en x,y,z
 80 80 80
Coordenadas del punto inicial r0
x0 y0 z0 : -5.0 -5.0 -5.0
Coordenadas del punto final rN
xN yN zN : 5.0 5.0 5.0
[INFO CPU ] Para calcular      512000 puntos
[INFO CPU ] Usamos      33.941984 ms
[INFO CUDA] Blocks en X:      45
[INFO CUDA] Blocks en Y:      45
[INFO CUDA] Hilos en X:       16
[INFO CUDA] Hilos en Y:       16
[INFO CUDA] Hilos totales:    518400
[INFO CUDA] Hilos ociosos:    6400
[INFO GPU ] Para calcular      512000 puntos
[INFO GPU ] Usamos      87.007046 ms
[INFO ERRO] El mayor error cuadratico entre los dos conjuntos es
[INFO ERRO] El RMSD entre los dos conjuntos es de : 7.49986E-33
[INFO ERRO] MaxError :      7.8886090522E-31
[INFO ERRO] Los dos conjuntos son estadisticamente iguales

```



**Figura 3.10:** Isosuperficie de la función escalar (3.29). Los datos fueron calculados en la GPU.



---

---

## CAPÍTULO 4

---

### PROGRAMAS DESARROLLADOS

Usando la información descrita en capítulos anteriores, se ha desarrollado una serie de aplicaciones para su posterior uso en sistemas tratados por la química teórica. Estos programas podrán encontrarse para futuras revisiones en la siguiente página web:

<https://github.com/rayhe88>.

A continuación se hará una breve descripción de estos programas.

**GPUAM:** Es un código escrito en CUDA-C y módulos en OPEN-MP. Consta de 31690 líneas, de las cuales un 45% es código en CUDA. Permite la evaluación de cantidades escalares y vectoriales de campos relacionados a la química cuántica, también permite realizar la búsqueda de puntos críticos de la densidad electrónica y del laplaciano. GPUAM determina analíticamente las funciones, así como sus primeras derivadas. El programa está codificado para usar más de una GPU, si es que en el ordenador se encuentran más, o en su defecto permite evaluar y realizar las búsquedas sólo usando CPUS y OPEN-MP.

**CUBE3D:** Es una herramienta, que en versiones posteriores será añadida a GPUAM. Consta de 4716 líneas escritas en C y algunos módulos usando OPEN-MP. Esta herramienta permite determinar los gradientes, laplaciano y hessiano de un campo 3D cualquiera. Para ello usa esquemas de interpolación para generar a trozos la función que genera el campo 3D.

Con la evaluación de estas cantidades, el programa realiza la búsqueda de puntos críticos y trayectorias de enlace en sistemas donde no se cuente con una forma analítica de la densidad electrónica.

FINIT1-2D: Es un conjunto de programas que consiste de dos módulos escritos en C. El primero permite realizar el mallado sobre una dimensión y resolver numéricamente la ecuación de Schrödinger monoeléctrica dado un potencial cualquiera. Este programa consta de 3658 líneas. Por otra lado, el segundo programa realiza el mallado sobre elipses, siguiendo un esquema de Delaunay y resolviendo la ecuación de Schrödinger monoeléctrica para sistemas diatómicos, consta de 2843 líneas de código.

## 4.1 GPUAM

GPUAM (*Graphics Processing Units for Atoms and Molecules*) es un código que permite evaluar y analizar diversos campos escalares y vectoriales de uso amplio en la química teórica. GPUAM ha sido desarrollado para hacer uso de tarjetas gráficas Nvidia, aprovechando la arquitectura en paralelo de dichos dispositivos. El código tiene como datos de entrada funciones de onda provenientes de distintas aproximaciones (semiempírica, HF, DFT, post-HF).

GPUAM puede evaluar los campos usando como funciones de base tanto orbitales tipo Gaussianos (GTO) como orbitales tipo Slater (STO). Los primeros son usados ampliamente en cálculos de estructura electrónica desde que fueron propuestos por Boys [49] y su uso ha sido tan extendido, ya que es posible encontrarlas en diversos códigos. La forma de acceder a la información de la base, así, como los coeficientes de la expansión lineal, es a través de archivos con el formato *wfn* o *wfx*. Estos archivos se pueden obtener a partir de distintos paquetes computacionales como Gaussian [50], Gamess-US [51] o NWChem [52].

En contraste a los GTOs existen los STOs, que aunque describen de mejor forma la zonas cercanas a los núcleos, su uso en códigos no ha sido tan explotado debido a la dificultad de evaluar integrales multicéntricas. Por otro lado, este tipo de funciones se ha usado en códigos que hacen uso de hamiltonianos semiempíricos, donde la evaluación de integrales se ve reducida considerablemente [53]. La forma que tenemos de acceder a esta información es vía archivos tipo *mgf* o archivos *aux*, provenientes del código Mopac 2016 [54].

Como se mencionó anteriormente, GPUAM permite usar dos tipos de funciones: Gaussianas y tipo Slater. En la tabla 4.1 se encuentran de forma condensada, los campos y análisis que



**Tabla 4.1:** Campos y análisis programados en GPUAM, para cada tipo de función de base.

Campos	STO	GTO
Orbitales moleculares	✓	✓
Densidad electrónica	✓	✓
Laplaciano de la densidad	✓	✓
Módulo del gradiente de la densidad	✓	✓
Gradiente reducido de la densidad	✓	✓
Detector de electrones localizados	✓	✓
Gradiente de orbitales moleculares	–	✓
Gradiente de la densidad	✓	✓
Densidad de energía cinética	–	✓
Función de localización electrónica	–	✓
Detector de orbitales localizados	–	✓
Potencial electrostático	–	✓
Análisis sobre campos		
Búsqueda de puntos críticos de la densidad	✓	✓
Búsqueda de puntos críticos del laplaciano	–	†
Índice de interacciones no covalentes	✓	✓
Integración sobre las cuencas de AEM	–	†

† tareas en desarrollo, – tareas que no han sido implementadas.

permite realizar GPUAM. Existen algunas tareas que están en desarrollo, por lo que los tiempos de ejecución o resultados no han sido probados minuciosamente, igualmente, hay tareas que no han sido implementadas en el código.

La densidad electrónica es escrita como una combinación del cuadrado de  $N$  orbitales moleculares ( $\phi_i$ ), multiplicada por la ocupación ( $\omega_i$ ) de cada orbital,

$$\rho(\mathbf{r}) = \sum_i^N \omega_i |\phi_i(\mathbf{r})|^2. \quad (4.1)$$

Por otro lado, los orbitales moleculares se escriben como una combinación lineal de  $K$  funciones conocidas  $g_\mu(\mathbf{r}; \alpha, \mathbf{A}, \mathbf{a})$ , mientras que los coeficientes de la combinación ( $c_\mu^i$ ) son obtenidos a partir de distintos métodos de estructura electrónica:

$$\phi_i(\mathbf{r}) = \sum_\mu^K c_\mu^i g_\mu(\mathbf{r}; \alpha, \mathbf{A}, \mathbf{a}), \quad (4.2)$$

las funciones de base  $g_\mu(\mathbf{r}; \alpha, \mathbf{A}, \mathbf{a})$  son escritas como funciones tipo gaussiana de la siguiente

forma:

$$g_{\mu}(\mathbf{r}; \alpha, \mathbf{A}, \mathbf{a}) = (x - A_x)^{a_x} (y - A_y)^{a_y} (z - A_z)^{a_z} \exp(-\alpha |\mathbf{r} - \mathbf{A}|^2), \quad (4.3)$$

donde  $\mathbf{A}$  indica dónde está centrada la gaussiana  $\mu$ ,  $\mathbf{a}$  es el tipo de primitiva y  $\alpha$  es el exponente de esta, es importante mencionar que el coeficiente  $c_{\mu}^i$  en la expresión (4.2) es el producto de los coeficientes variacionales, las constantes de normalización y las constantes de contracción. Por lo que las funciones  $g_{\mu}$  son funciones base primitivas.

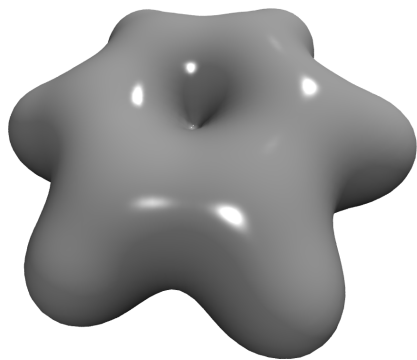
En la forma analítica de las funciones STO, aparece un término extra ( $|\mathbf{r} - \mathbf{A}|^{n-1}$ ).

$$f_{\mu}(\mathbf{r}; n, \zeta, \mathbf{A}, \mathbf{a}) = (x - A_x)^{a_x} (y - A_y)^{a_y} (z - A_z)^{a_z} |\mathbf{r} - \mathbf{A}|^{n-1} \exp(-\zeta |\mathbf{r} - \mathbf{A}|), \quad (4.4)$$

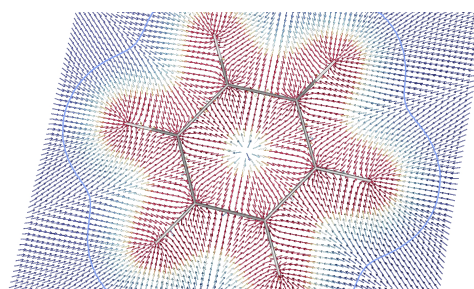
donde  $\mathbf{A}$  y  $\mathbf{a}$  tienen el significado descrito anteriormente. El exponente está dado por  $\zeta$  y  $n$ , el término extra indica el número cuántico principal.

### 4.1.1 Tipo de Tareas

Todos los campos escalares se escriben en archivos con formato cube, que es un estándar para la visualización de estos campos en química cuántica, para la visualización de ellos existen múltiples códigos tanto libres como comerciales (Vmd [29], Jmol [30], Pymol [31], Chimera [55]). Por el contrario, para los campos vectoriales, la salida se reporta en archivos con formato VTK [33], en este formato se generan dos archivos, uno con el campo vectorial evaluado en cada punto y otro con la geometría, para la visualización de éstos archivos se recomienda el uso del visualizador VISIT [34] o PARAVIEW [56], figura 4.1.



(a) Isosuperficie de la densidad electrónica.



(b) Gradiente de la densidad electrónica sobre el plano que contiene a los átomos.

**Figura 4.1:** Ejemplo de visualización de campos escalares y vectoriales para la molécula de benceno.

### 4.1.2 Campos Escalares y Vectoriales

Lo que podría considerarse el núcleo del programa es la evaluación de los orbitales y de sus primeras derivadas, dado que con esta información es posible evaluar otros campos relacionados con la densidad. De esta forma, sean  $p$  y  $q$  dos coordenadas cualesquiera ( $p, q \in \{x, y, z\}$ ), las formas analíticas de las primeras derivadas serán:

$$\rho(\mathbf{r}) = \sum_i^N \omega_i \phi_i(\mathbf{r}) \phi_i(\mathbf{r}), \quad (4.5)$$

$$\frac{\partial \rho(\mathbf{r})}{\partial q} = 2 \sum_i^N \omega_i \left[ \phi_i(\mathbf{r}) \left( \frac{\partial \phi_i(\mathbf{r})}{\partial q} \right) \right], \quad (4.6)$$

$$\frac{\partial^2 \rho(\mathbf{r})}{\partial q^2} = 2 \sum_i^N \omega_i \left[ \left( \frac{\partial \phi_i(\mathbf{r})}{\partial q} \right)^2 + \phi_i(\mathbf{r}) \left( \frac{\partial^2 \phi_i(\mathbf{r})}{\partial q^2} \right) \right], \quad (4.7)$$

$$\frac{\partial^2 \rho(\mathbf{r})}{\partial q \partial p} = 2 \sum_i^N \omega_i \left[ \left( \frac{\partial \phi_i(\mathbf{r})}{\partial q} \right) \left( \frac{\partial \phi_i(\mathbf{r})}{\partial p} \right) + \phi_i(\mathbf{r}) \left( \frac{\partial^2 \phi_i(\mathbf{r})}{\partial q \partial p} \right) \right]. \quad (4.8)$$

#### Orbitales moleculares

Evalúa la función orbital para el  $i$ -ésimo orbital escogido por el usuario.

$$\phi_i(\mathbf{r}) = \sum_{\mu}^K c_{\mu}^i g_{\mu}(\mathbf{r}; \alpha, \mathbf{A}, \mathbf{a}), \quad (4.9)$$

es importante mencionar que por la forma en que está escrito el archivo wfn o wfx, sólo es posible realizar el cálculo sobre los orbitales ocupados.

#### Densidad electrónica

Esta cantidad es de vital importancia en la TFD y AEM, su análisis permite particionar el espacio y generar los grafos moleculares, donde se indica la conectividad de los átomos dentro de la molécula.

$$\rho(\mathbf{r}) = \sum_i^N \omega_i \left( \sum_{\mu}^K c_{\mu}^i g_{\mu}(\mathbf{r}; \alpha, \mathbf{A}, \mathbf{a}) \right)^2. \quad (4.10)$$

#### Laplaciano de la densidad

Este campo está relacionado con las zonas de concentración y depleción de la densidad electrónica, sus puntos críticos están relacionados con los pares de electrones.

$$\nabla^2 \rho(\mathbf{r}) = \frac{\partial^2 \rho(\mathbf{r})}{\partial x^2} + \frac{\partial^2 \rho(\mathbf{r})}{\partial y^2} + \frac{\partial^2 \rho(\mathbf{r})}{\partial z^2}. \quad (4.11)$$

### Módulo del gradiente de la densidad

Esta cantidad corresponde al modulo del gradiente, se utiliza para la generación de otros campos. En la búsqueda de puntos críticos, esta cantidad es función objetivo.

$$|\nabla\rho(\mathbf{r})| = \sqrt{\left(\frac{\partial\rho(\mathbf{r})}{\partial x}\right)^2 + \left(\frac{\partial\rho(\mathbf{r})}{\partial y}\right)^2 + \left(\frac{\partial\rho(\mathbf{r})}{\partial z}\right)^2}. \quad (4.12)$$

### Gradiente reducido de la densidad

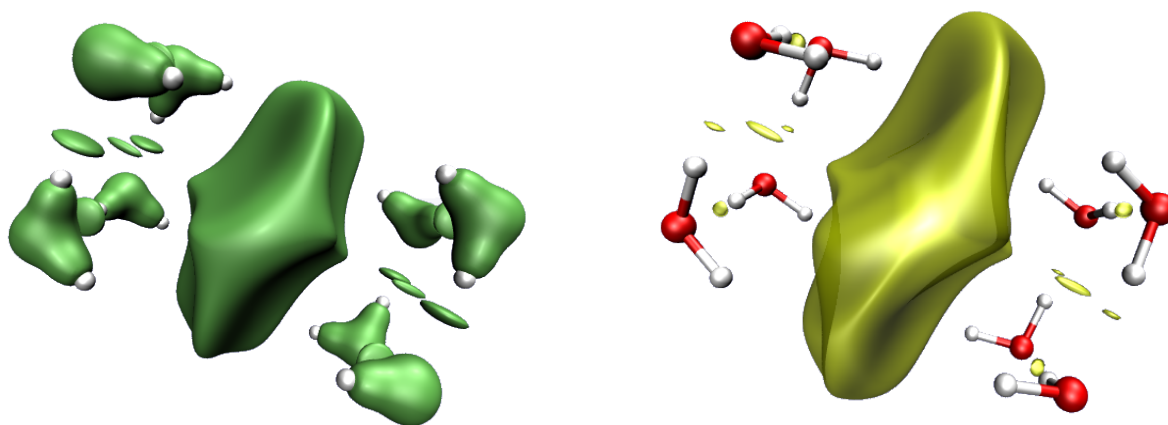
Esta cantidad es de vital importancia al determinar el índice de interacciones no covalentes (INC), una representación del gradiente reducido puede verse en la figura 4.2a,

$$s(\mathbf{r}) = \frac{1}{2(3\pi^2)^{1/3}} \frac{|\nabla\rho(\mathbf{r})|}{\rho^{4/3}(\mathbf{r})}. \quad (4.13)$$

### Detector de electrones localizados

Es una función similar al gradiente reducido, definida por Bohorquez y Boyd [57], esta función se puede apreciar en 4.2b,

$$\tilde{P}(\mathbf{r}) = \frac{|\nabla\rho(\mathbf{r})|}{2\rho(\mathbf{r})}. \quad (4.14)$$



(a) Gradiente reducido.

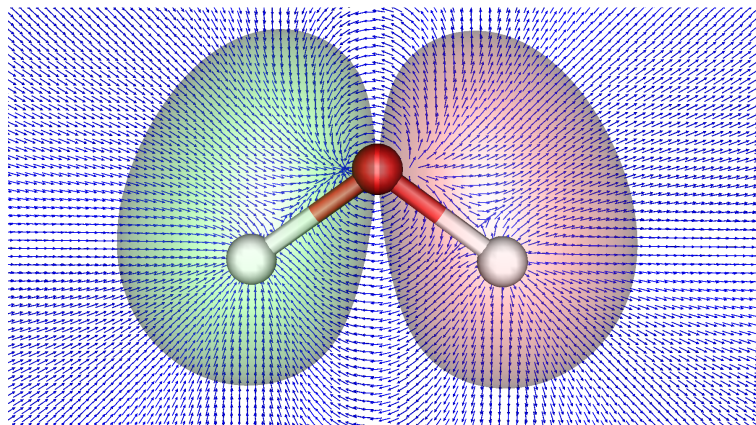
(b) Detector de electrones localizados (LED).

**Figura 4.2:** Ejemplo de dos campos relacionados con el módulo del gradiente para el sistema  $8 \cdot (\text{H}_2\text{O})^-$ .

### Gradiente de orbitales moleculares

El módulo de esta cantidad aparece en la definición de otros campos más complejos, GPUAM permite evaluarlos de forma independiente para su posterior visualización, figura 4.3,

$$\nabla\phi_i(\mathbf{r}) = \hat{i}\frac{\partial\phi_i(\mathbf{r})}{\partial x} + \hat{j}\frac{\partial\phi_i(\mathbf{r})}{\partial y} + \hat{k}\frac{\partial\phi_i(\mathbf{r})}{\partial z}. \quad (4.15)$$



**Figura 4.3:** Gradiente del orbital HOMO-2, para la molécula de agua, sobre el plano que contiene a los 3 átomos, se grafica también el orbital correspondiente. Visualizado con VisIt.

### Gradiente de la densidad

Corresponde al campo vectorial asociado a la densidad electrónica, e indica la dirección en la cual la densidad varía más rápido, su módulo representa el ritmo de variación de la densidad.

$$\nabla\rho(\mathbf{r}) = \hat{i}\frac{\partial\rho(\mathbf{r})}{\partial x} + \hat{j}\frac{\partial\rho(\mathbf{r})}{\partial y} + \hat{k}\frac{\partial\rho(\mathbf{r})}{\partial z}. \quad (4.16)$$

### Densidad de energía cinética (G)

Dado que la densidad de energía cinética no está definida de forma única, existen dos formas alternativas para determinarla, se sabe que de forma local son distintas pero integradas sobre todo el espacio son idénticas. La forma Lagrangiana o también llamada forma definida positiva es la siguiente:

$$G(\mathbf{r}) = \frac{1}{2} \sum_i^N \omega_i |\nabla\phi_i(\mathbf{r})|^2. \quad (4.17)$$

### Densidad de energía cinética (K)

Por otro lado, también es posible definir la densidad de energía cinética en una forma alternativa conocida como forma hamiltoniana,

$$K(\mathbf{r}) = -\frac{1}{2} \sum_i^N \omega_i \phi_i(\mathbf{r}) \nabla^2 \phi_i(\mathbf{r}). \quad (4.18)$$

Ambas formas locales de la densidad de energía cinética están relacionadas vía el laplaciano de la densidad,

$$\frac{1}{4} \nabla^2 \rho(\mathbf{r}) = G(\mathbf{r}) - K(\mathbf{r}). \quad (4.19)$$

**Función de localización electrónica (ELF)**

Esta función fue originalmente propuesta por Becke y Edgecombe [58]. Esta cantidad mide la localización espacial de un electrón de referencia y proporciona un método para el mapeo de la probabilidad del par de electrones.

$$\text{ELF}(\mathbf{r}) = \frac{1}{1 + \left(\frac{D(\mathbf{r})}{D_0(\mathbf{r})}\right)^2}, \quad (4.20)$$

con los siguientes términos:

$$D_0(\mathbf{r}) = \frac{3}{10} (3\pi^2)^{2/3} \rho^{5/3}(\mathbf{r}) \quad (4.21)$$

y

$$D(\mathbf{r}) = \frac{1}{2} \sum_i^N \omega_i |\nabla \phi_i(\mathbf{r})|^2 - \frac{1}{8} \frac{|\nabla \rho(\mathbf{r})|^2}{\rho(\mathbf{r})}. \quad (4.22)$$

**Detector de orbitales localizados (LOL)**

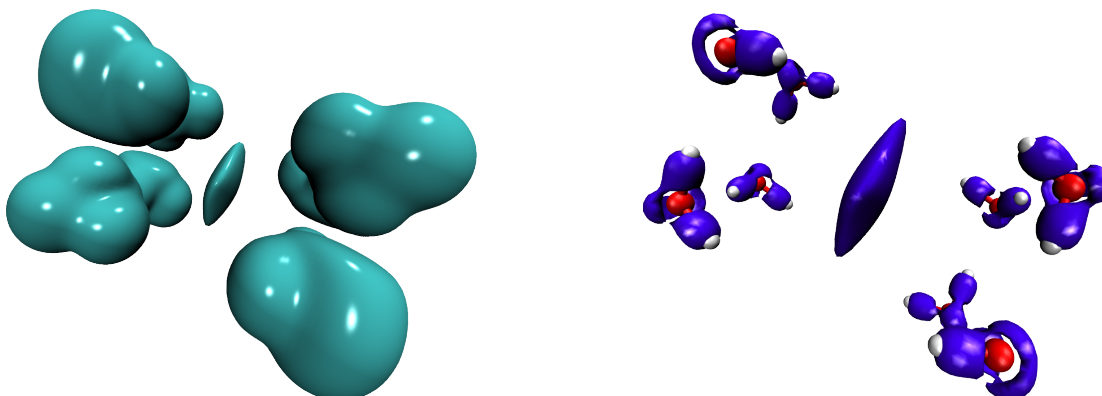
Esta función es similar a la función de localización electrónica (ELF), fue definida originalmente por Schmider y Becke [59].

$$\text{LOL}(\mathbf{r}) = \frac{\tau(\mathbf{r})}{1 + \tau(\mathbf{r})}, \quad (4.23)$$

con  $\tau(\mathbf{r})$  definido como:

$$\tau(\mathbf{r}) = \frac{D_0(\mathbf{r})}{\frac{1}{2} \sum_i^N \omega_i |\nabla \phi_i(\mathbf{r})|^2}, \quad (4.24)$$

y  $D_0(\mathbf{r})$  definida anteriormente (4.21). Ambas funciones de localización pueden verse en la figura 4.4, para el sistema  $8 \cdot (\text{H}_2\text{O})^-$ .



(a) Función de localización electrónica (ELF).

(b) Localizador de orbitales localizados (LOL).

**Figura 4.4:** Dos funciones de localización electrónica para el sistema  $8 \cdot (\text{H}_2\text{O})^-$ .

**Potencial electrostático**

Es un cantidad ampliamente usada en química orgánica e inorgánica. En bioquímica se ha usado para poder determinar sitios de unión entre proteínas y enzimas. También brinda un panorama general de la formación de enlaces de hidrógeno [60, 61].

El potencial electrostático en el punto  $\mathbf{r}$ , se determina como la contribución de un conjunto discreto de cargas puntuales (núcleos) y una distribución continua (electrones), como puede verse en la ecuación (4.25).

$$V(\mathbf{r}) = \sum_A^M \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} - \int d\mathbf{r}' \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}, \quad (4.25)$$

donde  $\mathbf{R}_A$  es el vector de posición del núcleo  $A$  con carga  $Z_A$  y  $\rho(\mathbf{r}')$  es la densidad electrónica. Haciendo uso de la expresión de la densidad dada por la ecuación (4.10), es posible escribir el segundo término del potencial electrostático en términos de integrales sobre los orbitales.

$$\int d\mathbf{r}' \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} = \sum_i^N \omega_i \int d\mathbf{r}' \frac{\phi_i(\mathbf{r}')\phi_i(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}. \quad (4.26)$$

Debido a que los orbitales son combinaciones lineales de funciones de base gaussianas, es que expresamos a esta integral como integrales sobre las primitivas  $g_\mu$  y  $g_\nu$ .

$$\sum_i^N \omega_i \int d\mathbf{r}' \frac{\phi_i(\mathbf{r}')\phi_i(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} = \sum_i^N \sum_\mu^K \sum_\nu^K \omega_i c_\mu^i c_\nu^i \int d\mathbf{r}' \frac{g_\mu(\mathbf{r}'; \alpha, \mathbf{A}, \mathbf{a})g_\nu(\mathbf{r}'; \beta, \mathbf{B}, \mathbf{b})}{|\mathbf{r} - \mathbf{r}'|} \quad (4.27)$$

$$= \sum_i^N \sum_\mu^K \sum_\nu^K \omega_i c_\mu^i c_\nu^i I_{\mu\nu}(\mathbf{r}) \quad (4.28)$$

Usando la definición de las funciones gaussianas, la integral de la ecuación anterior se escribe como:

$$I_{\mu\nu}(\mathbf{r}) = \int d\mathbf{r}' \frac{g_\mu(\mathbf{r}'; \alpha, \mathbf{A}, \mathbf{a})g_\nu(\mathbf{r}'; \beta, \mathbf{B}, \mathbf{b})}{|\mathbf{r} - \mathbf{r}'|} \quad (4.29)$$

$$= \int d\mathbf{r}' \frac{e^{-\alpha|\mathbf{r}' - \mathbf{A}|^2} e^{-\beta|\mathbf{r}' - \mathbf{B}|^2}}{|\mathbf{r} - \mathbf{r}'|} \prod_{j=1}^3 (x'_j - A_j)^{a_j} (x'_j - B_j)^{b_j}. \quad (4.30)$$

El denominador de la integral se transforma via una transformada de Laplace,

$$\frac{1}{|\mathbf{r} - \mathbf{r}'|} = \frac{2}{\sqrt{\pi}} \int_0^\infty du e^{-u^2|\mathbf{r} - \mathbf{r}'|^2}, \quad (4.31)$$

aprovechando la identidad gaussiana y el cambio de variable  $t^2 = u^2/(q + u^2)$ , se expresa la integral de la densidad electrónica en términos de una integral sobre un dominio finito y que

depende de un polinomio en  $t^2$ . Este polinomio es dependiente de los momentos angulares de las gaussianas, en el caso particular de gaussianas con simetría esférica este polinomio se reduce a la unidad, por lo que la integral se puede realizar via la función de error.

$$\int d\mathbf{r}' \frac{\rho(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} = \sum_{\mu}^K \sum_{\nu}^K \tilde{G}_{\mu\nu} \int_0^1 dt \mathcal{P}(t^2) e^{-qt^2|\mathbf{Q}-\mathbf{r}|^2}, \quad (4.32)$$

con  $q$ ,  $\mathbf{Q}$  y  $G_{\mu\nu}$  definidos de la siguiente forma,

$$q = \alpha + \beta, \quad (4.33)$$

$$\mathbf{Q} = \frac{\alpha\mathbf{A} + \beta\mathbf{B}}{q}, \quad (4.34)$$

$$G_{\mu\nu} = e^{-\frac{\alpha\beta}{q}|\mathbf{A}-\mathbf{B}|^2}. \quad (4.35)$$

En la ecuación (4.32), aparece un término  $\tilde{G}_{\mu\nu}$ , el cual es la matriz de densidad,  $\sum_i \omega_i c_{\mu}^i c_{\nu}^i$ , multiplicada por el factor prexponencial  $G_{\mu\nu}$ .

$$\tilde{G}_{\mu\nu} = \frac{2\pi G_{\mu\nu}}{q} \sum_i^N \omega_i c_{\mu}^i c_{\nu}^i. \quad (4.36)$$

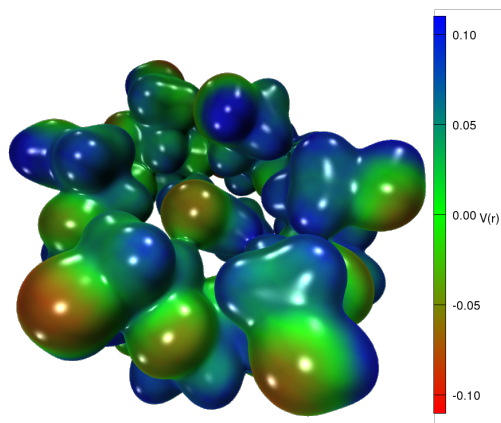
La última integral se resuelve en esta implementación de forma numérica usando un esquema de cuadratura de Gauss-Legendre, así mismo esta está relacionada con la que aparece en las integrales bielectrónicas [62],

$$F_m(T) = \int_0^1 dt t^{2m} e^{-Tt^2}. \quad (4.37)$$

Siendo esta última la función de Boys y depende paramétricamente de un valor  $m$ , mismo que esta relacionado al vector de momento angular de las gaussianas.

Por otro lado, la visualización del potencial electrostático suele ser en algunos casos un poco difícil de interpretar, por lo que generalmente se hace un mapeo de esta cantidad sobre una isosuperficie de densidad electrónica, esta forma de visualizar el potencial electrostático puede verse en la figura 4.5 para un complejo de inclusión de ciclodextrina. Para este ejemplo las zonas azules estan ubicadas sobre átomos de hidrógeno y las aonzas rojas sobre átomos de oxígeno y nitrógeno.





**Figura 4.5:** Potencial electrostático para un complejo de ciclodextrina y formamida,  $\rho(\mathbf{r}) = 0.01 \text{ u.a.}$

### 4.1.3 Análisis de Campos y Búsqueda de Puntos Críticos

#### Búsqueda de puntos críticos de la densidad

Para efectuar esta búsqueda se hace uso de la densidad electrónica, el módulo del gradiente y la matriz de segundas derivadas de la densidad (hessiano), se localizan todos aquellos puntos donde se cumpla la siguiente igualdad,

$$\nabla\rho(\mathbf{r}) = \mathbf{0}. \quad (4.38)$$

Para la hallar los puntos críticos se usa el método de Newton-Raphson dentro de cada cubo que forma la malla, de acuerdo con el siguiente algoritmo

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbb{H}^{-1}|_{\mathbf{r}_i} \nabla\rho(\mathbf{r}_i). \quad (4.39)$$

El esquema para realizar la búsqueda usando CPU-GPU, es realizar una primera búsqueda en la GPU y posteriormente refinarla en el CPU.

Posterior a la localización del punto crítico, este se caracteriza de acuerdo a los valores propios  $\{\lambda_i\}$  de la matriz hessiana, en atractores nucleares y no nucleares (3,-3), puntos críticos de enlace (3,-1), de anillo (3,1) y de caja (3,3). Finalmente, a partir de las coordenadas de los puntos críticos de enlace, se determinan las trayectorias de enlace (*bond paths*), resolviendo la ecuación diferencial asociada.

La salida de la búsqueda de puntos críticos para la densidad genera los siguientes archivos, donde foobar es un nombre genérico.

- ☒ `foobarCrit.log`: Archivo con una descripción de los puntos críticos encontrados, densidad, laplaciano en cada punto, así como también las coordenadas de estos. En el mismo archivo se encuentran separados por tipos, sólo en los puntos críticos de enlace se reporta la elipticidad.

`foobarCrit.log`

```

1 Critical Point      1 of type (3,-1)
2
3 Between the attractors      1 0      and      3 H
4 Euclidian distance between the attractors [A] :    0.96642495
5 Euclidian distance BCP - Attractor( 1) [A] :    0.72915781
6 Euclidian distance BCP - Attractor( 3) [A] :    0.23730635
7 Angle Att1 - BCP1 - Att3      [degrees] :    178.80077789
8
9 Coordinates [A]    -7.19078976E-17  -5.86482199E-01  -3.19208200E-01
10 Density          3.30931259E-01
11 NormGrad        1.06854994E-16
12 Laplacian       -1.44695951E+00
13 Density/Laplacian -2.28708030E-01
14
15 Kinetic Energy G   7.19511795E-02
16 Kinetic Energy K   4.33691058E-01
17 Virial Field      -5.05642237E-01
18 Lagrangian Density 1.80869939E-01
19 KenergyG/Density  2.17420318E-01
20 ELF              9.75542873E-01
21 LOL              8.63354455E-01
22
23 Eigenvalues       -1.43884324E+00  -1.38869738E+00  1.38058111E+00
24 Bond Ellipticity  3.61100015E-02
25 Eta index        1.04220117E+00
26
27 Local Shannon Entropy (density) 3.65958546E-01
28 Disequilibrium      (density) 1.09515498E-01
29 Complexity LMC      (density) 1.57909901E-01
30 Local Shannon Entropy (sigma) 1.12795593E-01
31 Disequilibrium      (sigma) 1.09515498E-03
32 Complexity LMC      (sigma) 1.22591988E-03

```

- ☒ `foobar_critpoints.xyz`: Se trata de un archivo en formato xyz para ser visualizado en cualquier programa que soporte este formato, las unidades de este archivo están dadas en Ångström.
- ☒ `foobar.xyz`: Igual que el archivo anterior, este está escrito en formato xyz. A diferencia del archivo anterior, este contiene las trayectoria de enlace como una serie de puntos.

Para los archivos xyz, se emplea la siguiente nomenclatura.

**BCP:** Para los puntos críticos de enlace (3,-1).

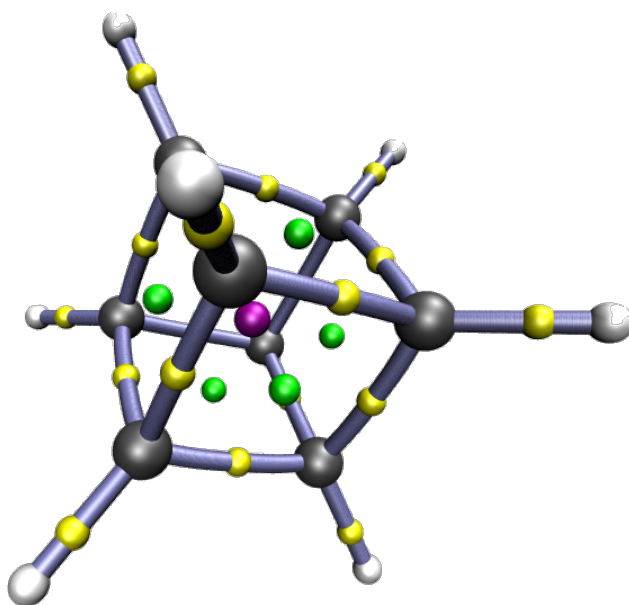
**RCP:** Para los puntos críticos de anillo (3, 1).

**CCP:** Para los puntos críticos de caja (3, 3).

**NNACP:** Para los puntos críticos correspondientes a atractores no nucleares (3,-3).

**BP0001:** Para las trayectorias de enlace, donde la cadena de caracteres 0001 esta relacionada con el número del BCP.

Con la información de `foobar_critpoints.xyz` y `foobar.xyz` se pueden visualizar estos elementos topológico; ejemplo de ello es la figura 4.6, donde pueden observarse los distintos tipos de puntos críticos y trayectorias de enlace.



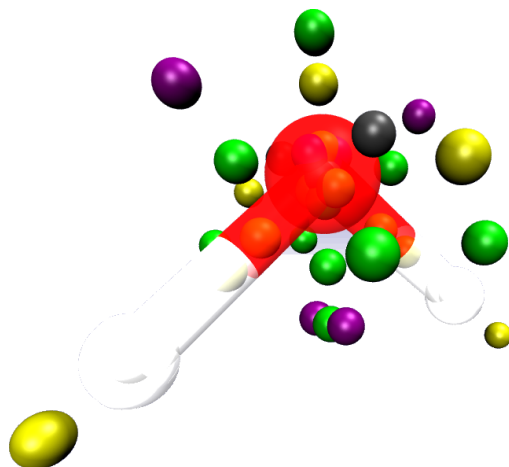
**Figura 4.6:** Puntos críticos de la densidad para la molécula de cubano: en amarillo los puntos críticos de enlace (BCP), en verde los puntos críticos de anillo (RCP), en púrpura un punto crítico de caja (CCP), las trayectorias de enlace en lila.

### Búsqueda de puntos críticos del laplaciano

Para efectuar esta búsqueda se hace uso del laplaciano de la densidad electrónica, el módulo del gradiente del laplaciano y la matriz de segundas derivadas del laplaciano (hessiano). Se localizan todos aquellos puntos donde se cumpla la siguiente igualdad:

$$\nabla(\nabla^2\rho(\mathbf{r})) = \mathbf{0}. \quad (4.40)$$

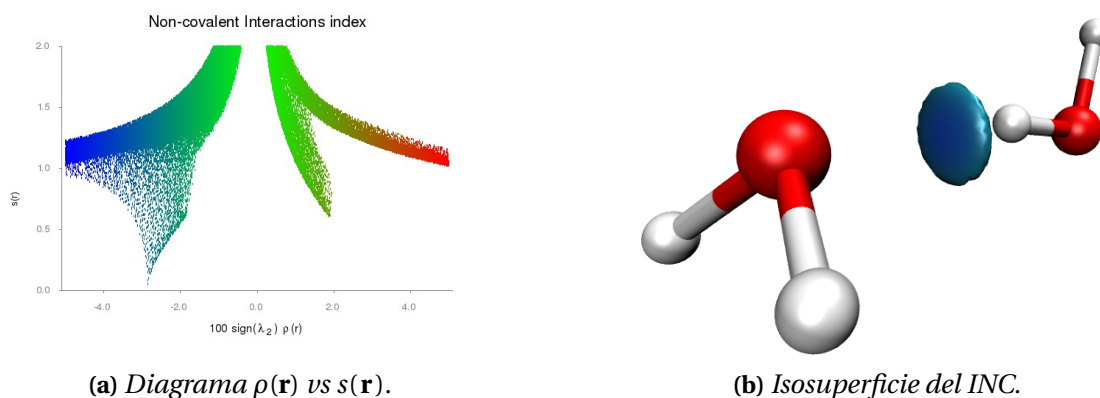
Los puntos críticos para este campo presentan mayor complejidad, generalmente los atractores no nucleares (NNACP) están relacionados con los pares de Lewis, figura 4.7.



**Figura 4.7:** Puntos críticos del laplaciano para la molécula de agua. Visualizado con el código VMD en amarillo los (3,-1), verde (3,1), morado (3,3), negro (3,-3).

### Índice de interacciones no covalentes

Representa de forma visual, aquellas interacciones que corresponden a interacciones débiles. Este análisis, como se mencionó en el marco teórico, corresponde a dos campos. El primero es el gradiente reducido, delimitado a la región  $0 \leq s(\mathbf{r}) \leq 2.0$ , y la densidad en el intervalo  $0 \leq \rho(\mathbf{r}) \leq 0.05$ . Para poder diferenciar entre las zonas atractivas y repulsivas, la densidad electrónica es multiplicada por el segundo valor propio de la matriz hessiana, por lo tanto, es necesario determinar las segundas derivadas y su posterior diagonalización. Como puede verse en la figura 4.8 para un dímero de agua, las zonas negativas corresponden a interacciones atractivas, las zonas positivas a interacciones repulsivas, y la zona intermedia a fuerzas de van der Waals.



**Figura 4.8:** Índice de interacciones no covalentes para el dímero de agua.

GPUAM genera tres archivos, dos con formato cube y uno con formato dat, los dos tipos de archivos están en formato ASCII, por lo que son fácilmente editables y su tamaño depende de la cantidad de puntos que forman el cubo.

- ☒ `foobar.dat`: Este archivo contiene la información necesaria para graficar el diagrama de densidad vs gradiente reducido. Este tiene un formato por columnas que puede ser graficado por `GNUPLOT`, figura 4.8a.
- ☒ `foobar_NCI.cube`: Este archivo contiene el gradiente reducido  $s(\mathbf{r})$ , delimitado a la región  $0 \leq s(\mathbf{r}) \leq 2.0$ .
- ☒ `foobar_den.cube`: Este archivo contiene el valor de la densidad, multiplicado por el segundo valor del hessiano  $\lambda_2$  y por un factor de 100, de tal forma que el intervalo total es  $-5.0 \leq 100\lambda_2\rho(\mathbf{r}) \leq 5.0$ . Se necesitan ambos archivos para obtener isosuperficies como la mostrada en la figura 4.8b.

#### 4.1.4 Ejecución de GPUAM

El formato de ejecución es de forma interactiva, de tal manera que el programa irá solicitando la siguiente información:

1. Nombre del archivo con la función de onda. En el ejemplo que se muestra este corresponde a `../examples/wfx/agua.wfx`.
2. Nombre de salida, sin extensión. En el ejemplo siguiente se escribió `DensidadAgua`.
3. Elección del campo o análisis a efectuar. En este ejemplo, B corresponde a la densidad electrónica, cualquier otro campo necesitará una letra distinta.
4. Si se elige A o F, ambos campos necesitarán información extra sobre qué orbital trabajar, en dado caso ingresar el número del orbital.
5. Elegir la malla, GPUAM cuenta con 5 opciones, malla fina (A), malla mediana (B), malla gruesa (C). La elección (D) es para usar otra cantidad de puntos por unidad atómica, mientras que la opción (E) es para opciones especiales.
  - A, B y C no necesitan parámetros extras.
  - D necesita una variable entera extra, entre mayor sea el valor, más fina será la malla resultante.
  - E para opción especial, en el caso de querer trabajar sobre regiones de la molécula.
    - Coordenadas del punto de inicio (3 valores flotantes): `x0 y0 z0`.

- Puntos por cada dirección (3 valores enteros):  $nx$   $ny$   $nz$ .
  - El espaciamiento de la malla (1 valor flotante):  $h$ .
6. En el caso de tener más de una GPU, será necesario ingresar el número de GPUs con el cual se desea realizar el cálculo.

## Ejecución usando archivos wfn/wfx

```

./Gpuam.x
  Evaluation of fields derived from
  the electron density using GPUs
      GPUAM PROJECT
      UAMI - BUAP
      v. 1.0
      GPU version

Compilation date 04:42:35                      Nov 30 2016
=====
Files with extension wfx, wfn, mgf or aux
agua.wfx
=====
Input filename:
  ../examples/wfx/agua.wfx
=====
Output filename:
  DensidadAgua
=====
Data file GTO
=====
What field ?
(A) ith Molecular orbital
(B) Electron density
(C) Laplacian of the electron density
(D) Module gradient density
(E) Reduced gradient density
(F) Gradient of ith molecular orbital
(G) Gradient electron density
(H) Kinetic energy density
(I) Electron localization function
(J) Localized orbital locator
(K) Search critical points of electron density
(L) Search critical points of laplacian of the electron density
(M) Electrostatic potential
(N) Non-covalent interactions index
(O) Localized electrons detector
(P) Non-covalent interactions and critical points
(Q) Electron density of the basins
Your choice is: B
=====
What mesh?
(A) Fine mesh 10 p/au
(B) Medium mesh 05 p/au
(C) Coarse mesh 01 p/au
(D) Other number of points (int > 1)
(E) Special options
Your choice is: A
=====
Graphics Processing Unit Info
GPUs in your system      :      2

GPU 00  GeForce GTX 760
Global Memory in the GPU : 2047  MB
Clock rate                : 1124  MHz
Multiprocessors           :      6
Maximum number of threads per MP : 2048
Total threads             : 12288

GPU 01  GeForce GT 610
Global Memory in the GPU : 1023  MB
Clock rate                : 1620  MHz
Multiprocessors           :      1
Maximum number of threads per MP : 1536
Total threads             : 1536

```

```

=====
Working files ready
=====
There are 2 available GPUs, how many do you want? 1
=====
Number of GPUs in use : 1
=====
                          System Info
Primitives      :          21
Orbitals        :           5
Nuclei          :           3
Points in X     :          80
Points in Y     :         110
Points in Z     :           91
Total points    :        800800
Memory (MB)     :           3.055
Step            :         0.100000
=====
GPU: 0 grid 5x626 blocks and 16x16 threads
      Points evaluated in the GPU: 800800
=====
Number of CPUs to distribute over GPUs : 1
=====
File [DensidadAgua.cube] was closed successfully

      Elapsed time : 1.000000 s
=====

```

Finalmente, para cálculos que involucran funciones de onda provenientes de MOPAC, la ejecución será la siguiente.

#### Ejecución usando archivos mgf/aux

```

./Gpuam.x
      Evaluation of fields derived from
      the electron density using GPUs
      GPUAM PROJECT
      UAMI - BUAP
      v. 1.0
      GPU version

Compilation date 04:42:35                               Nov 30 2016
=====
Files with extension wfx, wfn, mgf or aux
=====
Input filename:
  ../examples/mgf/agua.mgf
=====
Output filename:
  salida
=====
Data file STO
=====
What field
(A) ith Molecular orbital
(B) Electron density
(C) Laplacian of the electron density
(D) Module gradient density
(E) Reduced gradient density
(F) Search critical points of electron density
(G) Non-covalent interactions index
(H) Localized electrons detector
(I) Gradient of electron density
Your choice is: B
=====
What mesh?
(A) Fine mesh 10 p/au
(B) Medium mesh 05 p/au
(C) Coarse mesh 01 p/au
(D) Other number of points (int > 1)
(E) Special options
Your choice is: A
=====

```

```

Graphics Processing Unit Info
GPUs in your system          :      2

GPU 00  GeForce GTX 760
Global Memory in the GPU    :    2047  MB
Clock rate                   :    1124  MHz
Multiprocessors             :         6
Maximum number of threads per MP :    2048
Total threads                :   12288

GPU 01  GeForce GT 610
Global Memory in the GPU    :    1023  MB
Clock rate                   :    1620  MHz
Multiprocessors             :         1
Maximum number of threads per MP :    1536
Total threads                :    1536

=====
Working files ready
=====
There are 2 available GPUs, how many do you want? 1
=====
Number of GPUs in use :    1
=====
System Info
Primitives      :         6
Orbitals        :         4
Nuclei          :         3
Points in X     :        56
Points in Y     :        46
Points in Z     :        40
Total points    :   103040
Memory (MB)     :     0.393
Step            :     0.100000
=====
GPU: 0 grid 4x115 blocks and 16x16 threads
Points evaluated in the GPU: 103040
=====
Number of CPUs to distribute over GPUs : 1
=====
File [salida.cube] was closed successfully

Elapsed time : 0.000000 s
=====

```

### 4.1.5 Implementación y Rendimiento

El siguiente código es el *kernel* para la evaluación de campos escalares en la GPU, se observa cómo a partir de los índices de los hilos (línea 5), se generan los índices de la malla problema, y con base en estos últimos se generan las componentes  $x$ ,  $y$ , y  $z$  en las líneas 30-32.

**Código 4.1:** Función kernel para la evaluación de campos escalares en la GPU.

```

1 __global__ void Dev_Campo (float h, int *datos, float *iniciales, int *vang, int *ctro,
2                          float *coor, float *eprim, float *nocc, float *coef,
3                          int work, int iesimo, float x0, int puntosx, float *campo) {
4
5     int hilo_global_x, hilo_global_y;
6     int indice_f, indice_x, indice_y, indice_z;
7     int orb, prim;
8     int puntosy, puntosz;
9     float x, y, z, y0, z0;
10
11     orb = datos[0];

```



```

12 prim    = datos[1];
13 puntosy = datos[5];
14 puntosz = datos[6];
15
16 y0 = iniciales[0];
17 z0 = iniciales[1];
18
19 hilo_global_x = threadIdx.x + blockIdx.x * blockDim.x;
20 hilo_global_y = threadIdx.y + blockIdx.y * blockDim.y;
21
22 indice_x = hilo_global_x;
23 indice_y = hilo_global_y/puntosz;
24 indice_z = hilo_global_y%puntosz;
25
26 indice_f = hilo_global_x*puntosy*puntosz + hilo_global_y;
27
28 if (indice_x < puntosx && indice_y < puntosy && indice_z < puntosz) {
29
30     x = x0 + (float) indice_x*h;
31     y = y0 + (float) indice_y*h;
32     z = z0 + (float) indice_z*h;
33
34     campo[indice_f] = field(x,y,z,orb,prim,vang,ctro,coor,eprim,nocc,coef);
35
36 } else {
37     campo[indice_f] = 0.;
38 }
39
40 __syncthreads();
41
42 }

```

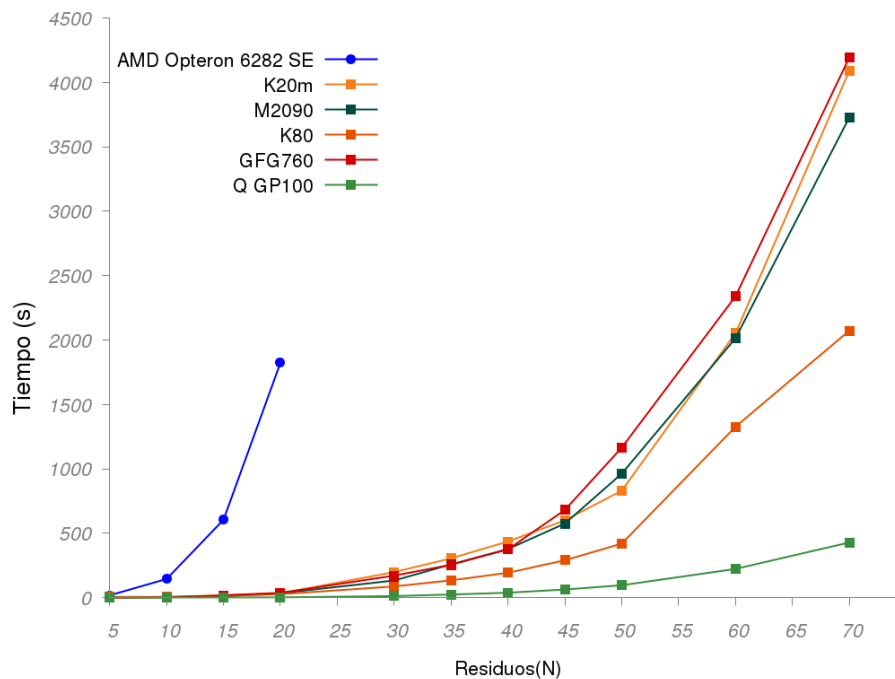
Por otro lado, se midió el rendimiento para la evaluación en CPU *vs* GPU, se seleccionó una familia de polialaninas, es decir unidades de N-residuos de alanina unidas vía enlace peptídico; la determinación de los coeficientes se realizó usando el método PM7 [11] programado en el paquete computacional Mopac [54]. Una vez generados los archivos con la función de onda, se evaluó la densidad electrónica, generando los archivos cube para cada sistema. En la tabla siguiente se reporta el número de residuos, funciones de base, orbitales y puntos evaluados.

Se puede apreciar en la figura 4.9, el crecimiento de los tiempos de cómputo para la evaluación de la densidad electrónica de distintas polialaninas; en azul se reporta el tiempo para CPU (AMD Opteron a 2.6 GHz), el cual tiene un crecimiento considerable, aun para pocos residuos. Se observa que para determinar la 20-polialanina, el tiempo es comparable con el necesario para determinar la 60-polialanina en tarjetas gráficas como K20M, M2090, GFG760. Incluso se puede observar que la tarjeta gráfica K80, pudo evaluar en esa cantidad de tiempo el sistema con 70 residuos. Caso aparte es el rendimiento ofrecido por la tarjeta gráfica Q GP100, que ofrece los menores tiempos de cálculo, esto es debido a que se trata de una GPU más reciente y con mejores capacidades de cómputo. También es importante mencionar que la tarjeta GFG760 es una tarjeta para videojuegos, liberada en el año 2013 y cuenta con 2 GB de memoria RAM. En esta tarjeta gráfica (línea roja), GPUAM tiene un buen desempeño comparado con el equivalente en CPU, por lo que este tipo de GPUs puede ser un buen candidato para realizar

**Tabla 4.2:** Comparación para una serie de polialaninas con  $N$ -residuos, evaluando la densidad electrónica sobre distintas arquitecturas.

N	Func	Orb	Puntos	Tiempo/s					
				K20m	M2090	K80	GFG760	Q GP100	CPU
5	131	74	129729	1	1	7	0	0	19
10	256	144	270336	7	4	6	3	0	146
15	381	214	471744	18	12	14	19	2	605
20	506	284	744480	34	35	28	38	3	1823
30	756	424	1665216	199	134	88	172	13	10526
35	881	492	2076624	306	259	135	256	25	18386
40	1006	564	2755368	437	378	194	376	39	–
45	1131	634	3573648	601	576	291	681	64	–
50	1256	704	4499712	831	963	420	1165	97	–
60	1506	844	7046592	2053	2014	1327	2339	224	–
70	1756	984	10032000	4088	3727	2073	4193	429	–

análisis de la función de onda, si no se cuenta con *hardware* especializado.



**Figura 4.9:** Comparación entre distintas arquitecturas en la determinación de la densidad electrónica.

Otra cantidad donde se analizó el rendimiento de GPUAM fue en la determinación del potencial electrostático, comparándolo con el *software* Multiwfn [63]. Se observó que conforme crece el sistema, la rapidez ver en paréntesis en la Tabla 4.3 es mayor, esto debido a los tiempos

de latencia de subir datos a la tarjeta gráfica y regresarlos al CPU. Para una ciclodextrina, GPUAM es casi 50 veces más rápido.

**Tabla 4.3:** Comparación de velocidad entre Multiwfn y Gpuam para distintos sistemas

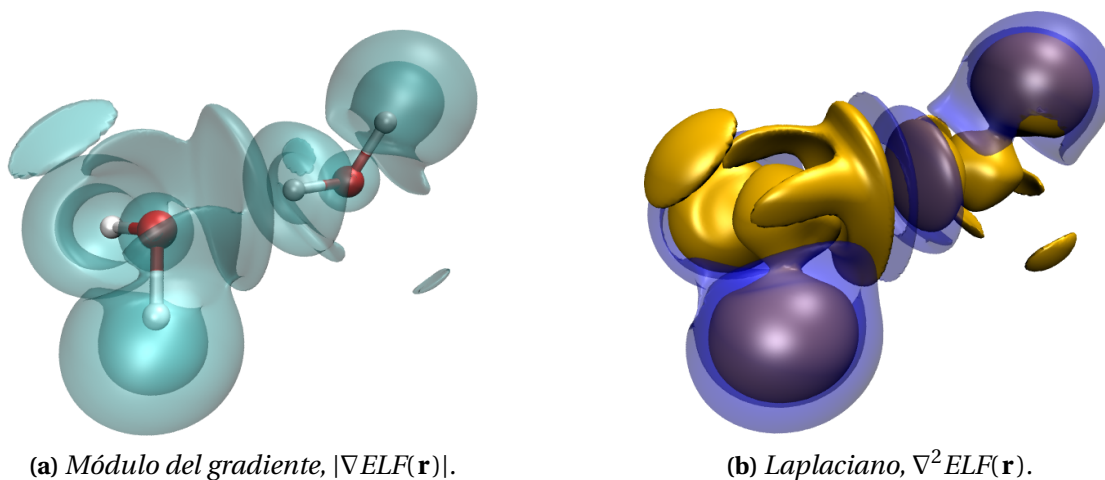
Sistema	Núc	Prim	Puntos	Tiempo/s	
				Multiwfn <sup>†</sup>	GPUAM <sup>‡</sup>
$H_2O \cdots H_2O$	6	80	132 480	7	4 (2x)
$C_2H_5OH \cdots CH_3OH$	15	240	132 440	72	24 (3x)
$(H_2O)_8^-$	24	480	133 632	500	62 (8x)
ciclodextrina + DMF	138	2808	132 240	36 749	754 (49x)

<sup>†</sup> Calculados con Intel(R) Core(TM) i7-4770 CPU @ 3.40 GHz.

<sup>‡</sup> Calculados AMD Opteron(tm) 6282 SE @ 2.6 GHz y GPU Tesla M2090 de 512 cores @ 1.3GHz.

## 4.2 Cube3D

CUBE3D es un programa que determina las primeras y segundas derivadas, de forma numérica, para cualquier campo escalar. Se ha encontrado que esta aproximación es de gran ayuda para analizar sistemas periódicos, donde no se cuenta con la forma analítica del campo escalar, si no a partir de datos en una malla tridimensional, o para campos cuya forma analítica de las primeras y segundas derivadas resulte complicada de evaluar. Por ejemplo, en la figura 4.10 se puede apreciar el módulo del gradiente y el laplaciano, ambos correspondientes a la ELF, para el sistema formado por dos moléculas de agua. Para evaluar numéricamente el gradiente



**Figura 4.10:** Campos para el dimero de agua determinados usando un esquema numérico.

y el laplaciano, se ha empleado un esquema basado en interpoladores de Lagrange [64, 65] en

3 dimensiones. En una dimensión, para generar el interpolador de Lagrange de orden  $n$ , se necesita un conjunto de  $n + 1$  puntos  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ .

$$L(x) = \sum_{j=0}^n y_j \ell_j(x) \quad (4.41)$$

con  $\ell_j$  formando una base polinomial, que cumple con la siguiente ecuación:

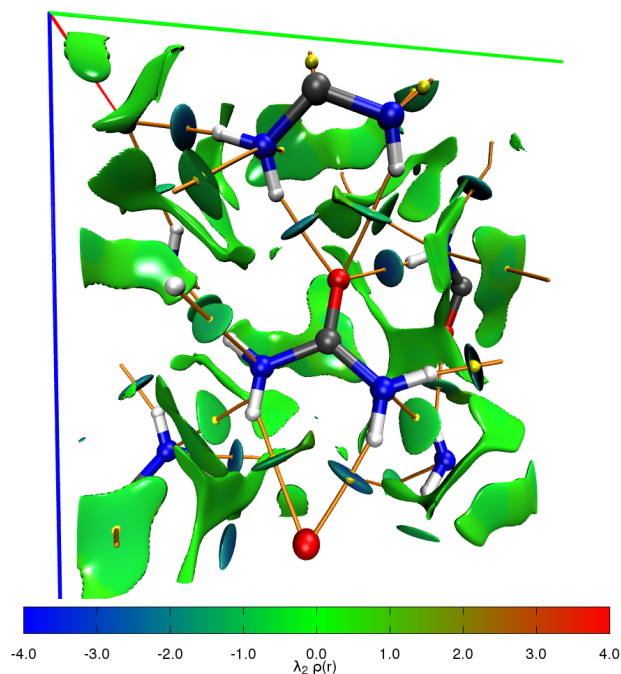
$$\ell_j(x) = \prod_{\substack{i=0 \\ i \neq j}}^n \frac{x - x_i}{x_j - x_i}, \quad (4.42)$$

Esta forma de los polinomios de Lagrange, cumple con la siguiente relación  $\ell_j(x_k) = \delta_{jk}$ , por lo que la función interpolada pasará por el conjunto de puntos que se usó como control. Con estas características podemos aplicarlo a sistemas en tres dimensiones de la forma siguiente:

$$f(\mathbf{r}) = f(x, y, z) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n f_{ijk} \ell_i(x) \ell_j(y) \ell_k(z) \quad (4.43)$$

donde  $f_{ijk}$  es el valor del campo en el índice  $i, j$  y  $k$  de la retícula tridimensional.

A partir de la ecuación (4.43) se pueden determinar las primeras y segundas derivadas necesarias para determinar el gradiente, el gradiente reducido, el laplaciano y el hessiano. También es posible determinar el índice INC y todos estos elementos nos permiten determinar la búsqueda de puntos críticos y trayectorias de enlace, como se muestra en la figura 4.11 donde se ha sobrepuesto el INC, con los puntos críticos de enlace (puntos amarillos) y las trayectorias de enlace (líneas color naranja). Es importante mencionar que la determinación se realizó a partir de un cubo de densidad proveniente de Crystal 14 [66] y lo que se grafica es la celda unitaria para la urea.



**Figura 4.11:** Índice de interacciones no covalentes, puntos críticos de enlace y trayectorias de enlace para un cristal de urea.

Nuevamente vemos que existen dos sistemas de índices, el local  $(i, j, k)$  y el global. Este último relacionado con como ha sido guardado el valor del campo en la memoria RAM,

$$i_{\text{global}} = i N_y N_z + j N_z + k, \quad (4.44)$$

donde  $N_y$  es la cantidad de puntos en el eje  $Y$ , y  $N_z$  en el eje  $Z$ . En la implementación que se ha realizado, el programa recibe como datos de entrada la información del archivo cube, el cual tiene toda la información necesaria del campo, así como la geometría del sistema.

El siguiente es un ejemplo de archivo de entrada para el programa, son importantes las etiquetas que aparecen.

#### Archivo entrada Cube3D

```

1 >> INPUT
2 ejemplo.cube
3 >> OUTPUT
4 salida
5 >> TASK
6 nci
7 >> POLY
8 2
9 >> NCI PROP
10 0.03 1.5
11 >> PERIODIC
12 no

```

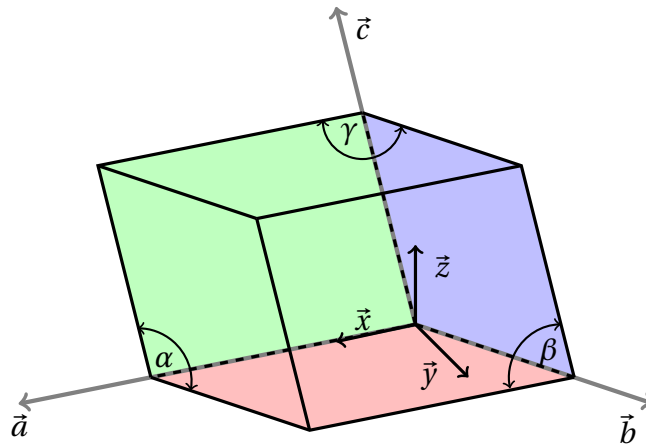
- » **INPUT** Se especifica el nombre del archivo de entrada, en este ejemplo ejemplo.cube, línea 2.
- » **OUTPUT** Aquí se da el nombre genérico para los archivos de salida, la extensión se añade conforme al parámetro de la tarea.
- » **TASK** Es la tarea que se desea realizar.
- **TEST**: Esta bandera para el proceso de depuración, no generará archivos de salida.
  - **GRAD**: Con esta bandera activa, se calculará el gradiente del campo, genera un archivo con formato cube.
  - **LAP**: Si se activa esta bandera, se determinará el laplaciano del campo, genera un archivo con formato cube.
  - **NCI**: Esta bandera determina el índice de interacciones no covalentes, si el campo de origen es la densidad electrónica, genera dos archivos con formato cube, uno para el gradiente reducido y otro para el campo multiplicado por el segundo valor del hessiano. Así mismo, genera archivo data, con información para realizar el mapa bidimensional de INC.
  - **CRIT**: Calcula los puntos críticos del campo especificado, esta tarea genera dos archivos con formato xyz con las coordenadas de los puntos críticos y, en el otro, las trayectorias de enlace, y de forma análoga a GPUAM, genera un archivo log con información extra de los puntos críticos.
- » **POLY** Especifica el orden del polinomio ( $n$ ) que se usará para la interpolación de Lagrange dada por la ecuación (4.43).
- » **NCI PROP** En este apartado, que es opcional, se especifica, con dos valores de tipo flotante, el intervalo donde se analizará tanto el gradiente reducido como la densidad electrónica, en el ejemplo (línea 10) se puede ver cómo están restringidos a los intervalos,  $0 \leq \rho(\mathbf{r}) \leq 0.03$  y  $0 \leq s(\mathbf{r}) \leq 1.5$ , respectivamente.
- » **PERIODIC** Finalmente si al realizar la búsqueda de puntos críticos se tomará un cubo, con condiciones periódicas o si se trata de un sistema aislado.

El programa CUBE3D permite realizar la búsqueda y determinación de campos derivados, tanto de mallas ortogonales como de mallas oblicuas, en estas últimas los ejes no son ortogonales. Para ello es necesario utilizar la matriz de transformación entre el sistema oblicuo y el

ortogonal

$$\begin{pmatrix} x_{ort} \\ y_{ort} \\ z_{ort} \end{pmatrix} = \begin{pmatrix} a & b \cos(\gamma) & c \cos(\beta) \\ 0 & b \sin(\gamma) & c (\cos(\alpha) \csc(\gamma) - \cot(\gamma) \cos(\beta)) \\ 0 & 0 & c \csc(\gamma) \end{pmatrix} \begin{pmatrix} x_{obl} \\ y_{obl} \\ z_{obl} \end{pmatrix}, \quad (4.45)$$

donde  $a$ ,  $b$  y  $c$  son las normas de los ejes oblicuos y  $\alpha$ ,  $\beta$ ,  $\gamma$ , los ángulos que forman los ejes, como puede verse en la figura 4.12.



**Figura 4.12:** Celda unitaria para un sistema oblicuo, pueden verse los ejes que no son ortogonales y los ángulos que forman entre ellos.

### 4.3 Finit1D

Como se mencionó al inicio del capítulo esta aplicación corresponde a dos programas relacionados. FINIT1D permite resolver la ecuación de Schrödinger independiente del tiempo por el método de elemento finito (MEF). El programa la resuelve para distintos potenciales unidimensionales "modelo" con diferentes grados de aproximación en el polinomio de interpolación, las matrices elementales se determinaron previamente con un programa hecho en MATHMATICA [67], el cual puede revisarse en el apéndice B.

En la Tabla 4.4, se reportan las energías obtenidas para los primeros 10 estados del oscilador armónico, en un intervalo  $-6 \leq x \leq 6$ . Para poder efectuar una comparación justa, se mantuvo constante el número de nodos totales, por lo tanto, las matrices usadas en el método son siempre del mismo tamaño, en particular fueron matrices de tamaño  $121 \times 121$ , para ello el número de elementos se varió de acuerdo con la Tabla 4.5.

**Tabla 4.4:** Comparación para distintos ordenes de polinomios de interpolación en el MEF.

Estado	Orden 1	Orden 2	Orden 3	Orden 4	Orden 5	Orden 6	Exacto
1	1.00229128	1.00000415	1.00000005	1.00000000	1.00000000	1.00000000	1
2	3.00478815	3.00002896	3.00000042	3.00000001	3.00000000	3.00000000	3
3	5.00977799	5.00010311	5.00000191	5.00000005	5.00000000	5.00000000	5
4	7.01725609	7.00025895	7.00000598	7.00000017	7.00000001	7.00000000	7
5	9.02721776	9.00052839	9.00001481	9.00000050	9.00000002	9.00000000	9
6	11.03965830	11.00094287	11.00003129	11.00000125	11.00000009	11.00000003	11
7	13.05457323	13.00153358	13.00005915	13.00000294	13.00000044	13.00000032	13
8	15.07195956	15.00233272	15.00010427	15.00000766	15.00000281	15.00000254	15
9	17.09182378	17.00338095	17.00018200	17.00002653	17.00001782	17.00001729	17
10	19.11422163	19.00476366	19.00035321	19.00011568	19.00010100	19.00009999	19

**Tabla 4.5:** Número de elementos para cada grado del polinomio en la aproximación.

	Grado del polinomio					
	1	2	3	4	5	6
Elementos	120	60	40	30	24	20

Puede observarse que para el caso del oscilador armónico, el polinomio de grado 3 permite obtener la energía hasta la octava cifra decimal, mientras que para orden 5 y 6 es prácticamente igual a la energía exacta. Se grafica la función de onda y densidad para el décimo estado de energía 4.13, así como el espectro de energías, con la aproximación de orden 6.

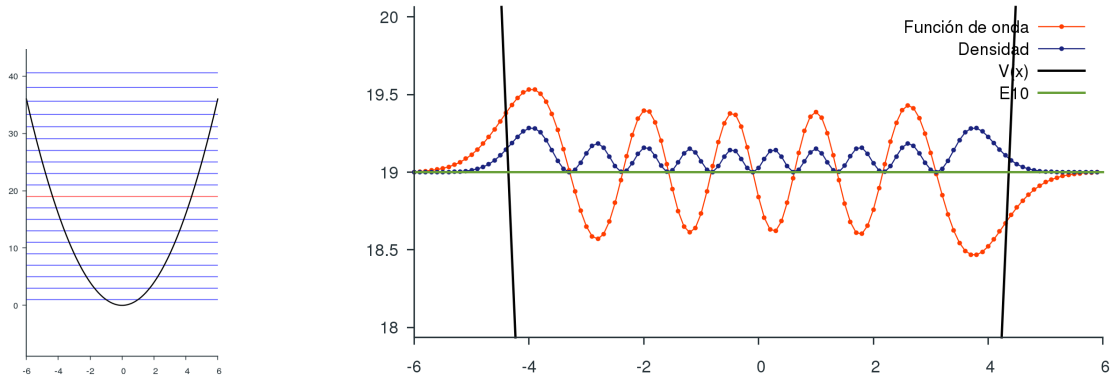
Cabe mencionar que para sistemas como el oscilador armónico, caja de potencial infinito y potencial triangular, las mallas uniformemente espaciadas son las más adecuadas, pero cuando se implementó un potencial coulombico, esta malla no fue la que ofreció el mejor desempeño.

Así, en el método de elemento finito, la elección de la malla correcta resulta vital para el buen desempeño del método. En las siguientes figuras se puede observar la distribución de puntos en el intervalo  $0 \leq x \leq 10$  para una malla uniformemente espaciada (figura 4.14), una malla de Froese-Fischer (figura 4.15) y una malla modificada (figura 4.16) a la que se le nombró de Chebyshev, con un polinomio interpolador de grado 3 y 20 elementos.

Las gráficas en las figuras 4.14, 4.15 y 4.16 fueron obtenidas al resolver la siguiente ecuación diferencial:

$$-\frac{1}{2} \frac{d^2}{dr^2} F(r) + \left( \frac{\ell(\ell+1)}{2r^2} - \frac{Z}{r} \right) F(r) = \varepsilon F(r), \quad (4.46)$$





(a) Espectro de energías.

(b) Función de onda y densidad electrónica.

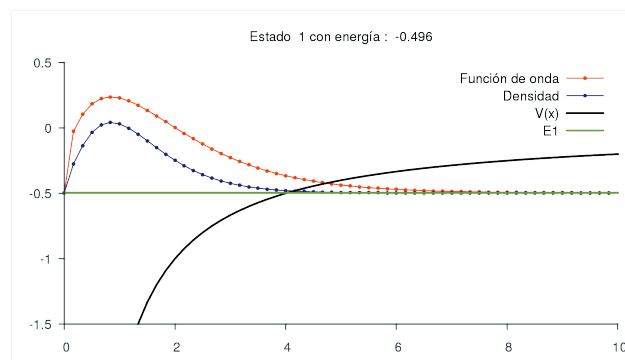
**Figura 4.13:** Décimo estado en energía, determinado con polinomios de orden 6 para el oscilador armónico.

con

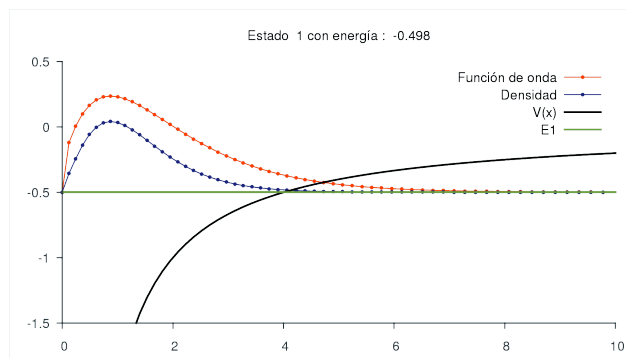
$$F(r) = rR(r), \tag{4.47}$$

donde el potencial de la ecuación (4.46) está dado por un potencial coulombiano y un término centrífugo, que depende de  $\ell$ . Cabe recordar que esta ecuación sólo corresponde a la parte radial, por lo que se tiene la consideración de que el potencial tiene simetría esférica.

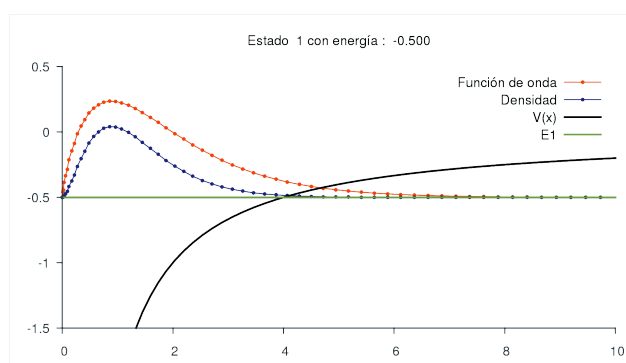
Obteniendo via elemento finito una energía de  $-0.496$  u.a. para una malla uniformemente espaciada,  $-0.498$  u.a. para la malla de Froese-Fischer y  $-0.500$  u.a. para la malla de Chebyshev.



**Figura 4.14:** Malla uniformemente espaciada para el átomo de hidrógeno, realizada con 20 elementos y polinomios de grado 3. La energía obtenida es de  $-0.496$  u.a.



**Figura 4.15:** Malla de Froese-Fischer para el átomo de hidrógeno, realizada con 20 elementos y polinomio de grado 3. La energía obtenida es de  $-0.498$  u.a.



**Figura 4.16:** Malla de Chebyshev para el átomo de hidrógeno, realizada con 20 elementos y polinomio de grado 3. La energía obtenida es de  $-0.500$  u.a.

Como puede verse, para el átomo de hidrógeno, es importante tener mayor cantidad de puntos cerca del origen, esta es la razón por la cual las últimas dos mallas son las que ofrecen mejores resultados.

### 4.3.1 Malla Froese-Fischer

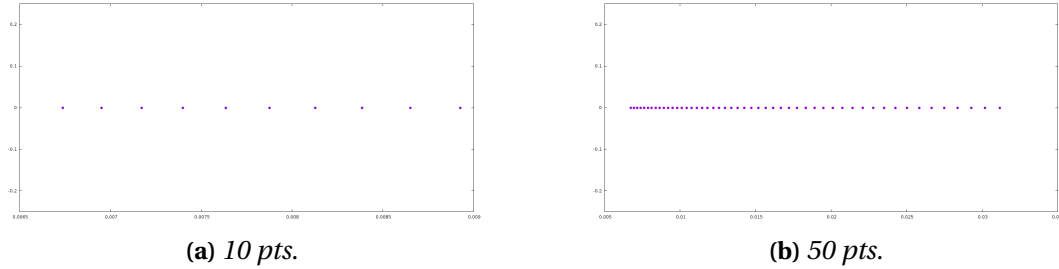
La malla tipo Froese-Fischer [68] tiene la siguiente forma analítica para cada punto.

$$x_i = \frac{1}{z} \exp\left(\frac{i-1}{32} - 5\right), \quad (4.48)$$

donde  $z$  es la carga nuclear.

Cuando se programó este algoritmo de obtención de nodos, se observó que crecía a ritmos distintos para un número dado de puntos, por lo que no la hace del todo adecuada para poder

emplearla en MEF. Esta malla es muy utilizada para sistemas libres, donde la función de onda se extiende hasta infinito, pero no para el caso confinado. En la figura 4.17 se observa el ritmo de crecimiento de esta malla.



**Figura 4.17:** Puntos dados por Froese-Fisher para distintos  $N$ .

Para poder modificar el número de puntos y la extensión del dominio, se realizó una transformación lineal para pasar del dominio de la malla ( $[x_{min}, x_{max}]$ ) al dominio de interés ( $[\xi_{min}, \xi_{max}]$ ), en el cual resolvemos la ecuación de Schrödinger.

$$\xi_i = \alpha x_i + \beta \quad (4.49)$$

con

$$\alpha = \frac{\xi_{max} - \xi_{min}}{x_{max} - x_{min}} \quad \text{y} \quad \beta = \frac{\xi_{min}x_{max} - \xi_{max}x_{min}}{x_{max} - x_{min}} \quad (4.50)$$

### 4.3.2 Malla Chebyshev

La malla de Chebyshev original es usada para la resolución numérica de integrales sobre dominios seminfinitos. En [69, 70] la variable radial ( $0 \leq r < \infty$ ) es transformada a un intervalo finito ( $-1 \leq x \leq 1$ ).

$$r = r_m \left( \frac{1+x}{1-x} \right). \quad (4.51)$$

En esta ecuación  $r_m$  es un parámetro del tamaño atómico, el cual está relacionado con un medio del radio de Bragg-Slater para cada átomo [71], por otro lado, la variable  $x$  es entonces discretizada usando puntos para una malla de cuadratura de Gauss-Chebyshev.

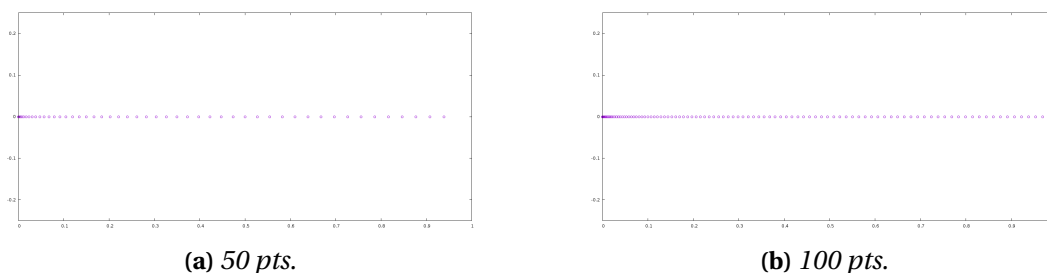
$$x_i = \cos \left( \frac{\pi i}{N+1} \right). \quad (4.52)$$

La malla que está implementada en el programa de elemento finito usa la siguiente forma modificada de los puntos de Chebyshev

$$x_i = 1 - \cos\left(\frac{\pi i}{2(N+1)}\right). \quad (4.53)$$

Como puede verse en las figuras 4.18, el conjunto de  $\{x_i\}$  se encuentran en el intervalo  $0 \leq x < 1$ , para poder utilizarlo en MEF; se realizó una transformación lineal exactamente igual a la descrita anteriormente para la malla de Froeser-Fisher.

$$\xi_i = \gamma x_i + \delta. \quad (4.54)$$



**Figura 4.18:** Puntos dados por Chebyshev para distintos  $N$ .

### 4.3.3 Átomos Hidrogenoides y Distintos Confinamientos

Los sistemas cuánticos al ser restringidos espacialmente presentan nuevas propiedades que difieren del caso libre. Por ejemplo, para un sistema atómico libre, los electrones tendrán una distinta distribución en el espacio que el correspondiente sistema confinado y, si las reacciones químicas las analizamos desde el punto de vista electrónico, esta redistribución de la densidad repercutirá de forma directa en la reactividad [72].

Por otro lado, sabemos que existe una relación entre volumen y presión y al disminuir el volumen implica que la presión externa sobre el sistema crece. Fue en 1935 cuando E. Wigner y H. B. Huntington predijeron la existencia de hidrógeno [73] en una forma alotrópica con características de metal. Se sabe que éste se encuentra a altas presiones, similares a las presentes en los núcleos de planetas, de esta forma, la caracterización de esta forma exótica de hidrógeno se ha ganado el papel del *Santo Grial* de la física de altas presiones, mismas que podemos analizar en los modelos de confinamiento cuántico.

### 4.3.3.1 Confinamiento Esférico por Paredes Impenetrables

Se aplicó el método de elemento finito, para determinar las energías para distintos tipos de confinamientos, siendo el primero de ellos, lo que se conoce como confinamiento de paredes duras o impenetrables. La ecuación de Schrödinger asociada a dicho problema es

$$-\frac{1}{2} \frac{d^2 F(r)}{dr^2} + V_c(r)F(r) = \varepsilon F(r) \quad \text{con} \quad F(r) = rR(r), \quad (4.55)$$

donde el potencial queda definido como:

$$V_c(r) = \begin{cases} \frac{\ell(\ell+1)}{2r^2} - \frac{Z}{r} & r < R_c, \\ \infty & r \geq R_c. \end{cases} \quad (4.56)$$

siendo  $R_c$  lo que se conoce como radio de corte. Si  $R_c$  tiende a infinito, se tiene el caso libre. Para este potencial se compara la implementación hecha con el MEF, en el programa FINIT1D, con los datos obtenidos por Rodríguez-Bautista y colaboradores [74]. Para ello se discretizó el espacio en 100 elementos con polinomios de grado 5, obteniendo los resultados de la tabla 4.6.

**Tabla 4.6:** Comparación de la energía para el estado basal del átomo de hidrógeno, para distintos radios de corte ( $R_c$ ) y un confinamiento de paredes impenetrables. En la segunda columna se muestran resultados usando bases localizadas (tomadas de la referencia [74]) y la tercera columna los resultados de este trabajo.

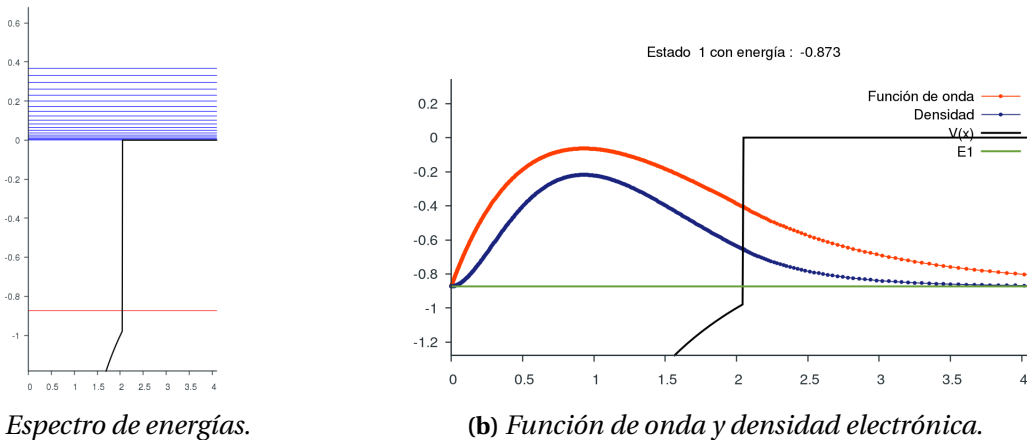
$R_c/\text{u.a.}$	BSIII/Ry	MEF/Ry
0.72288	11.9110	11.9048
0.83155	8.1728	8.1690
0.85089	7.6638	7.6603
0.96240	5.3470	5.3449
1.00000	4.7498	4.7479
1.25921	2.0989	2.0984
1.43831	1.1213	1.1210
2.04918	-0.3103	-0.3103
2.51847	-0.6793	-0.6793
3.15412	-0.8797	-0.8797
3.45203	-0.9233	-0.9234
4.08889	-0.9707	-0.9708
4.87924	-0.9913	-0.9913
5.77827	-0.9979	-0.9979

### 4.3.3.2 Confinamiento Esférico por Paredes Penetrables

Este confinamiento de paredes duras es el más sencillo de implementar y de resolver pero la naturaleza dista mucho de presentar este tipo de delimitaciones espaciales. Un modelo más cercano a la realidad es aquel con paredes penetrables donde la ecuación diferencial a resolver es (4.55), pero modificando el potencial de confinamiento a:

$$V_c(r) = \begin{cases} \frac{\ell(\ell+1)}{2r^2} - \frac{Z}{r} & r < R_c, \\ \frac{\ell(\ell+1)}{2r^2} - U_0 & r \geq R_c. \end{cases} \quad (4.57)$$

Aquí el potencial está regulado por una pared con potencial dado por  $U_0$ . Un caso particular es cuando  $U_0 = 0$ , el cual simula un confinamiento dado por un cascarón esférico que se encuentra conectado a tierra. Este tipo de confinamiento presenta un escalón en  $R_c$ , por lo que la malla se adecuó para tener la mayor cantidad de puntos en la posición del núcleo como lo indica el término coulombico y también se aumentó la densidad de puntos en torno al escalón, figura 4.19.



**Figura 4.19:** Estado basal para el átomo de hidrógeno confinado por paredes penetrables y potencial  $U_0 = 0$  en  $R_c = 2.04918u.a.$

Como se mencionó anteriormente,  $U_0 = 0$  es un caso particular, el cual tiene solución analítica, la cual está reportada por Ley-Koo y colaboradores [75]. Se comparan las energías para distintos grados de confinamiento usando bases localizadas y optimizadas en [74], el método de elemento finito del presente trabajo y la solución analítica dada por Ley-Koo.

**Tabla 4.7:** Comparación de la energía para el estado basal del átomo de hidrógeno, para distintos radios de corte ( $R_c$ ), y un confinamiento de paredes penetrables ( $U_0 = 0$ ).

$R_c/\text{u.a.}$	BSI/Ry	MEF/Ry	Exacto/Ry
0.72288	0.0001	0.0003	0.0000
0.83155	-0.0617	-0.0619	-0.0625
0.85089	-0.0807	-0.0810	-0.0816
0.96240	-0.2048	-0.2057	-0.2066
1.00000	-0.2479	-0.2491	-0.2500
1.25921	-0.5069	-0.5094	-0.5102
1.43831	-0.6364	-0.6393	-0.6400
2.04918	-0.8694	-0.8731	-0.8734
2.51847	-0.9394	-0.9424	-0.9426
3.15412	-0.9785	-0.9802	-0.9803
3.45203	-0.9868	-0.9881	-0.9881
4.08889	-0.9954	-0.9960	-0.9960
4.87924	-0.9988	-0.9990	-0.9990
5.77827	-0.9998	-0.9998	-0.9998

#### 4.3.3.3 Confinamiento Esférico por un Dieléctrico Continuo

Otro confinamiento que se exploró con el MEF fue el confinamiento dado por un dieléctrico continuo, el cual surge al estudiar los modelos de disolvente como casos especiales de confinamiento cuántico. Es en el dieléctrico donde se genera una cavidad y dentro de ella se localizará el soluto, en este caso, un átomo hidrogenoide. Usualmente, en este modelo, se usan conjuntos de base que han sido previamente optimizados para el caso libre, por lo que estas bases no tienen la información de la cavidad y el comportamiento asintótico no será el adecuado, pues la perturbación provocada por el confinamiento cambia sustancialmente las propiedades del sistema.

La ecuación a resolver es la ecuación (4.55), con el potencial dado por la siguiente expresión:

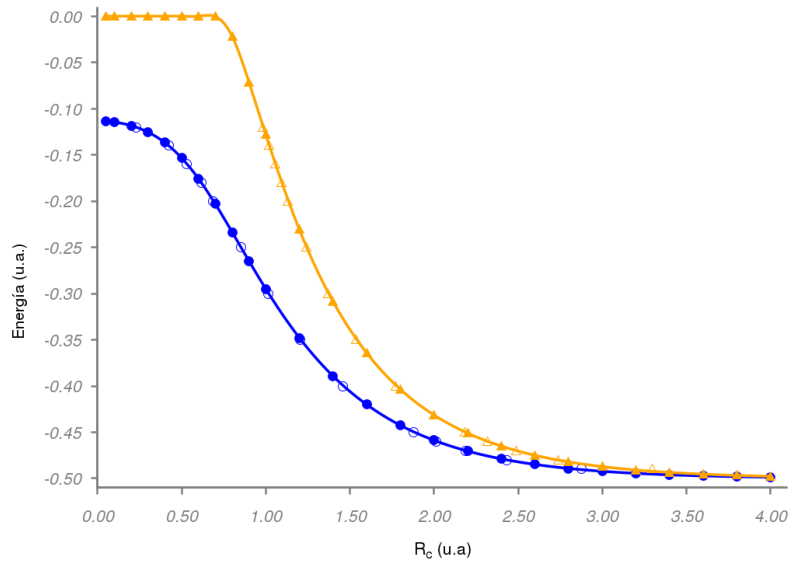
$$V_c(r) = \begin{cases} \frac{\ell(\ell+1)}{2r^2} - \frac{Z}{r} & r < R_c, \\ \frac{\ell(\ell+1)}{2r^2} - \frac{Z}{\epsilon r} & r \geq R_c, \end{cases} \quad (4.58)$$

donde  $\epsilon$  representa la permitividad, también llamada constante dieléctrica, relativa a un medio. Los datos presentados a continuación corresponden a dos valores de  $\epsilon$  representativos, teflón con  $\epsilon = 2.1$  y agua con  $\epsilon = 80.0$ . Los resultados exactos fueron determinados por Martínez-Sánchez y colaboradores [76]. Cabe mencionar que en el desarrollo de Martínez-Sánchez, igual que en el de Ley-Koo [75], se propone un valor de energía y se obtiene el radio de confinamiento

( $R_c$ ) asociado a dicha energía, caso contrario en el MEF donde se propone un radio y se obtiene la energía asociada, por lo que la comparación no es posible realizarla en igualdad de términos. Para ello, lo que se realizó fue calcular un espectro de energías para distintos radios de confinamiento y sobreponer los datos de la referencia [76].

En la figura 4.20 se representa la energía para el estado basal del átomo de hidrógeno confinado por un dieléctrico continuo. Los círculos corresponden a una constante dieléctrica de  $\epsilon = 2.1$ , asociada al teflón, mientras que los triángulos corresponden a la constante dieléctrica del agua ( $\epsilon = 80.0$ ). Los puntos vacíos corresponden al valor exacto, mientras que los puntos y líneas pertenecen al MEF, las cuales se determinaron con 100 elementos y polinomios interpoladores de grado 5.

Puede observarse que conforme el radio de confinamiento ( $R_c$ ) se incrementa, el valor de la energía tiende al valor del caso libre ( $E = 0.5$  u.a.). Por otro lado, el potencial del dieléctrico con valor de  $\epsilon$  igual al del agua, ioniza el átomo de hidrógeno para radios menores a una unidad atómica. Por otro lado, el potencial para el dieléctrico teflón, no ioniza al átomo de hidrógeno. Para comprobar esta observación, se determinó el valor esperado de  $\langle r \rangle$  y se graficó en la figura 4.21.

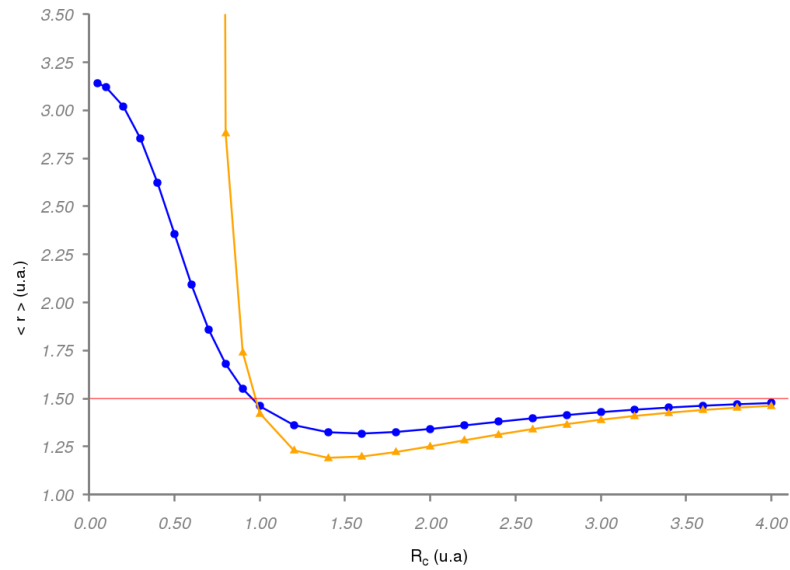


**Figura 4.20:** Energía del estado basal para el átomo de hidrógeno, a distintos radios, los círculos corresponden a una constante dieléctrica de 2.1, mientras los triángulos a la constante dieléctrica de 80.0, las figuras llenas corresponden a datos determinados via MEF mientras las figuras vacías a su contraparte exacta.

Nuevamente los círculos corresponden a un valor de 2.1 en la constante dieléctrica y los

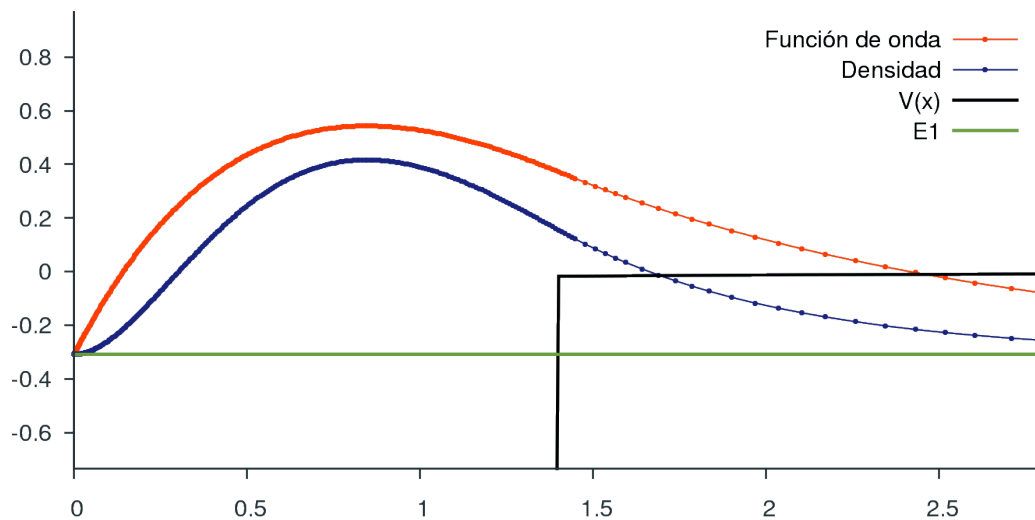


triángulos a 80.0. Igualmente se observa que, a radios grandes, la energía y el valor esperado de  $r$  convergen al caso libre. Donde el valor esperado de  $r$  para el sistema libre es igual a  $3/2$  u.a.



**Figura 4.21:** Valor esperado  $\langle r \rangle$  para el átomo de hidrógeno a distintos radios, los círculos corresponden a una constante dieléctrica de 2.1, mientras los triángulos a la constante dieléctrica de 80.0, la línea horizontal corresponde al valor esperado para el caso libre.

La densidad electrónica y el potencial puede verse en la figura 4.22, que corresponde al valor de la constante dieléctrica asociada al agua.



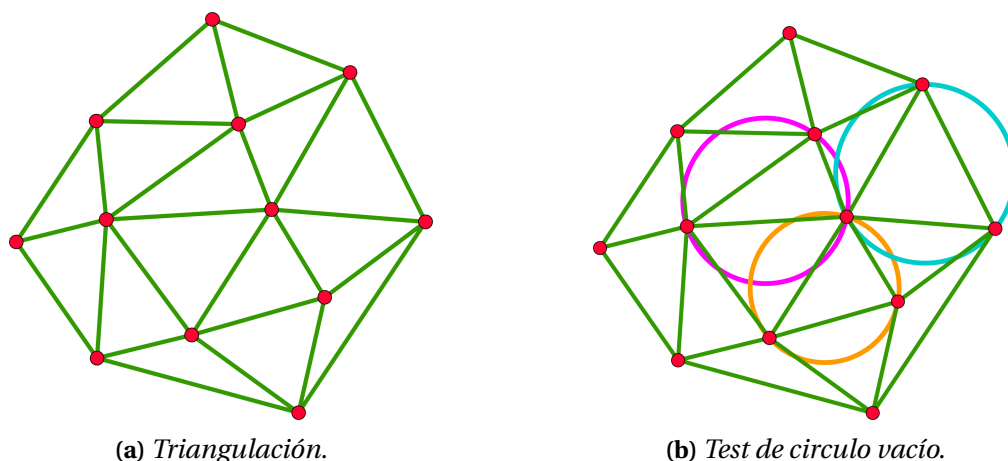
**Figura 4.22:** Función de onda y densidad electrónica para el estado basal para del átomo de hidrógeno confinado por un dieléctrico, con constante dieléctrica de 80.0.

## 4.4 Finit2D

En el caso de confinamiento a nivel atómico existe una gran cantidad de trabajos [72, 77–80], pero son relativamente pocos para el caso molecular. Para moléculas diatómicas se ha realizado el análisis de forma analítica con los núcleos en los focos de elipsoides para  $H_2^+$ ,  $HeH^{++}$  [81] y de forma variacional para sistemas diatómicos con dos electrones [82, 83]. Estos últimos trabajos, de R. Colín y S. Cruz desacoplan las coordenadas de los núcleos de los focos, pudiendo variar de manera independiente la forma de la cavidad confinante y la distancia internuclear.

Debido al problema de optimización de la base, que dependerá del confinamiento dado así como también de la distancia internuclear, se ha explorado resolver la ecuación de Schrödinger de forma numérica usando las herramientas ofrecidas por el método de elemento finito, se han codificado dos programas, uno que hace uso de bases localizadas, el programa CLAUSUS y otro que implementa el método de elemento finito en dos dimensiones FINIT2D.

FINIT2D corresponde a una subrutina en C, que dada la geometría confinante y las posiciones nucleares obtiene la triangulación de Delaunay. El método propuesto es una triangulación incremental, donde se añaden puntos y se genera la conectividad con el nuevo punto añadido, así hasta terminar con los puntos.



**Figura 4.23:** *Triangulación de Delaunay para 12 puntos.*

La parte medular consiste en, dado 4 puntos, realizar el test de círculo vacío, o también llamado en geometría computacional test *InCircle*, el cual está relacionado con la condición de Delaunay, la cual establece que para un triángulo, la circunferencia circunscrita del mismo no debe contener ningún otro vértice de la triangulación en su interior [27, 28].

El código para la generación de la malla (malla no estructurada) está formado por las si-

guientes rutinas principales:

**TriangulacionIncremental:** Función principal de la triangulación.

**inserDelaunay:** Realiza una triangulación local, dentro de un triángulo mayor.

**changDelaunay:** Obtiene el triángulo sobre el que se añadirán puntos extras.

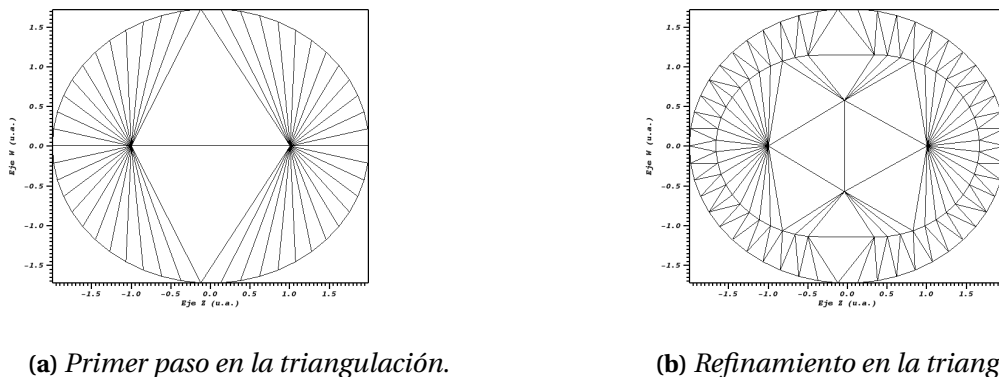
**RefinaMesh:** Refina la triangulación añadiendo puntos en el vértice de Voronoi.

El paso posterior es un refinamiento por inserción en el vértice de Voronoi, debido a que la triangulación de Delaunay y el diagrama de Voronoi están íntimamente relacionados. Y se sabe que, añadir un nodo en el vértice de Voronoi mantiene las propiedades globales de la triangulación.

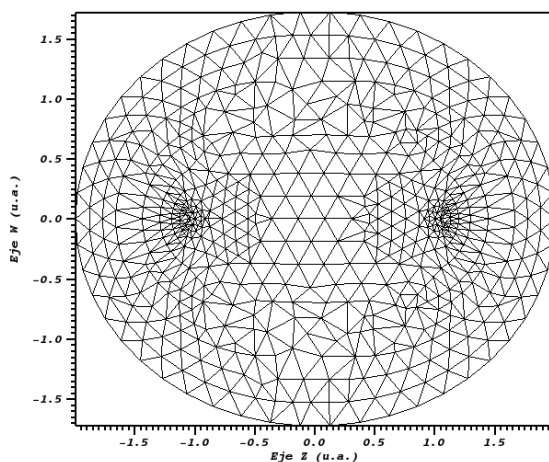
Obviamente, la triangulación deberá de ser más refinada en puntos cercanos a los núcleos, para ello se ha propuesto un esquema aparte del de Voronoi, uno que minimice las áreas de los triángulos dentro de la zona internuclear y nuclear.

Los pasos dentro del programa, para la generación de la malla en 2D son:

- Dada la geometría y posiciones nucleares se obtiene la primera triangulación. En este punto ya tenemos los nodos Dirichlet, como se muestra en la figura 4.24a.
- Se refina la triangulación tomando en cuenta la distancia a los núcleos y la carga nuclear, figura 4.24b.
- Se continúa refinando la triangulación sólo en aquellos triángulos que sea necesario, dadas las condiciones de distancia al núcleo y carga nuclear, hasta obtener la triangulación final, figura 4.25.
- En el último punto de la triangulación se tiene la matriz de ligadura (matriz necesaria para ensamblar la matrices), en este punto se sabe cuáles son nodos internos y externos, así como también los triángulos frontera.
- Con la información de la matriz de ligadura y la forma analítica de las integrales involucradas, se ensamblan las matrices.
- Se resuelve el problema de valores propios. Para ello el programa hace uso de la biblioteca LAPACKE.



**Figura 4.24:** Proceso de triangulación de Delaunay.



**Figura 4.25:** Último paso de la triangulación.

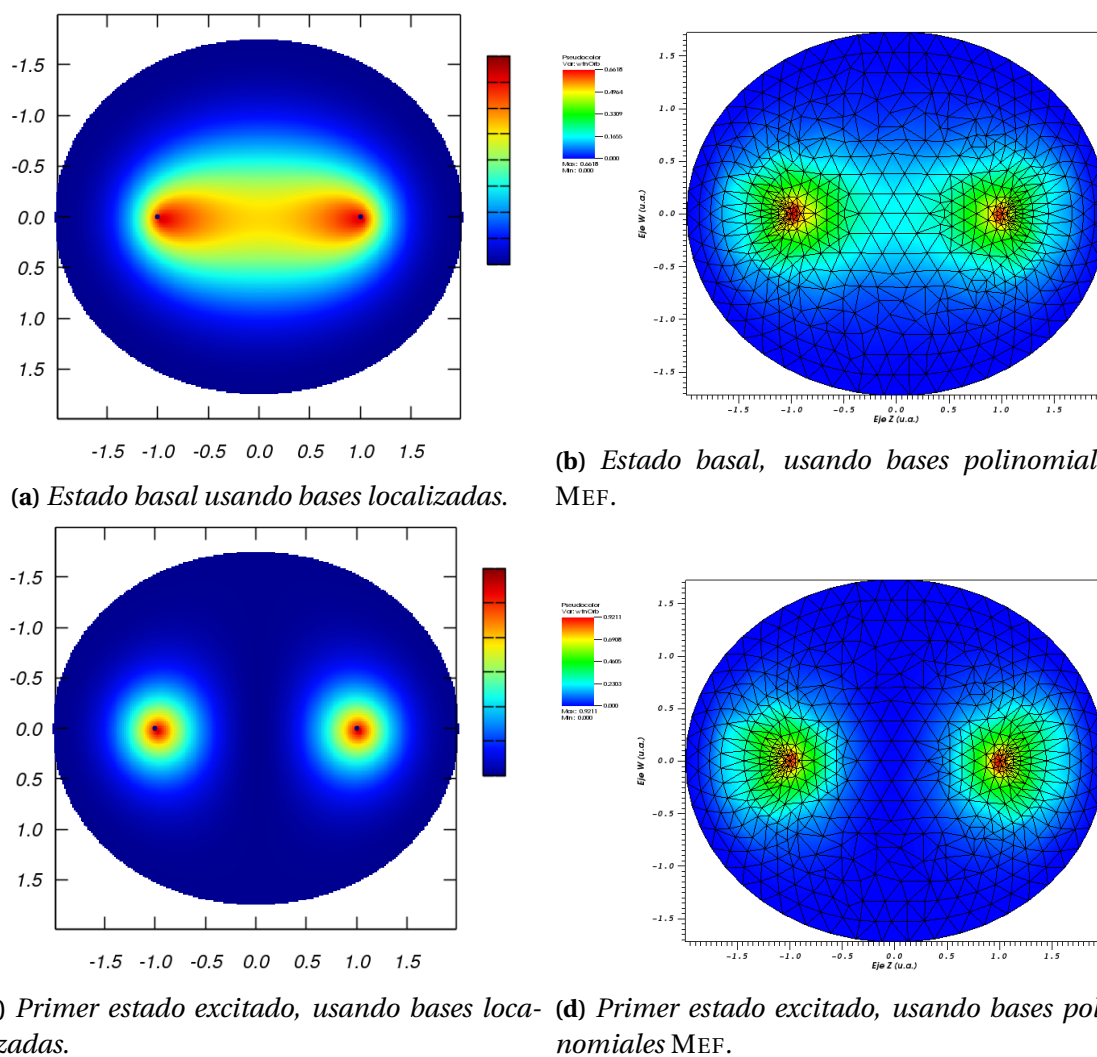
Teniendo la triangulación, y guardada con un formato vtk, se ha resuelto la ecuación de Schrödinger para el ion molecular  $H_2^+$ , el cual consta de un sólo electrón y de dos núcleos, el valor exacto es reportado en [81] y, con bases localizadas en el programa CLAUSUS, también se reporta la densidad electrónica en el centro geométrico ( $\mathbf{r}_C$ ).

La comparación entre los valores de la densidad electrónica, entre las funciones de base localizadas (CLAUSUS) y las funciones polinomiales (FINIT2D), se puede ver en la figura 4.26.

Conforme el sistema se confina, la densidad electrónica migra de los núcleos a la zona internuclear, generando un máximo donde había un punto de silla de primer orden, dicho de otra forma, un punto crítico de enlace por efecto de la presión se transformó en un punto crítico no-nuclear, figura 4.27.

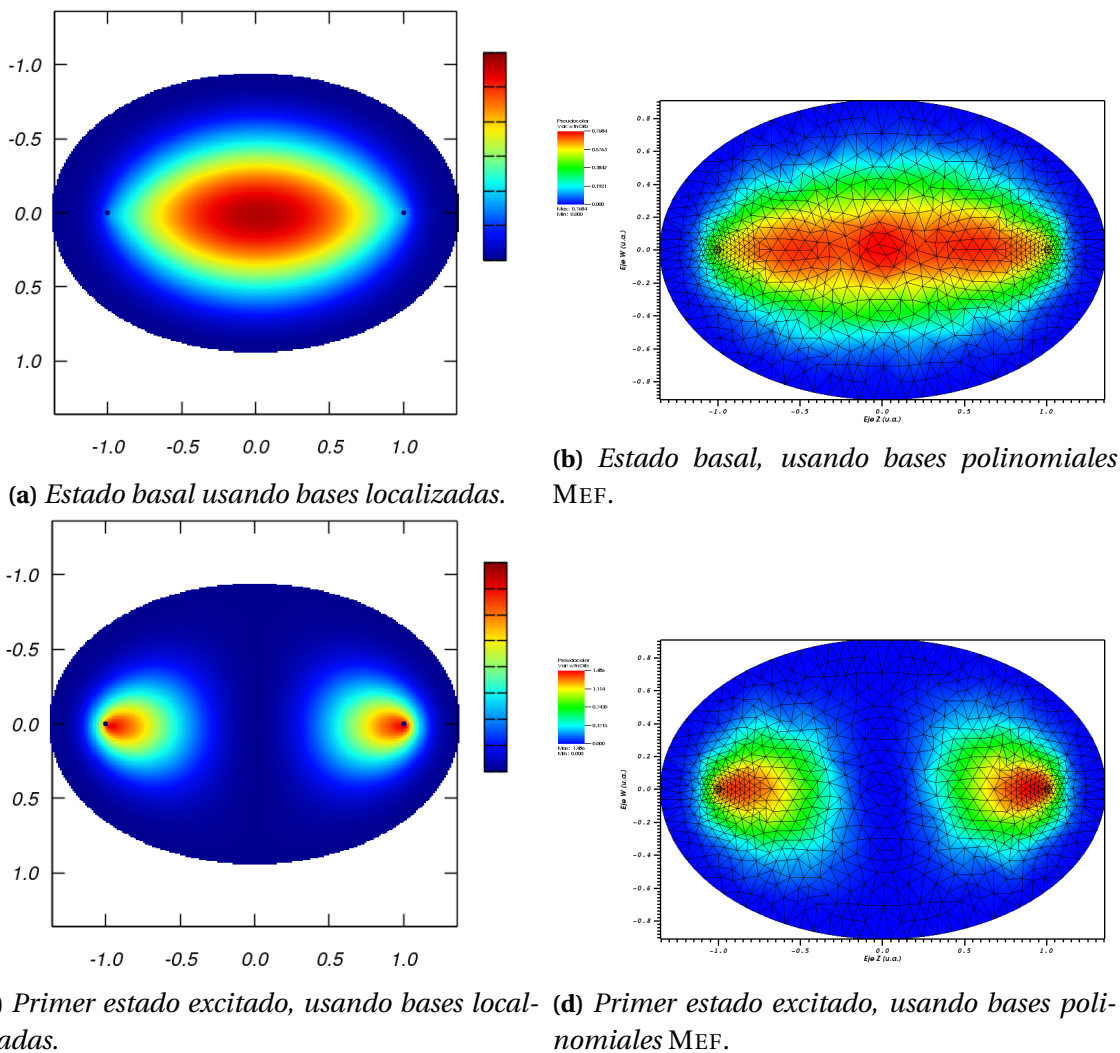
**Tabla 4.8:** Comparación energética para el confinamiento de  $H_2^+$ , a distintos niveles de  $\xi_0$ , la distancia internuclear fue de 2.0 u.a.

$\xi_0$	$E_{exacta}$	$E_{aprox}$	$\rho(\mathbf{r}_c)$
2.2237	-0.75	-0.7499	0.196
1.9934	-0.50	-0.4999	0.258
1.7434	0.025	0.0258	0.395
1.6150	0.5	0.5025	0.525
1.3621	2.5	2.5214	1.134



**Figura 4.26:** Densidad para los primeros dos estados de la molécula ion  $H_2^+$ , usando dos metodologías distintas, para un confinamiento elipsoidal de  $\xi_0 = 1.9934$  u.a.

A diferencia del programa FINIT1D, FINIT2D consta principalmente de la triangulación, y sólo se ha programado polinomios de interpolación lineal, una posterior mejora será incrementar el orden del polinomio, pero es interesante ver cómo aún se observan tendencias similares a



**Figura 4.27:** Densidad para los primeros dos estados de la molécula ion  $H_2^+$ , usando dos metodologías distintas, para un confinamiento elipsoidal de  $\xi_0 = 1.3621$  u. a.

las encontradas con bases localizadas sobre los núcleos.

---

---

## CAPÍTULO 5

---

### APLICACIONES

A continuación se presentan algunas aplicaciones donde se ha usado GPUAM, ya sea para responder algunas interrogantes o para brindar información extra sobre la naturaleza de distintas interacciones químicas. Esto se realizó usando la teoría de átomos en moléculas [16–18] y el índice de interacciones no covalentes [22, 23].

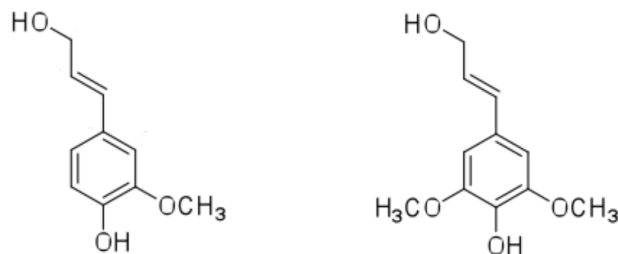
Cabe mencionar que estas aplicaciones, fueron realizadas como apoyo a los investigadores Pablo López-Albarran (Interacción entre oligolignoles y celulosa) y José Eduardo Barrios-Vargas (Interacción oro y nucleótido).

#### **5.1 Interacciones entre oligolignoles y celulosa**

El diseño de adhesivos para la industria de la madera ha migrado a los adhesivos amigables para el ambiente, donde la lignina es de los materiales más estudiados. Ésta es una clase de polímero orgánico, presente en los tejidos de soporte de las plantas y junto a la celulosa, es uno de los componentes principales de la madera. Las ligninas presentan un elevado peso molecular dado que resultan de la unión de varios alcoholes fenilpropiólicos. En este trabajo se analizaron las in-

teracciones entre distintos oligolignoles y la celulosa; teniendo como hipótesis que el oligolignol con mayor cantidad de interacciones no covalentes, será el que mejor funcione como posible adhesivo.

Se analizarán cinco complejos formados por un oligolignol y celulosa. De los cuales, tres corresponden a trilignoles y dos a dilignoles, siendo las unidades básicas de estos el coniferol (CA) y sinapol (SA), cuya estructura puede verse en la figura 5.1.



**Figura 5.1:** Estructura de dos lignoles: izquierda coniferol, derecha sinapol.

La nomenclatura de los oligolignoles puede resultar confusa, sin embargo a continuación se enuncia el nombre con el cual han sido referenciados en la siguiente publicación [84] y un nombre abreviado para este escrito, vease la Tabla 5.1.

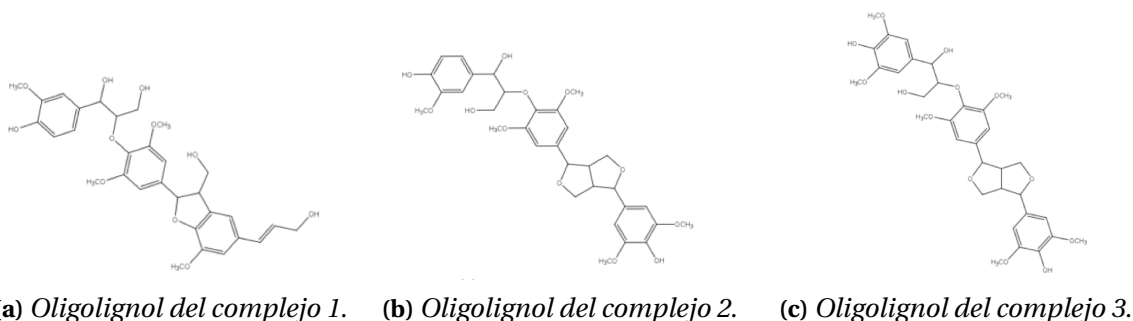
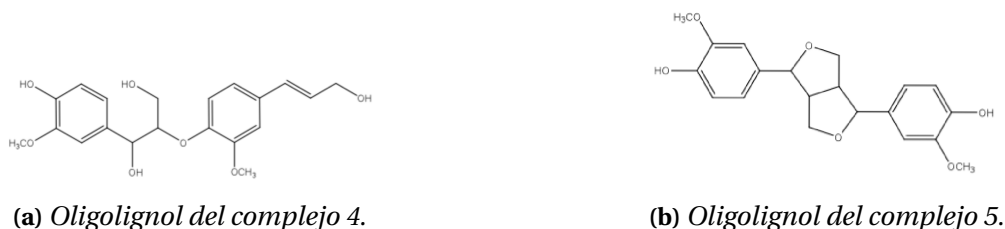
**Tabla 5.1:** Nomenclatura de este trabajo para los complejos oligolignol-celulosa.

<b>Trilignoles</b>	
Nombre del oligolignol en [84]	Nombre en este trabajo
CA_βO4_SA_β5_CA	Complejo 01
CA_βO4_SA_ββ_SA	Complejo 02
SA_βO4_SA_ββ_SA	Complejo 03
<b>Dilignoles</b>	
CA_βO4_CA	Complejo 04
CA_ββ_CA	Complejo 05

Las estructuras químicas de los trilignoles estudiados pueden verse en la figura 5.2, mientras que para los dilignoles en la figura 5.3.

A partir de estructuras del complejo oligolignol-celulosa que previamente fueron optimizadas, se obtuvo el archivo con la información de la función de onda usando el paquete computacional Gaussian [50], empleando para ello el funcional M06-2x [85] y el conjunto de base cc-pVDZ dentro del marco de la TFD. Con esta información se efectuó la búsqueda de puntos



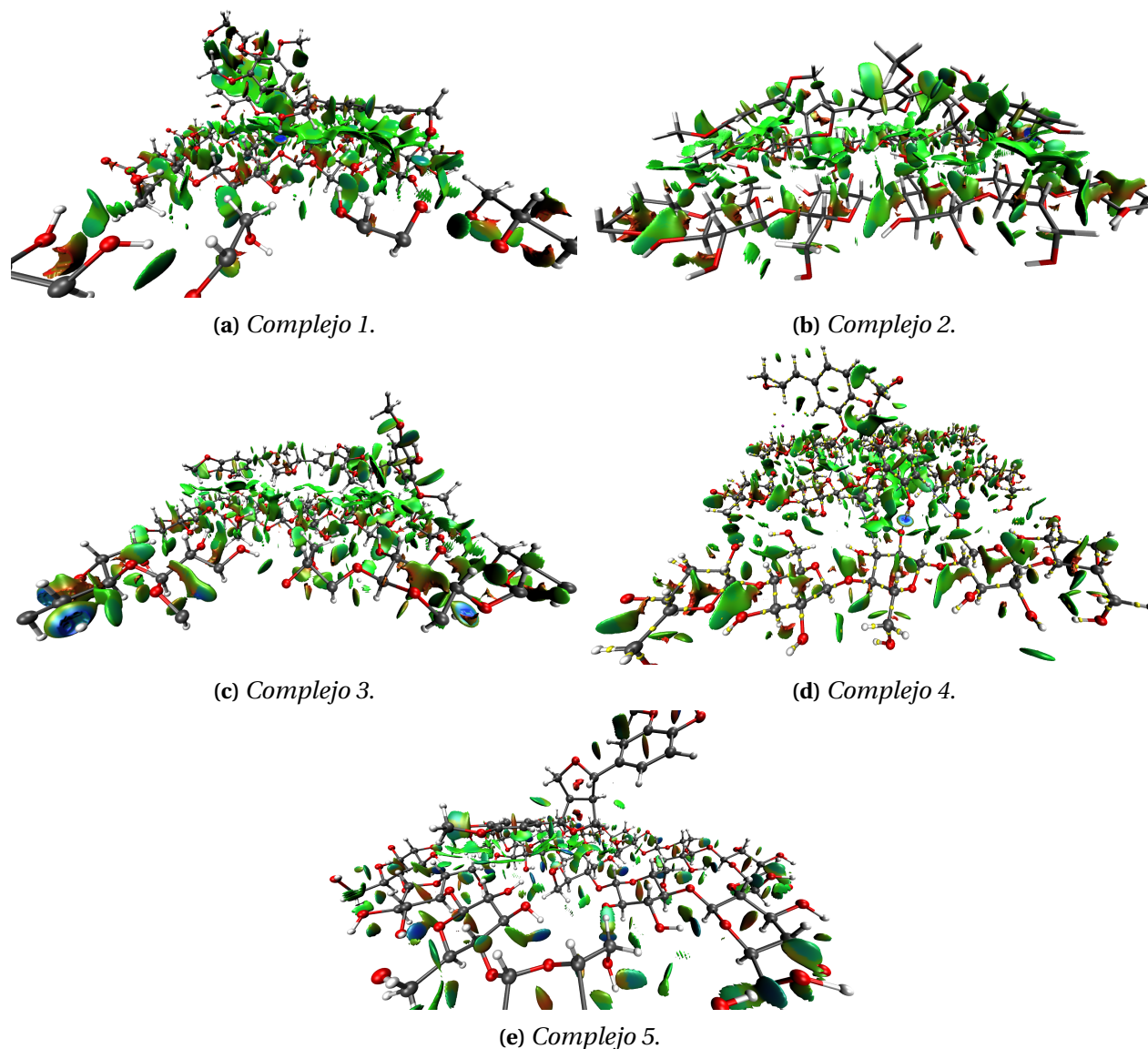
**Figura 5.2:** Trilignoles estudiados.**Figura 5.3:** Dilignoles estudiados.

críticos de la densidad usando GPUAM y complementando la búsqueda de puntos críticos, se realizó el análisis de interacciones no covalentes.

Las interacciones no covalentes entre los complejos oligolignol - celulosa fueron analizadas usando el INC, se observó que en todos los casos hay grandes zonas de interacción intermolecular (baja densidad electrónica y bajo gradiente reducido), y en términos generales las interacciones que prevalecen son en su mayoría van der Waals y en menor medida enlaces de hidrógeno fuertes, ver figura 5.4.

Como puede observarse para cada complejo las interacciones covalentes no aparecen entre cada par oligolignol y celulosa, por lo tanto, la adhesión se promueve mediante interacciones débiles. Por otro lado, las interacciones repulsivas, en color rojo, están asociadas a las interacciones intramoleculares (tensión anular) involucradas dentro del modelo de celulosa.

Además del análisis de INC, el enfoque de AEM fue usado para determinar de forma puntal los contactos entre los oligolignoles y la celulosa, a través las trayectorias de enlace. Al analizar las trayectorias de enlace, los enlaces de hidrógeno que se forman son entre las especies:  $O-H \cdots O$ ,  $C-H \cdots O$  y  $C_{\pi}-H \cdots O$ . Un análisis detallado de estos se puede ver en las Tablas 5.2-5.6. Las interacciones se clasifican en: 1) enlace de hidrógeno entre el grupo hidroxilo del oligolignol y el oxígeno la celulosa ( $oli - OH \cdots O-Cel$ ), 2) enlace de hidrógeno entre un oxígeno del oligolignol y el hidroxilo de la celulosa ( $oli-O \cdots HO-Cel$ ), 3) enlace de hidrógeno entre el grupo metilo



**Figura 5.4:** Análisis de interacciones no covalentes para los complejos oligolignol- celulosa, las zonas verdes corresponden a interacciones de van der Waals, los pequeños discos azulados corresponden a interacciones fuerte tipo puente de hidrógeno, mientras que las zonas rojas indican interacciones no enlazantes. Las esferas en color gris sugieren carbono, rojo para oxígeno y blanco para hidrógeno.

del oligolignol a un oxígeno a un oxígeno de la celulosa (oli-CH $\cdots$  O-Cel), 4) entre oxígeno de éter entre el oligolignol a un hidrógeno de la celulosa (oli-O $\cdots$  HC-Cel), 5) intracción  $\pi$  entre el oligolignol y un hidroxilo de la celulosa (oli-C $\pi$  - H  $\cdots$  O-Cel). Todas estas interacciones se informan en estas tablas donde  $d(A\cdots H)$ ,  $d(A\cdots D)$  y  $\alpha$  representan las distancias aceptor - átomo de hidrógeno, aceptor - donante y el ángulo A $\cdots$ H-D, respectivamente. Finalmente,  $\rho(\text{bcp})$  representa la densidad de electrones evaluada en un punto crítico de enlace.

Un resumen de todos los enlaces de hidrógeno encontrados en los cuatro complejos se

**Tabla 5.2:** Caracterización de los enlaces de hidrógeno para el complejo 1.

Oligolignol	Celulosa	d(A···H)/Å	d(A···D)/Å	$\alpha/^\circ$	$\rho(\text{bcp})/\text{u.a.}$
OH	O	1.76	2.74	171.0	0.040
OH	O	1.78	2.71	155.7	0.038
OH	O	2.10	3.06	175.3	0.019
O	HC	2.21	3.19	150.1	0.018
O	HO	2.18	2.96	136.8	0.016
CH	O	2.30	3.15	133.7	0.015
CH	O	2.30	3.25	144.6	0.015
O	HC	2.63	3.69	164.6	0.008
O	HC	2.71	3.40	121.4	0.007
CH	O	2.73	3.72	152.0	0.006
O	HC	2.75	3.42	119.3	0.006
OH	O	3.06	3.93	149.5	0.002
CH	O	3.35	3.93	114.4	0.002

**Tabla 5.3:** Caracterización de los enlaces de hidrógeno para el complejo 2.

Oligolignol	Celulosa	d(A···H)/Å	d(A···D)/Å	$\alpha/^\circ$	$\rho(\text{bcp})/\text{u.a.}$
OH	O	1.81	2.64	141.6	0.036
O	HO	1.99	2.77	136.4	0.025
CH	O	2.29	3.23	142.7	0.014
O	HO	2.25	2.92	125.9	0.013
O	HC	2.45	3.35	139.1	0.011
O	HC	2.45	3.47	155.7	0.010
O	HC	2.49	3.43	144.6	0.009
O	HC	2.59	3.55	147.0	0.008
O	HC	2.58	3.41	144.1	0.006
CH	O	2.83	3.38	110.6	0.005
O	HC	2.89	3.73	134.2	0.004
O	HC	2.91	3.78	137.9	0.004
CH	O	2.99	3.85	135.4	0.003
CH	O	3.00	3.86	136.0	0.003
CH	O	3.05	3.63	113.3	0.003
CH	O	3.13	3.90	127.9	0.003
C <sub><math>\pi</math></sub> H	O	3.16	4.02	137.4	0.002
O	HC	3.36	4.16	131.7	0.001
O	HC	3.44	4.03	115.8	0.001

**Tabla 5.4:** Caracterización de los enlaces de hidrógeno para el complejo 3.

Oligolignol	Celulosa	d(A···H)/Å	d(A···D)/Å	$\alpha/^\circ$	$\rho(\text{bcp}) / \text{u.a.}$
O	HO	1.87	2.8	162.4	0.031
CH	O	2.38	3.32	143.2	0.012
CH	O	2.42	3.41	150.3	0.011
CH	O	2.4	3.45	158.9	0.011
CH	O	2.43	3.41	118.4	0.011
CH	O	2.43	3.41	148.9	0.011
O	HC	2.48	3.49	153.5	0.010
CH	O	2.51	3.34	132.2	0.010
CH	O	2.54	3.03	106.0	0.009
CH	O	2.57	3.38	130.4	0.008
CH	O	2.61	3.55	144.6	0.008
CH	O	2.66	3.59	143.0	0.007
O	HC	2.69	3.54	134.4	0.007
O	HC	2.71	3.74	156.8	0.006
CH	O	2.74	3.33	113.7	0.006
O	HC	2.82	3.33	108.6	0.006
C $\pi$ H	O	2.89	3.43	110.4	0.005
O	HC	2.93	3.80	137.1	0.004
CH	O	3.16	4.12	147.0	0.002
O	HC	3.28	4.10	132.8	0.002
CH	O	3.36	4.38	155.8	0.001

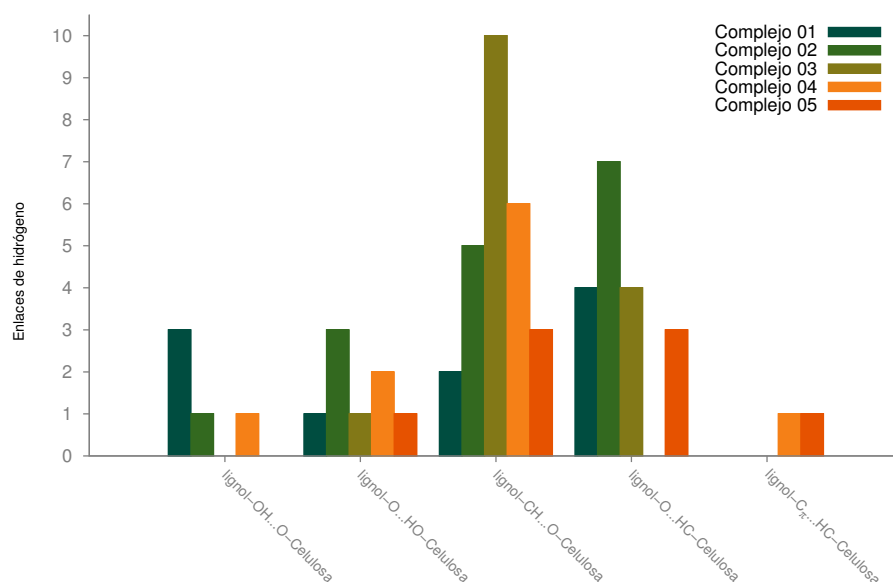
**Tabla 5.5:** Caracterización de los enlaces de hidrógeno para el complejo 4.

Oligolignol	Celulosa	d(A···H)/Å	d(A···D)/Å	$\alpha/^\circ$	$\rho(\text{bcp}) / \text{u.a.}$
OH	O	1.75	2.73	173.1	0.039
C $\pi$ H	O	2.35	2.96	114.1	0.014
O	HC	2.31	3.38	166.9	0.013
CH	O	2.39	3.47	171.1	0.012
C $\pi$ H	O	2.61	3.09	106.0	0.009
CH	O	2.56	3.21	117.5	0.008
C $\pi$ H	O	2.71	3.71	152.6	0.006
O	HC	2.80	3.71	142.0	0.005
O	HO	2.72	3.68	167.6	0.005
O	HC	2.91	3.75	133.9	0.003
CH	O	3.06	3.91	134.6	0.003

**Tabla 5.6:** Caracterización de los enlaces de hidrógeno para el complejo 5.

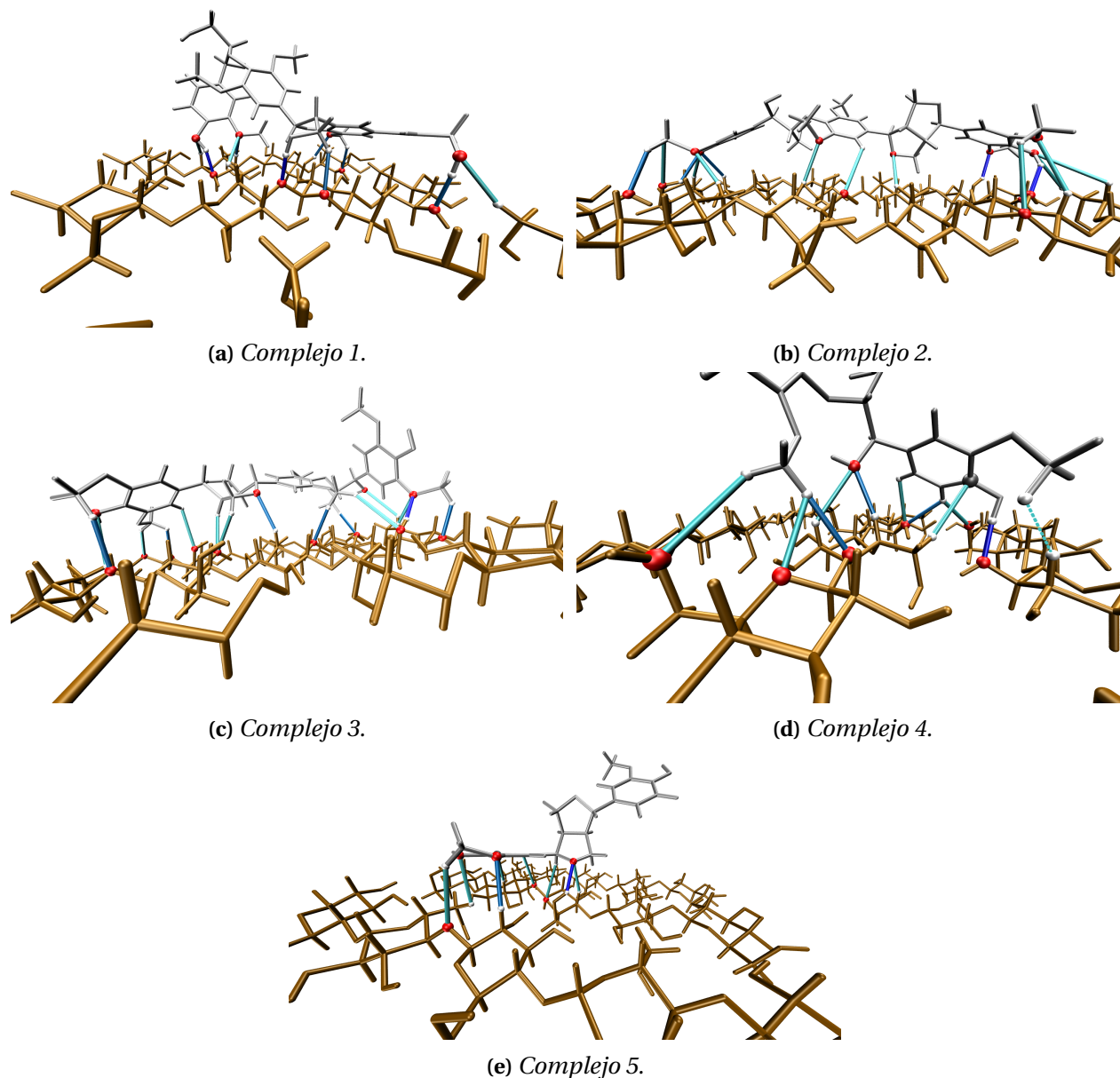
Oligolignol	Celulosa	$d(A \cdots H)/\text{Å}$	$d(A \cdots D)/\text{Å}$	$\alpha/^\circ$	$\rho(\text{bcp}) / \text{u.a.}$
O	HO	2.00	2.79	137.7	0.023
O	HC	2.38	3.36	149.4	0.012
O	HC	2.68	3.69	155.2	0.007
O	HC	2.77	3.16	100.8	0.007
CH	O	2.81	3.64	131.9	0.005
CH	O	2.77	3.69	141.6	0.005
$C_\pi H$	O	3.12	3.90	129.2	0.003

presenta en la figura 5.5, donde claramente el contacto  $CH \cdots O$  predomina sobre  $OH \cdots O$  y  $C_\pi H \cdots O$ . Por lo tanto, una interacción débil como la presentada por el contacto  $CH \cdots O$  tiene una relevancia importante para la adhesión entre oligolignoles y celulosa.

**Figura 5.5:** Clasificación de los enlaces de hidrógeno en los complejos oligolignol-celulosa.

Estas interacciones pueden verse en la figura 5.6, siguiendo el siguiente código de colores: marrón para la celulosa, gris para el oligolignol, rojo para el oxígeno, blanco para el átomo de hidrógeno. Finalmente los enlaces de hidrogeno se han marcado en una paleta de colores azules, a mayor saturación mas fuerte el enlace de hidrógeno.

A partir de estos resultados, es evidente que los parámetros geométricos exhibidos por algunos de estos enlaces de hidrógeno están fuera del rango de lo que se llamaría enlaces de



**Figura 5.6:** *Análisis de átomos en moléculas para las interacciones intermoleculares.*

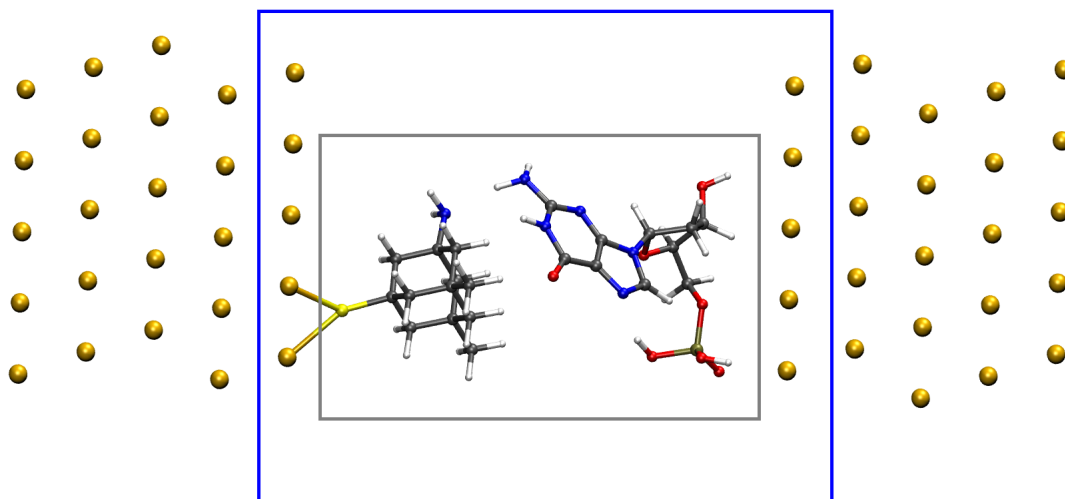
hidrógeno comunes. De estas tablas, claramente la densidad en el punto crítico de enlace es bastante pequeña para enlaces de hidrógeno con parámetros geométricos no favorecidos.

Del grupo de trilignoles es tal vez el complejo 3 y el complejo 2 lo que actuaría como mejores adhesivos debido a que generan mayores contactos vía enlace de hidrógeno, con 21 y 19 respectivamente. Complementando, el dilignol del complejo 4 genera 11 enlaces de hidrógeno, comparado contra el complejo 5 que es el que genera el menor número de contactos.

## 5.2 Interacciones intermoleculares en diamantoides

Los diamantoides son pequeñas moléculas con la estructura tridimensional del diamante, formando pequeñas cajas. El diamantoide más pequeño es el adamantano, de estructura condensada ( $C_{10}H_{16}$ ), es posible encontrarlo en pequeñas cantidades en el petróleo, algunos derivados de él han sido empleados con éxito en medicina, como medicamentos para el Parkinson o antivirales. Sin embargo, también pueden usarse para funcionalizar electrodos, por ejemplo sobre electrodos de Au(111), lo que promueve la fijación de ciertas bases nitrogenadas y le confiere características óptimas para desarrollar biosensores y secuenciadores de ADN [86].

A partir de la estructura (5.7), se realizaron cortes manteniendo la zona de interés intacta, la primera zona corresponde a la marcada con el rectángulo gris y corresponde al complejo Diamantoide (D)  $\cdots$  nucleótido de guanina (Gua) a este lo llamaremos sistema 1.



**Figura 5.7:** Estructura Au-Diamantoide  $\cdots$  Nucleótido de guanina - Au, vista lateral del sistema. Las esferas grises representan átomos de carbono, las azules nitrógeno, rojas oxígeno, blancas hidrógeno, amarilla azufre y naranja oro.

La siguiente zona corresponde al complejo del sistema 1, y una capa de oro a cada lado, a este lo nombraremos como sistema 2, y se encuentra marcada en la figura 5.7 con el recuadro azul.

Para analizar las interacciones entre el diamantoide y el nucleótido de guanina se realizaron los cálculos de estructura electrónica para los sistemas 1 y 2 usando el paquete computacional NWChem 6.6 [52], con estos cálculos se exportó el archivo wfx.

Los cálculos se realizaron en el marco de la teoría de funcionales de la densidad (TFD), para tener una correcta descripción de las interacciones no covalentes (van der Waals, enlaces de hidrógeno) usamos el funcional de Truhlar M06-2X [85]. Por otro lado, se utilizó el conjunto de base LANL2DZ para los átomos correspondientes a los 2 primeros periodos de la tabla periódica. Para los restantes se utilizó el pseudopotencial asociado a esta base (LANL2DZ) [87], esto con el fin de aligerar la carga computacional.

Con la información de la función de onda (archivo wfx) se realizaron dos tipos de análisis:

- Análisis de interacciones no covalentes.
- Búsqueda y caracterización de los puntos críticos de la densidad.

Estos cálculos se realizaron con GPUAM, sobre una arquitectura híbrida GPU-CPU. Se determinaron las zonas de interacciones no covalentes, los puntos críticos y las trayectorias de enlace, así como las propiedades topológicas asociadas a cada punto crítico de interés.

Para cada punto crítico de enlace, se reportan la densidad electrónica ( $\rho(\mathbf{r})$ ), el laplaciano de esta ( $\nabla^2\rho(\mathbf{r})$ ), la densidad de energía cinética Lagrangiana ( $G(\mathbf{r})$ ), la densidad de energía potencial ( $V(\mathbf{r})$ ) y la densidad de energía total ( $H(\mathbf{r})$ ), todos estos datos son reportados en el archivo log de GPUAM.

Existen trabajos que han relacionado la densidad de energía potencial con la energía de interacción para sistemas enlazados por puente de hidrógeno. Dado que las interacciones que aparecen en este sistema son de este tipo, resulta fácil estimar la energía de interacción con los datos de los puntos críticos [20]:

$$E_{HB} = \frac{1}{2}V(\mathbf{r}_{BCP}). \quad (5.1)$$

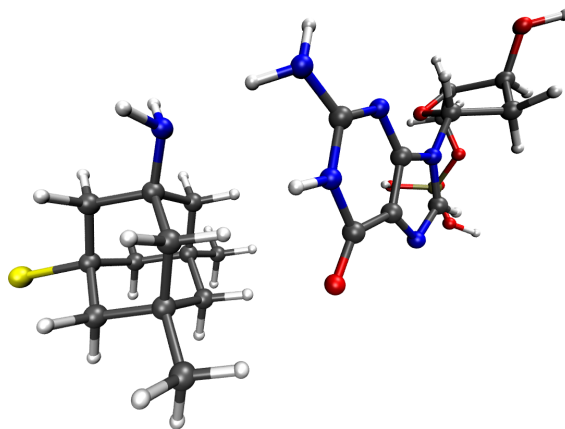
Lo anterior permite definir, de forma empírica una energía de interacción general con base en los valores de la densidad de energía potencial para cada punto crítico de enlace:

$$E_{A-B} = \frac{1}{2} \sum_i V(\mathbf{r}_i) \quad \forall \quad \mathbf{r}_i \in \{\mathbf{r}_{BCP}\} \text{ entre A y B.} \quad (5.2)$$



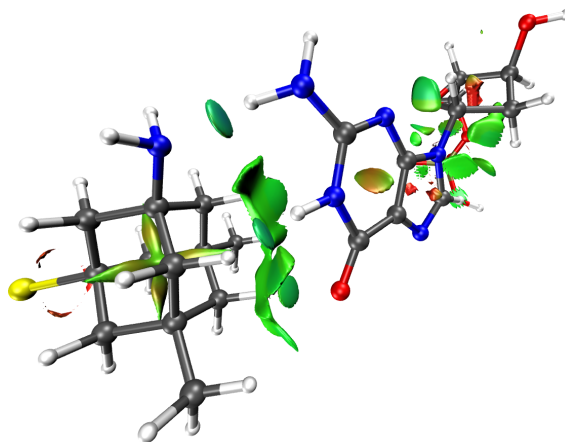
### 5.2.1 Sistema 1: Diamantoide - Guanina

Este sistema corresponde sólo a la parte interna, la conformada por el diamantoide y el nucleótido de guanina, figura 5.8.



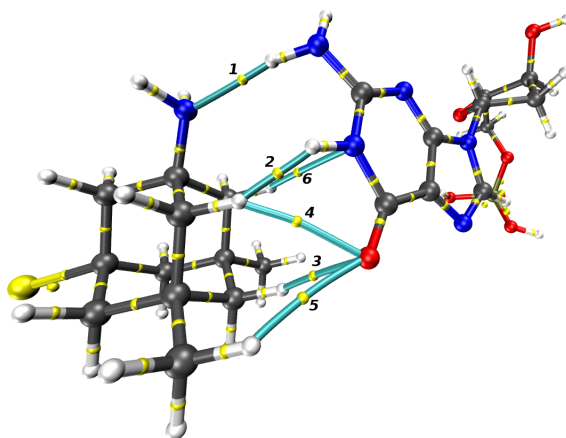
**Figura 5.8:** Estructura Diamantoide... Nucleótido de guanina. Las esferas grises representan átomos de carbono, las azules nitrógeno, rojas oxígeno, blancas hidrógeno y amarilla azufre.

Las interacciones no covalentes son las que prevalecen en la interfaz diamantoide - nucleótido, estas están marcadas por las grandes zonas verdes de la figura (5.9). Por otro lado, existen interacciones que aunque son no covalentes, son más direccionadas, las cuales se presentan como dos enlace de hidrógeno y un enlace dihidrógeno, zonas ligeramente azules. Mientras que las zonas con un ligero tono rojo son interacciones repulsivas que se esperan debido a la tensión anular de la guanina y del azúcar del nucleótido.



**Figura 5.9:** Índice de interacciones no covalentes para diamantoide - nucleótido.

El siguiente paso fue la búsqueda de puntos críticos; se obtuvieron 6 puntos críticos en la interfaz. Como el índice de interacciones no covalentes (INC) mostró, existen 3 interacciones más fuertes que las demás, estas corresponden a los puntos críticos 1, 2 y 3 de la figura 5.10.



**Figura 5.10:** Puntos críticos y trayectorias de enlace entre el diamantóide y el nucleótido, los puntos críticos se muestran en amarillo, mientras que la trayectoria de enlace en azul, se encuentran ordenados de acuerdo a la densidad en el punto crítico.

**Tabla 5.7:** Puntos críticos de enlace en el sistema 1, los datos están en unidades atómicas, la energía (E) en kcal/mol.

Punto	Tipo	$\rho(\mathbf{r})$	$\nabla^2\rho(\mathbf{r})$	$G(\mathbf{r})$	$V(\mathbf{r})$	$H(\mathbf{r})$	E
1	$D-H_2N\cdots H_2N-G$	0.022	0.066	0.016	-0.016	-0.000026	-5.156
2	$D-H\cdots HN-G$	0.017	0.064	0.015	-0.014	0.000977	-4.392
3	$D-H\cdots O=G$	0.012	0.050	0.010	-0.008	0.002183	-2.546
4	$D-H\cdots O=G$	0.007	0.023	0.005	-0.004	0.000983	-1.193
5	$D-H\cdots O=G$	0.005	0.023	0.004	-0.003	0.001452	-0.875
6	$D-H\cdots NH-G$	0.005	0.016	0.003	-0.002	0.000869	-0.682

Un resumen de los valores de densidad, laplaciano, densidad de energías cinética, potencial y total, así como un estimado de la energía de interacción se muestran en la tabla (5.7).

Dentro de la teoría de átomos en moléculas (AEM), es posible caracterizar las interacciones de enlace en interacciones de capa cerrada (iónicas y van der Waals) y en interacciones de capa compartida (covalentes).

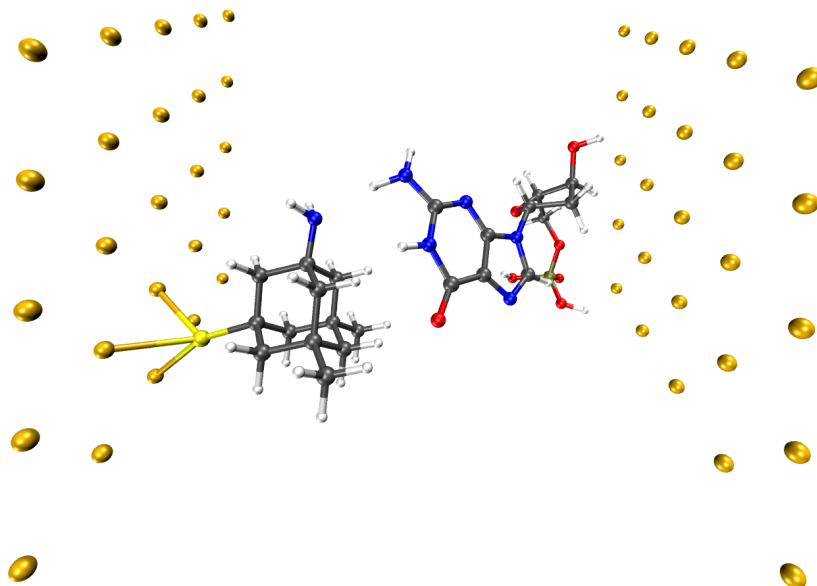
Las interacciones de capa cerrada tienen valores del laplaciano mayores a cero ( $\nabla^2\rho(\mathbf{r}) > 0$ ), debido a una reducción de la densidad electrónica en la zona internuclear. Por lo contrario, las de capa compartida tienen valores en el laplaciano menores a cero ( $\nabla^2\rho(\mathbf{r}) < 0$ ), debido a la acumulación electrónica en la zona internuclear.

Por otro lado, la densidad de energía total ( $H(\mathbf{r})$ ) también permite clasificar las interacciones, siendo mayor a cero para capa cerrada y menor a cero para capa compartida.

Podemos clasificar usando estos parámetros (densidad - laplaciano, y densidad de energía total  $H$ ) en interacciones de van der Waals y de puente de hidrógeno. Para poner en contexto, el enlace de hidrógeno en el dímero agua - agua, está caracterizado por  $\rho(\mathbf{r}_{BCP}) = 0.022$  u.a., tiene un laplaciano de  $\nabla^2\rho(\mathbf{r}_{BCP}) = 0.0232$  u.a. y energía de interacción de  $E = 4.46$  kcal/mol. De tal forma que la interacción uno y dos, son comparables a las encontradas en un enlace de hidrógeno como el presente en el dímero de agua.

### 5.2.2 Sistema 2: Sistema 1 con Oro

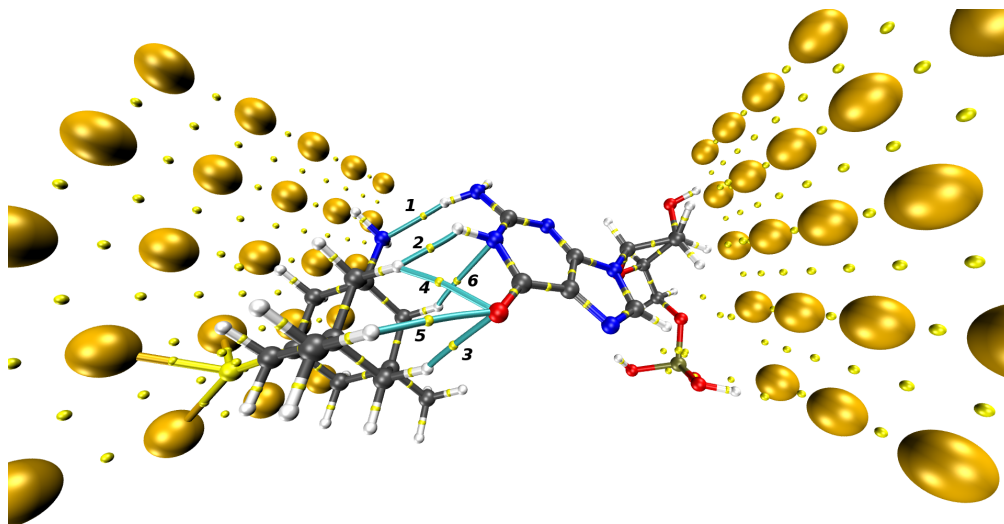
Este sistema corresponde a la parte interna más una capa de oro por cada lado, como se puede ver en la figura 5.11,



**Figura 5.11:** Estructura Au - Diamantoide ... Nucleótido ... Au.

Como en el sistema anterior, se obtuvo el archivo WFX, con toda la información necesaria para construir la densidad electrónica y campos derivados, se realizaron los análisis sobre las interacciones no covalentes. En este sistema, debido a la presencia de oro, se observó 3 zonas interfaciales. La primera entre el diamantoide y el nucleótido, tal y como se observó en el sistema 1, la segunda, entre el oro y el diamantoide. Finalmente, la tercera entre el nucleótido y el oro (figura 5.11).

Esta interfaz coincide con la del sistema 1, de igual forma existen 6 puntos críticos de enlace; en términos generales, podemos decir que la topología de la densidad electrónica se mantiene en ambos sistemas, véase la figura 5.10 y 5.12.



**Figura 5.12:** Puntos críticos y trayectorias de enlace entre el diamantoide y el nucleótido, los puntos críticos se muestran en amarillo, mientras que la trayectoria de enlace para la interfaz 1 en azul.

Un resumen de las propiedades topológicas de este sistema se exponen en la tabla (5.8).

**Tabla 5.8:** Puntos críticos de enlace en el sistema 2 y la interfaz diamantoide - nucleótido, los datos están en unidades atómicas, la energía (E) en kcal/mol.

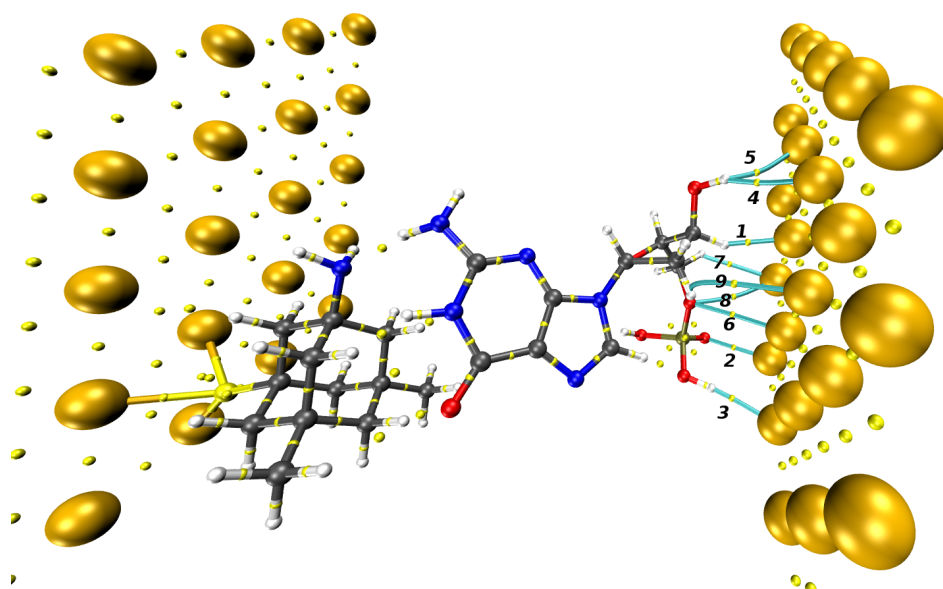
Punto	Tipo	$\rho(\mathbf{r})$	$\nabla^2\rho(\mathbf{r})$	$G(\mathbf{r})$	$V(\mathbf{r})$	$H(\mathbf{r})$	E
1	$D - H_2N \cdots H_2N - G$	0.022	0.066	0.016	-0.016	0.000033	-5.133
2	$D - H \cdots HN - G$	0.017	0.064	0.015	-0.014	0.000992	-4.384
3	$D - H \cdots O = G$	0.012	0.050	0.010	-0.008	0.002177	-2.545
4	$D - H \cdots O = G$	0.007	0.023	0.005	-0.004	0.000982	-1.192
5	$D - H \cdots O = G$	0.005	0.023	0.004	-0.003	0.001451	-0.875
6	$D - H \cdots NH - G$	0.005	0.016	0.003	-0.002	0.000872	-0.681

Para caracterizar el efecto que provoca el oro en la interfaz diamantoide - nucleótido, se compararon los puntos críticos, así como sus elementos topológicos entre el sistema 1 y del sistema 2. Se calculó la diferencia porcentual, siendo menores al 0.67% y de forma promediada, menores al 0.2% estos datos están resumidos en la tabla (5.9).

Al añadir la capa de oro a la derecha, se generan nueve contactos entre el azúcar (Dx), el grupo fosfato (Ph) y la capa de oro (Au). De acuerdo al valor que toma la densidad se pueden apreciar tres interacciones ligeramente más fuertes entre un hidrógeno del azúcar con el oro, un oxígeno del grupo fosfato, y un hidrógeno del grupo fosfato. Así pues, entre el nucleótido y el oro existe una interacción pero no afecta de manera importante a la zona de interfaz diamantoide - nucleótido.

**Tabla 5.9:** Comparación entre los sistemas 1 y 2, para los puntos críticos encontrados en la interfaz diamantoide-nucleótido, los datos de la columna dos y tres corresponden a densidades electrónicas en el punto crítico, estos datos están en unidades atómicas.

Punto	Densidad Sis. 1	Densidad Sis. 2	Diferencia porcentual
1	0.021894	0.021747	-0.673 %
2	0.012060	0.012059	-0.011 %
4	0.006907	0.006906	-0.015 %
5	0.005401	0.005408	0.120 %
6	0.004863	0.004853	-0.197 %
<b>Promedio</b>			-0.171 %



**Figura 5.13:** Puntos críticos y trayectorias de enlace entre el diamantoide y el nucleótido, los puntos críticos se muestran en amarillo, mientras que la trayectoria de enlace para la interfaz 3 en azul.

A partir del análisis de INC y AIM se puede enunciar que, el oro funciona como un soporte noble, que no altera las propiedades topológicas de la densidad entre diamantoide y la base nitrogenada, razón de ello es la diferencia encontrada en la densidad electrónica del sistema condensado y sin el oro, entre el diamantoide y el nucleótido. Sin embargo es la segunda capa de oro la que presenta el mayor número de contactos con los oxígenos del grupo fosfato de la base nitrogenada, lo que promovería cierta adherencia entre la base nitrogenada y el oro. Es necesario evaluar esta energía de unión, y analizar si el uso del oro es el adecuado para ser usado como electródo de soporte.

**Tabla 5.10:** Puntos críticos de enlace en el sistema 2, y la interfaz nucleótido - oro, los datos están en unidades atómicas, la energía (E) en kcal/mol.

Punto	Tipo	$\rho(\mathbf{r})$	$\nabla^2\rho(\mathbf{r})$	$G(\mathbf{r})$	$V(\mathbf{r})$	$H(\mathbf{r})$	$E$
1	$G-Dx-H\cdots Au$	0.034	0.101	0.031	-0.037	-0.005849	-11.585
2	$G-Ph-O\cdots Au$	0.033	0.144	0.038	-0.039	-0.001517	-12.277
3	$G-Ph-H\cdots Au$	0.023	0.059	0.017	-0.019	-0.001964	-5.883
4	$G-Dx-OH\cdots Au$	0.009	0.025	0.005	-0.004	0.000943	-1.362
5	$G-Dx-OH\cdots Au$	0.009	0.025	0.005	-0.004	0.001028	-1.352
6	$G-Ph-O\cdots Au$	0.008	0.027	0.006	-0.005	0.000982	-1.467
7	$G-Dx-H\cdots Au$	0.008	0.026	0.005	-0.004	0.001388	-1.181
8	$G-Ph-O\cdots Au$	0.007	0.019	0.004	-0.004	0.000595	-1.147
9	$G-dx-H\cdots Au$	0.005	0.016	0.003	-0.002	0.000994	-0.652

---

---

## CAPÍTULO 6

---

### CONCLUSIONES

En esta tesis se presentó, el uso de métodos basados en mallas, tanto para la evaluación de campos escalares y vectoriales de importancia en la química computacional y teórica, así como también para la búsqueda de puntos críticos. Esto ha llevado a la implementación de dos programas, GPUAM y CUBE3D, que explotan el paralelismo a nivel de datos, sobre mallas tridimensionales.

También es importante mencionar que la implementación hecha de GPUAM sobre una arquitectura híbrida CPU-GPU explota las herramientas actuales de computo. Se puede concluir que las tarjetas gráficas son importantes herramientas en el computo científico, debido al aumento de la velocidad en tiempos de cálculo, esto permite abordar sistemas químicos que hace un par de años era inviable.

Por otro lado, también se exploró el método de elemento finito para la resolución de la ecuación de Schrödinger monoeléctronica y su aplicación a sistemas confinados, observándose lo siguiente:

- La malla modificada de Chebyshev tiene un desempeño adecuado para ser usada en el método de elemento finito de sistemas que tengan un potencial coulombico.

- La implementación puede utilizarse para cualquier potencial unidimensional, dando excelentes resultados, como se observó al compararlos con los resultados exactos.
- El método de elemento finito para el ión molecular  $H_2^+$ , si bien no presentó el resultado exacto, sí presentó de forma general la tendencia que se esperaba, es necesario en este caso usar funciones de interpolación de mayor grado.

En las aplicaciones llevadas a cabo, se pudo encontrar que:

- En los sistemas formados por los distintos lignoles y la celulosa, se presentan enlaces de hidrógeno no convencionales  $C - H \cdots O$ , siendo sustancialmente mayores en los lignoles que presentan propiedades más adhesivas.
- En el sistema del diamantoide, el oro funciona como un soporte noble, que no altera las propiedades topológicas entre el diamantoide y la base nitrogenada. Sin embargo, es la segunda capa de oro la que presenta el mayor número de contactos con los oxígenos del grupo fosfato de la base nitrogenada.

Finalmente es importante mencionar que la creación de programas propios provee la autonomía necesaria para implementar teorías, ideas o algoritmos, sin depender de otros grupos de investigación.



---

## PERSPECTIVAS

A partir del trabajo desarrollado y los códigos presentados en esta tesis, las perspectivas se orientan en dos direcciones: el método de elemento finito y la programación en tarjetas gráficas.

Para tener un código aplicable a sistemas multielectrónicos delimitados espacialmente, es necesario realizar una correcta implementación de las integrales bielectrónicas. De igual forma para darle robustez al código FINIT2D, el paso natural es incrementar los polinomios de interpolación.

Por otra parte implementar en GPUAM las derivadas numéricas y búsqueda de puntos críticos previamente programados en CUBE3D. Un tema bastante importante y que permitiría extender el análisis topológico es la rápida y eficiente evaluación de integrales sobre las cuencas atómicas.

Seguir con la programación y evaluación de campos escalares y vectoriales que ayuden al análisis de interacciones moleculares. En este sentido trabajar con la función de Fukui, programarla para obtenerla a partir de la información de dos cálculos de estructura electrónica y con la maquinaria de GPUAM, realizar la búsqueda y caracterización de puntos críticos.



---

---

# APÉNDICE A

---

## ARMADO DE MALLAS EN GPUS

El siguiente código está escrito en CUDA-C y permite ejemplificar el armado de la malla para su ejecución en la tarjeta gráfica.

En este caso se arman bloques de  $16 \times 16$  hilos, lo que da un total de 256 hilos por bloque, en este ejemplo cada hilo evalúa un punto de la malla 3D y evalúa las operaciones de la ecuación (A.1), que corresponde a un orbital  $3d_{z^2}$ .

$$f(\mathbf{r}) = (3z^2 - r^2)e^{-r} \quad \text{con} \quad r = \sqrt{x^2 + y^2 + z^2}, \quad (\text{A.1})$$

### Código A.1: Programa

```
1 // Llamo las bibliotecas a usar
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <time.h>
5 #include <float.h>
6 #include <sys/time.h>
7
8 // Definimos las funciones que usara C
9 int compareVal (int, double*, double*);
10
11 int printCube3D (int, int, int,
12                double, double, double,
13                double, double, double,
```

```

14         double*);
15
16 double funcion      (double, double, double);
17 double tiempo      (struct timeval*, struct timeval*);
18
19 // Definimos las funciones que usara CUDA
20
21 __device__ double funcionDev (double, double, double);
22 __global__ void kernel      (int, int, int, int,
23                               double, double, double,
24                               double, double, double,
25                               double*);
26
27
28 // Implementacion de la funcion main
29 int main(int argc, char* argv[]){
30
31     int i, j, k, idx;
32     int nx, ny, nz, npt;
33     double x, y, z;
34     double x0, y0, z0;
35     double xn, yn, zn;
36     double hx, hy, hz;
37     double *valorCpu;
38
39     size_t tamanio;
40
41     // Variables para tomar tiempo
42     struct timeval tini, tfin;
43
44     // Leemos datos desde la entrada estandar
45     printf(" Numero de puntos en x,y,z\n ");
46     scanf("%d %d %d",&nx,&ny,&nz);
47     printf(" Coordenadas del punto inicial r0\n x0 y0 z0 : ");
48     scanf(" %lf %lf %lf",&x0,&y0,&z0);
49     printf(" Coordenadas del punto final  rN\n xN yN zN : ");
50     scanf(" %lf %lf %lf",&xn,&yn,&zn);
51
52     // Calculo el paso entre puntos
53     hx = (xn - x0)/(double) (nx-1);
54     hy = (yn - y0)/(double) (ny-1);
55     hz = (zn - z0)/(double) (nz-1);
56
57     // Reservo memoria para valor
58     npt = nx*ny*nz;
59     tamanio = npt*sizeof(double);
60
61     valorCpu = (double*) malloc (tamanio);
62     if ( valorCpu == NULL ) {
63         printf(" Error al reservar memoria para [valorCpu]\n");
64         exit(EXIT_FAILURE);
65     }
66
67     // Entro a tres for anidados
68     // Donde movere en este orden z -> y -> x
69
70     gettimeofday(&tini, NULL);
71
72     /******
73     /*          PARTE EN C          */
74     /******
75
76     idx = 0;
77     for( i=0 ; i<nx ; i++){
78         x = x0 + i*hx;
79
80         for( j=0 ; j<ny ; j++){
81             y = y0 + j*hy;
82

```

```

83     for( k=0 ; k<nz ; k++ ){
84         z = z0 + k*hz;
85
86         valorCpu[idx] = funcion(x,y,z);
87         idx++;
88     }
89 }
90 }
91 gettimeofday(&tfin ,NULL);
92
93 printf(" [INFO CPU ] Para calcular %10d puntos\n",npt);
94 printf(" [INFO CPU ] Usamos % 10.61f ms\n",tiempo(&tini,&tfin));
95
96 /*****
97 /*          PARTE EN CUDA          */
98 /*****
99
100 int HILOS1D = 16;
101 int HILOS2D = HILOS1D*HILOS1D;
102
103 int posibles=npt;
104 int bloquesX,bloquesY;
105 int bloques2D;
106
107 double *valorGpu;
108 double *dev_valorGpu;
109
110 // Variables para generar la malla
111 bloques2D = posibles / (HILOS2D);
112 if( posibles%HILOS2D != 0)
113     bloques2D++;
114
115 bloquesX = (int) sqrt(bloques2D);
116 bloquesY = (int) bloques2D/bloquesX;
117
118 if( bloquesX*bloquesY < bloques2D)
119     bloquesX++;
120 if( bloquesX*bloquesY < bloques2D)
121     bloquesY++;
122
123 printf(" [INFO CUDA] Blocks en X: %10d\n",bloquesX);
124 printf(" [INFO CUDA] Blocks en Y: %10d\n",bloquesY);
125 printf(" [INFO CUDA] Hilos en X: %10d\n",HILOS1D);
126 printf(" [INFO CUDA] Hilos en Y: %10d\n",HILOS1D);
127 printf(" [INFO CUDA] Hilos totales:%10d\n",bloquesX*bloquesY*HILOS2D);
128 printf(" [INFO CUDA] Hilos ociosos:%10d\n",bloquesX*bloquesY*HILOS2D-posibles);
129
130 // Reservo memoria para valorGpu
131 valorGpu = (double*) malloc (tamano);
132 if ( valorGpu == NULL ) {
133     printf(" Error al reservar memoria para [valorGpu]\n");
134     exit(EXIT_FAILURE);
135 }
136
137 gettimeofday(&tini ,NULL);
138 // Reservo memoria para dev_valor en la GPU
139 cudaMalloc((void**)&dev_valorGpu , tamano);
140
141 // Genero mi malla
142 dim3 blocks(bloquesX,bloquesY,1);
143 dim3 threads(HILOS1D,HILOS1D,1);
144
145 // llamo al kernel
146 kernel <<<blocks , threads>>> (nx,ny,nz,npt,x0,y0,z0,hx,hy,hz,dev_valorGpu);
147
148
149 // Copio del device al host
150 cudaMemcpy(valorGpu , dev_valorGpu , tamano , cudaMemcpyDeviceToHost);
151

```

```

152 gettimeofday(&tfin ,NULL);
153
154 // Libero memoria en el device
155 cudaFree(dev_valorGpu);
156
157
158 printf(" [INFO GPU ] Para calcular %10d puntos\n",npt);
159 printf(" [INFO GPU ] Usamos % 10.61f ms\n",tiempo(&tini,&tfin));
160
161
162 printCube3D(nx,ny,nz,x0,y0,z0,hx,hy,hz,valorGpu);
163
164 compareVal(npt,valorCpu,valorGpu);
165
166 // libero memoria en el host
167 free(valorCpu);
168 free(valorGpu);
169
170
171 exit(EXIT_SUCCESS);
172 }
173
174 // Implementamos las funciones a usar
175
176 // La funcion que vamos a evaluar en cada punto de la grid—C
177 double funcion(double x, double y, double z){
178 //return (x*x + y*y + z*z);
179 double r2 = x*x + y*y + z*z;
180 double r = sqrt(r2);
181 double val;
182
183 val = exp(-r);
184
185 val *= ( 3.*z*z - r2);
186
187 return val;
188 }
189
190 // Una funcion que nos ayudara a imprimir en una forma
191 // facilmente visualizable
192 int printCube3D(int nx, int ny, int nz,
193               double x0, double y0, double z0,
194               double hx, double hy, double hz,
195               double *valor){
196
197 int i,j,npt;
198 FILE *out;
199
200 npt = nx*ny*nz;
201
202 out = fopen("salida.cube","w+");
203 if(out == NULL){
204 printf(" No se pudo abrir el archivo [salida.cube]\n");
205 exit(EXIT_FAILURE);
206 }
207
208 fprintf(out," Archivo generado para visualizar datos 3D\n");
209 fprintf(out," en el formato cube \n");
210 fprintf(out," %4d % 10.61f % 10.61f % 10.61f\n", 1,x0,y0,z0);
211 fprintf(out," %4d % 10.61f % 10.61f % 10.61f\n",nx,hx,0.,0.);
212 fprintf(out," %4d % 10.61f % 10.61f % 10.61f\n",ny,0.,hy,0.);
213 fprintf(out," %4d % 10.61f % 10.61f % 10.61f\n",nz,0.,0.,hz);
214 fprintf(out," %4d % 10.61f % 10.61f % 10.61f % 10.61f\n", 3,3.,0.,0.,0.);
215
216 for( i=0,j=0 ; i<npt ; i++ ){
217 j++;
218 fprintf(out," % 10.6E ",valor[i]);
219 if( j == 6 || i == npt-1){
220 fprintf(out,"\n");

```

```

221     j = 0;
222     }
223 }
224
225 fclose(out);
226
227 return 0;
228 }
229
230 // Esta funcion compara dos conjuntos de datos,
231 int compareVal(int npt, double *conjuntoA, double *conjuntoB){
232
233     int i;
234     double rmsd,maxerror, diff;
235
236     maxerror = -1.E10;
237
238     rmsd = 0.;
239     for( i=0 ; i<npt ; i++){
240
241         diff = (double) (conjuntoA[i] - conjuntoB[i]);
242         diff *= diff;
243
244         if( diff > maxerror)
245             maxerror = diff;
246
247         rmsd += diff;
248     }
249
250
251     rmsd /= (double) npt;
252
253     printf(" [INFO ERRO] El mayor error cuadratico entre los dos conjuntos es\n");
254     printf(" [INFO ERRO] El RMSD entre los dos conjuntos es de : %10.6G\n",rmsd);
255     printf(" [INFO ERRO] MaxError : % 20.10E\n",maxerror);
256
257     if( maxerror < FLT_EPSILON)
258         printf(" [INFO ERRO] \033[1;31m Los dos conjuntos son estadisticamente iguales\n\033[0m");
259
260
261     return 0;
262 }
263
264 // Funcion para medir tiempo en milisegundos
265 // OJO! solo funciona en linux
266 double tiempo(struct timeval *ti, struct timeval *tf){
267     double time;
268
269     time = (double) (tf-> tv_sec +(double) tf-> tv_usec/1000000.);
270     time -= (double) (ti-> tv_sec +(double) ti-> tv_usec/1000000.);
271
272     time *= 1000.;
273
274     return time;
275 }
276
277 // Funcion a evaluar en el grid—CUDA
278 __device__ double funcionDev(double x, double y, double z){
279     double r2 = x*x + y*y + z*z;
280     double r = sqrt(r2);
281     double val;
282
283     val = exp(-r);
284
285     val *= ( 3.*z*z - r2);
286
287     return val;
288 }
289

```

```
290 // Kernel que es llamado desde el host
291 // Cambia de la malla real (fisica ) a la malla virtual
292 // para generar cada x,y,z
293 __global__ void kernel (int nx,int ny, int nz, int npt,
294                        double x0, double y0, double z0,
295                        double hx, double hy, double hz,
296                        double *valor){
297
298 // indices-problema
299 unsigned int idx, idy, idz;
300 // indices-gpu
301 unsigned int indiceHilo = threadIdx.y*blockDim.x + threadIdx.x;
302 unsigned int indiceBloq = blockIdx.y*gridDim.x + blockIdx.x;
303
304 unsigned int indice = (blockDim.x*blockDim.y)*indiceBloq + indiceHilo;
305 double x,y,z;
306
307 if(indice < npt){
308 // Apartir de los indices-gpu genero los indices-problema
309 // Una malla 2D genera la malla 3D
310 idx = (unsigned int) (indice/nz)/ny;
311 idy = (unsigned int) (indice/nz)%ny;
312 idz = (unsigned int) (indice%nz);
313
314 x = x0 + idx*hx;
315 y = y0 + idy*hy;
316 z = z0 + idz*hz;
317
318 valor[indice] = funcionDev(x,y,z);
319
320 } else
321 valor[indice] = 0.;
322
323 __syncthreads();
324 }
```



---

---

## APÉNDICE B

---

### MATRICES ELEMENTALES

El siguiente es un programa hecho en MATHEMATICA [67] que genera las matrices del elemento maestro, para el método de elemento finito unidimensional (MEF 1D), usando polinomios interpoladores de Lagrange [26] de orden  $n$ .

Para la matriz que contiene el término de energía potencial  $V(x)$ , se busca tener una mayor flexibilidad en el método, por lo que en lugar de usar un potencial particular,  $V(x)$  se aproxima como un polinomio del mismo grado que la interpolación usada. Por lo que el potencial está descrito por la ecuación (B.1), siendo  $V_j$  el potencial en el nodo  $x_j$ .

$$V(x) = \sum_{j=0}^n V_j \left[ \prod_{\substack{i=0 \\ i \neq j}}^n \left( \frac{x - x_i}{x_j - x_i} \right) \right] \quad (\text{B.1})$$

Definimos variables

```

n = 1; (* Orden del polinomio*)
n += 1; (* Nodos totales en e*)
ξh = Table[0, {i, n}]; (* Tabla para los nodos*)
vh = Table[0, {i, n}]; (* Tabla para los valores discretos de V*)
matT = Table[0, {i, n}, {j, n}]; (* Tabla para las matriz  $\mathbb{T}^e$  *)
matS = Table[0, {i, n}, {j, n}]; (* Tabla para las matriz  $\mathbb{S}^e$  *)
matV = Table[0, {i, n}, {j, n}]; (* Tabla para las matriz  $\mathbb{V}^e$  *)
(* Consideramos puntos intermedios equiespaciados *)
h = 2 / (n - 1);
Print["Step: ", h];
(* Mientras v0,v1,...,vn son los potenciales *)
Clear[v0, v1, v2, v3, v4, v5, v6, v7, v8, v9, v10];

If[n == 2, vh[[1]] = v0; vh[[2]] = v1;]

If[n == 3, vh[[1]] = v0; vh[[2]] = v1; vh[[3]] = v2;]

If[n == 4, vh[[1]] = v0; vh[[2]] = v1; vh[[3]] = v2; vh[[4]] = v3;]

If[n == 5, vh[[1]] = v0; vh[[2]] = v1; vh[[3]] = v2; vh[[4]] = v3;
vh[[5]] = v4;]

If[n == 6, vh[[1]] = v0; vh[[2]] = v1; vh[[3]] = v2; vh[[4]] = v3;
vh[[5]] = v4; vh[[6]] = v5;]

If[n == 7, vh[[1]] = v0; vh[[2]] = v1; vh[[3]] = v2; vh[[4]] = v3;
vh[[5]] = v4; vh[[6]] = v5; vh[[7]] = v6;]

If[n == 8, vh[[1]] = v0; vh[[2]] = v1; vh[[3]] = v2; vh[[4]] = v3;
vh[[5]] = v4; vh[[6]] = v5; vh[[7]] = v6; vh[[8]] = v7;]

If[n == 9, vh[[1]] = v0; vh[[2]] = v1; vh[[3]] = v2; vh[[4]] = v3;
vh[[5]] = v4; vh[[6]] = v5; vh[[7]] = v6; vh[[8]] = v7;
vh[[9]] = v8;]

If[n == 10, vh[[1]] = v0; vh[[2]] = v1; vh[[3]] = v2; vh[[4]] = v3;
vh[[5]] = v4; vh[[6]] = v5; vh[[7]] = v6; vh[[8]] = v7;
vh[[9]] = v8; vh[[10]] = v9;]

index[i_, j_] := (i - 1) * n + (j - 1);
(* Para el elemento maestro los limites son  $\xi_1=-1$  y  $\xi_{n+1}=+1$  *)
Do[

```

APÉNDICE B. MATRICES ELEMENTALES

```

ξh[[i]] = -1 + h * (i - 1),
{i, 1, n}
Print["Nodos: ", MatrixForm[ξh]]
Print["Potencial: ", MatrixForm[vh]]
Lag[i_, ξ_] := Module[{j, valor},
  valor = 1;
  Do[If[i ≠ j, valor *= (
    
$$\frac{\xi - \xi h[[j]]}{\xi h[[i]] - \xi h[[j]]}$$

)], {j, n}];
  Return[valor]];
Do[
  φ[i_, ξ_] := FullSimplify[ExpandAll[Lag[i, ξ]]];
  dφ[i_, ξ_] := D[φ[i, η], η] /. η → ξ;
  , {i, 1, n}];
Vapx[ξ_] = Together[FullSimplify[Sum[Lag[i, ξ] * vh[[i]], {i, 1, n}]]]
Step: 2
Nodos:  $\begin{pmatrix} -1 \\ 1 \end{pmatrix}$ 
Potencial:  $\begin{pmatrix} v0 \\ v1 \end{pmatrix}$ 
 $\frac{1}{2} (v0 + v1 - v0 \xi + v1 \xi)$ 
If[n == 2, va[ξ_] :=  $\frac{1}{2} (v0 + v1 - v0 \xi + v1 \xi)$ ];
If[n == 3, va[ξ_] :=  $\frac{1}{2} (2 v1 - v0 \xi + v2 \xi + v0 \xi^2 - 2 v1 \xi^2 + v2 \xi^2)$ ];
If[n == 4, va[ξ_] :=  $\frac{1}{16} (-v0 + 9 v1 + 9 v2 - v3 + v0 \xi - 27 v1 \xi + 27 v2 \xi - v3 \xi +$ 
 $9 v0 \xi^2 - 9 v1 \xi^2 - 9 v2 \xi^2 + 9 v3 \xi^2 - 9 v0 \xi^3 + 27 v1 \xi^3 - 27 v2 \xi^3 + 9 v3 \xi^3)$ ];
If[n == 5, va[ξ_] :=  $\frac{1}{6} (6 v2 + v0 \xi - 8 v1 \xi + 8 v3 \xi - v4 \xi - v0 \xi^2 + 16 v1 \xi^2 - 30 v2 \xi^2 + 16 v3 \xi^2 -$ 
 $v4 \xi^2 - 4 v0 \xi^3 + 8 v1 \xi^3 - 8 v3 \xi^3 + 4 v4 \xi^3 + 4 v0 \xi^4 - 16 v1 \xi^4 + 24 v2 \xi^4 - 16 v3 \xi^4 + 4 v4 \xi^4)$ ];
If[n == 6, va[ξ_] :=  $\frac{1}{768} (9 v0 - 75 v1 + 450 v2 + 450 v3 - 75 v4 + 9 v5 - 9 v0 \xi + 125 v1 \xi -$ 
 $2250 v2 \xi + 2250 v3 \xi - 125 v4 \xi + 9 v5 \xi - 250 v0 \xi^2 + 1950 v1 \xi^2 - 1700 v2 \xi^2 -$ 
 $1700 v3 \xi^2 + 1950 v4 \xi^2 - 250 v5 \xi^2 + 250 v0 \xi^3 - 3250 v1 \xi^3 + 8500 v2 \xi^3 - 8500 v3 \xi^3 +$ 
 $3250 v4 \xi^3 - 250 v5 \xi^3 + 625 v0 \xi^4 - 1875 v1 \xi^4 + 1250 v2 \xi^4 + 1250 v3 \xi^4 - 1875 v4 \xi^4 +$ 
 $625 v5 \xi^4 - 625 v0 \xi^5 + 3125 v1 \xi^5 - 6250 v2 \xi^5 + 6250 v3 \xi^5 - 3125 v4 \xi^5 + 625 v5 \xi^5)$ ];
If[n == 7, va[ξ_] :=  $\frac{1}{80} (80 v3 - 4 v0 \xi + 36 v1 \xi - 180 v2 \xi + 180 v4 \xi - 36 v5 \xi + 4 v6 \xi + 4 v0 \xi^2 -$ 
 $54 v1 \xi^2 + 540 v2 \xi^2 - 980 v3 \xi^2 + 540 v4 \xi^2 - 54 v5 \xi^2 + 4 v6 \xi^2 + 45 v0 \xi^3 - 360 v1 \xi^3 +$ 
 $585 v2 \xi^3 - 585 v4 \xi^3 + 360 v5 \xi^3 - 45 v6 \xi^3 - 45 v0 \xi^4 + 540 v1 \xi^4 - 1755 v2 \xi^4 + 2520 v3 \xi^4 -$ 
 $1755 v4 \xi^4 + 540 v5 \xi^4 - 45 v6 \xi^4 - 81 v0 \xi^5 + 324 v1 \xi^5 - 405 v2 \xi^5 + 405 v4 \xi^5 - 324 v5 \xi^5 +$ 
 $81 v6 \xi^5 + 81 v0 \xi^6 - 486 v1 \xi^6 + 1215 v2 \xi^6 - 1620 v3 \xi^6 + 1215 v4 \xi^6 - 486 v5 \xi^6 + 81 v6 \xi^6)$ ];

```

Together[Integrate[va[ξ] \* va[ξ], {ξ, -1, 1}]]

$$\frac{2}{3} (v_0^2 + v_0 v_1 + v_1^2)$$

**Realizamos las integrales para los elementos de la matriz**

```
Do[
  Do[
    matS[[i, j]] = Integrate[φ[i, ξ] * φ[j, ξ], {ξ, -1, 1}];
    matT[[i, j]] = Integrate[dφ[i, ξ] * dφ[j, ξ], {ξ, -1, 1}];
    matV[[i, j]] = Integrate[φ[i, ξ] * va[ξ] * φ[j, ξ], {ξ, -1, 1}];
    matS[[j, i]] = matS[[i, j]];
    matT[[j, i]] = matT[[i, j]];
    matV[[j, i]] = matV[[i, j]];
    , {j, j = i, n}
  , {i, 1, n}]
```

**Imprimimos los elementos de matriz**

Print["Elementos para la matriz de traslape"]

```
Do[
  Do[
    Print["ms[" , index[i, j], "]=", SetPrecision[matS[[i, j]], 20], ";"];
    , {j, 1, n}
  , {i, 1, n}]
```

Print[" Elementos para la matriz de energia cinetica"]

```
Do[
  Do[
    Print["mt[" , index[i, j], "]=", SetPrecision[matT[[i, j]], 20], ";"];
    , {j, 1, n}
  , {i, 1, n}]
```

Print[" Elementos para la matriz de energia potencial "]

```
Do[
  Do[
    Print["mv[" , index[i, j], "]=", SetPrecision[matV[[i, j]], 20], ";"];
    , {j, 1, n}
  , {i, 1, n}]
```

## APÉNDICE B. MATRICES ELEMENTALES

---

Elementos para la matriz de traslape

```
ms[0]=0.6666666666666666667;
```

```
ms[1]=0.3333333333333333333;
```

```
ms[2]=0.3333333333333333333;
```

```
ms[3]=0.6666666666666666667;
```

Elementos para la matriz de energia cinetica

```
mt[0]=0.5000000000000000000;
```

```
mt[1]=-0.5000000000000000000;
```

```
mt[2]=-0.5000000000000000000;
```

```
mt[3]=0.5000000000000000000;
```

Elementos para la matriz de energia potencial

```
mv[0]=0.5000000000000000000 v0 + 0.1666666666666666667 v1;
```

```
mv[1]=0.1666666666666666667 v0 + 0.1666666666666666667 v1;
```

```
mv[2]=0.1666666666666666667 v0 + 0.1666666666666666667 v1;
```

```
mv[3]=0.1666666666666666667 v0 + 0.5000000000000000000 v1;
```

### Imprimimos las matrices elementales

```
Print["Matriz Se: ", MatrixForm[matS]]
```

```
Print["Matriz Te: ", MatrixForm[matT]]
```

```
Print["Matriz Ve: ", MatrixForm[matV]]
```

Matriz Se: 
$$\begin{pmatrix} 2 & 1 \\ 3 & 3 \\ 3 & 3 \end{pmatrix}$$

Matriz Te: 
$$\begin{pmatrix} 1 & -1 \\ 2 & 2 \\ -1 & 1 \\ 2 & 2 \end{pmatrix}$$

Matriz Ve: 
$$\begin{pmatrix} \frac{v0}{2} + \frac{v1}{6} & \frac{v0}{6} + \frac{v1}{6} \\ \frac{v0}{6} + \frac{v1}{6} & \frac{v0}{6} + \frac{v1}{2} \end{pmatrix}$$



---

---

## **APÉNDICE C**

---

### **PUBLICACIONES**

A lo largo de este trabajo se han generado las siguientes publicaciones.

Volume 35 | Issues 31–32 | 2014  
 Included in this print edition:  
 Issue 31 (December 5, 2014)  
 Issue 32 (December 15, 2014)

# Journal of COMPUTATIONAL CHEMISTRY

Organic • Inorganic • Physical  
 Biological • Materials

www.c-chem.org

**Editors:**  
 Charles L. Brooks III • Masahiro Ehara • George Frenking • Peter R. Schreiner

**GRID**

**GPU**

**CRITICAL POINTS**

**Parallel Computing**

On page 2272, Jorge Garza and co-workers report an efficient grid-based algorithm to find critical points on the electron density, which scales very well on CPUs and exhibits a high performance on GPUs. The convenience in using this new proposal is evidenced when non-nuclear attractors are found in a molecule and when the code is used on common GPUs, which are non-dedicated to high-performance applications. The cover was designed by Alfredo Garza.



# Grid-Based Algorithm to Search Critical Points, in the Electron Density, Accelerated by Graphics Processing Units

Raymundo Hernández-Esparza,<sup>[a]</sup> Sol-Milena Mejía-Chica,<sup>[a]†</sup> Andy D. Zapata-Escobar,<sup>[a]</sup> Alfredo Guevara-García,<sup>[a]</sup> Apolinar Martínez-Melchor,<sup>[a]</sup> Julio-M. Hernández-Pérez,<sup>[b]</sup> Rubicelia Vargas,<sup>[a]</sup> and Jorge Garza<sup>\*[a]</sup>

Using a grid-based method to search the critical points in electron density, we show how to accelerate such a method with graphics processing units (GPUs). When the GPU implementation is contrasted with that used on central processing units (CPUs), we found a large difference between the time elapsed by both implementations: the smallest time is observed when GPUs are used. We tested two GPUs, one related with video games and other used for high-performance computing (HPC). By the side of the CPUs, two processors were tested, one used in common personal computers and other used for HPC, both of last generation.

Although our parallel algorithm scales quite well on CPUs, the same implementation on GPUs runs around 10× faster than 16 CPUs, with any of the tested GPUs and CPUs. We have found what one GPU dedicated for video games can be used without any problem for our application, delivering a remarkable performance, in fact; this GPU competes against one HPC GPU, in particular when single-precision is used. © 2014 Wiley Periodicals, Inc.

DOI: 10.1002/jcc.23752

## Introduction

The atoms in molecules (AIM) model proposed by Bader<sup>[1–3]</sup> is one of the most useful tools to understand the chemical bond. Within this method, the points where the gradient of the electron density is canceled have a physical meaning and depending on their curvatures, they are classified as bond-, ring-, cage-, nuclear-, and non-nuclear-, critical points. A full review of the AIM model is out of the scope of this report, but it can be found in Ref. [1].

If the number of basis set functions and the number of orbitals are large then the search of critical points requires and important computational effort because, in principle, the electron density and its first and second derivatives must be evaluated.<sup>[4–7]</sup> Grid-based methods (GBMs) alleviate part of the computational work because the gradient can be evaluated numerically and consequently, this approach offers an alternative to find critical points of any scalar field defined on a grid.<sup>[8–11]</sup> In 2013, Rodríguez proposed an algorithm to find critical points, of the electron density, based on GBM.<sup>[12]</sup> The main point of the Rodríguez' report steams on the distribution of the Newton–Raphson (NR) method, launched on each point of the grid over the servers where the application is executed. Thus, his algorithm was designed to be used in parallel by calling message passing interface libraries. Intrinsically, this method is quite expensive compared with other strategies to find critical points, for example, methods that restrict the search of critical points in regions where chemical knowledge dictates there should be a critical point, as in the case of a bond critical point situated between the coordinates of two atoms separated by a distance smaller than the sum of their covalent radii. The GBM is a double-edged sword; on one side, it has the disadvantage of explore the whole space occupied

by a molecule, even in regions where the electron density exhibits small values and where there are few or null possibilities to find critical points. Conversely, this method is suitable to find critical points for scalar fields that exhibit a large number of oscillations or to find uncommon critical points as non-nuclear attractors (NNAs) or unexpected cage critical points (CCPs).<sup>[13,14]</sup> Evidently, new algorithms are needed to accelerate GBM to be widely used.

Currently, the graphics processing units (GPU) represent an important alternative to be used for high-performance computing (HPC) applications.<sup>[15–17]</sup> In particular, we consider that GPUs are quite convenient to be used for GBM. Thus, the aim of this report is to show how the critical points of the electron density are found through GBM implemented on GPUs. In the next section, we explain how the Rodríguez' algorithm can be implemented on GPUs, in fact, we propose a modification on his algorithm to obtain a scalable application. In the results' section, we apply our code on several systems and report the time elapsed by GPUs, and benchmarking against those obtained by

[a] R. Hernández-Esparza, S.-M. Mejía-Chica, A. D. Zapata-Escobar, A. Guevara-García, A. Martínez-Melchor, R. Vargas, J. Garza  
Departamento de Química, Universidad Autónoma Metropolitana-Iztapalapa. San Rafael Atlixco 186. Col. Vicentina, Iztapalapa, C. P. 09340, México  
E-mail: jgo@xanum.uam.mx

[b] J.-M. Hernández-Pérez  
Facultad de Ciencias Químicas, Benemérita Universidad Autónoma de Puebla, Av. San Claudio y 18 Sur. Puebla, Puebla, C. P. 72530, México  
†Present address: Departamento de Química, Facultad de Ciencias, Pontificia Universidad Javeriana, Carrera 7 No. 40-62, Bogotá, D.C., Colombia.  
Contract grant sponsor: CONACYT, México; Contract grant numbers: 154784, 155070, and 155698; Contract grant sponsor: Universidad de Antioquia, Colombia; Contract grant numbers: 10169 and 1744 (A.-D. Zapata-Escobar)

© 2014 Wiley Periodicals, Inc.

central processing units (CPU). Finally, we give the conclusions at the end of this report, which are supported by the comparisons done between CPU–CPUs, GPUs–CPUs, and GPUs–GPUs.

## Methodology

### Critical points search on CPUs

For Kohn–Sham,<sup>[18]</sup> Hartree–Fock or correlated methods,<sup>[19]</sup> the electron density can be expressed as

$$\rho(\vec{r}) = \sum_{i=1}^{\text{occ}} \omega_i \psi_i^*(\vec{r}) \psi_i(\vec{r}), \quad (1)$$

where occ denotes the number of occupied orbitals ( $\psi_i$ ) and  $\omega_i$  represents the occupancy of each orbital. In this work, the orbitals are represented in terms of Gaussian functions ( $f_i$ ),

$$\psi_i(\vec{r}) = \sum_{\mu=1}^K c_{\mu}^i f_{\mu}(\vec{r}). \quad (2)$$

Here, the number of functions in the basis set is represented by  $K$ , and the coefficients  $\{c_{\mu}^i\}$  are obtained from a quantum chemistry method. In particular, we use the WFX format given by the G09 code<sup>[20]</sup> or a development version of the NWChem v6.3 code.<sup>[21]</sup>

The critical points,  $\vec{R}_c$ , of the electron density are those points where

$$\nabla \rho(\vec{R}_c) = \vec{0}. \quad (3)$$

The Rodríguez' algorithm to find critical points<sup>[12]</sup> of the electron density, can be summarized in the following steps:

1. The molecule to be studied is immersed in a 3D-grid.
2. The NR method<sup>[22]</sup> is applied over each point of the grid. This is an iterative method, where the gradient of the electron density is expanded through a Taylor series, around  $\vec{r}_i$ , to first order

$$\nabla \rho(\vec{r}_{i+1}) = \nabla \rho(\vec{r}_i) + \mathbf{H}|_{\vec{r}_i} (\vec{r}_{i+1} - \vec{r}_i). \quad (4)$$

This equation yields an iterative method to obtain  $\vec{r}_{i+1}$ , because at this point  $\nabla \rho(\vec{r}_{i+1}) = \vec{0}$ . Thus, we have

$$\vec{r}_{i+1} = \vec{r}_i - \alpha \mathbf{H}^{-1}|_{\vec{r}_i} \nabla \rho(\vec{r}_i). \quad (5)$$

In this expression, we have introduced the parameter  $\alpha$ , which modules the displacement of the new point.

This iterative method predicts the point where the gradient is zero, therefore, it is important to give a good starting point to get a rapid convergence. For each iteration of the NR method, the electron density, its gradient and its Hessian must be evaluated, which can be quite expensive if the number of functions in the basis set, the number of occupied orbitals and the number of iterations are large.

3. The critical points found in the previous step are analyzed to discriminate just the unique points.
4. Finally, the critical points are characterized according to the eigenvalues of their Hessian.

Evidently, the critical points search using a GBM is expensive because the number of points in the grid could be large. Rodríguez saves computational time in his algorithm by imposing a screening on the electron density and a trust sphere on points where a critical point is found. However, this algorithm has an important issue: there are no boundaries on the NR method when it is launched over each point of the grid. This is the main reason why the Rodríguez' algorithm is not efficient, as himself discussed in his report.<sup>[12]</sup>

### Critical points search on GPUs

The work distribution on GPUs is done following the Rodríguez' proposal, although we modified the second point of his algorithm. Instead of launching the NR method on each point of the 3D-grid, we start the NR method on the center of each cube involved in the 3D-grid. The main difference of our algorithm with the Rodríguez' proposal is that we stop the NR method whether we find, or not, a critical point within the cube where the NR method was launched. Finally, we distribute the work involved in each cube on each GPU thread; the modified step 2 of Rodríguez' algorithm is carried out within each thread of the GPU. This means that each thread computes, using analytical expressions, the electron density and its corresponding first and second derivatives for each step of the NR method. The algorithm was implemented from scratch using C and CUDA programming techniques.<sup>[16,17,23]</sup>

In CUDA, there are two important restrictions to distribute the work on a GPU: (1) The maximum number of blocks for each dimension is 65,535. (2) The maximum number of threads for each block is 1024. Thus, the work distribution of our algorithm on a GPU was over a 2D-grid of threads, therefore, our 3D-grid is expanded on a 2D-grid in the GPU. The partition of the 2D-grid was in blocks of 256 threads (16 threads along the  $x$  axis and 16 threads along the  $y$  axis). We found that this is the number of threads that exhibits the best performance, compared with 16, 64, or 1024 threads. Clearly, the number of cubes in the 3D-grid does not always coincide with the number of threads in the 2D-grid, for this reason, we use more threads than cubes in our application. In Figure 1, we present the CUDA code involved in this transformation. In this figure, points\_x, points\_y, and points\_z represent the number of points on each direction of the 3D-grid. The variables threadIdx.x, blockDim.x, and blockDim.y are reserved words in CUDA, and the same for axis  $y$ . The origin ( $x_0, y_0, z_0$ ) gives the initial point to expand the 3D-grid. For our case, we use a 3D-grid which gives a rectangular parallelepiped that envelops the molecule with boundaries of one atomic unit (a. u.) beyond of the Cartesian coordinates of the molecule. As it is indicated in Figure 1, the NR method starts from the center of each cube of the 3D-grid.

When the condition of Figure 1 is satisfied, the NR method is applied within each cube and finishes if a critical point is found or if the prediction done,  $\vec{r}_{i+1}$ , by this method gives a point slightly larger than the half of the cube (0.65 times the length of the cube side), on each direction. We go out of each cube because we want to find the points localized, if any, at

```

..
global_thread_x = threadIdx.x + blockIdx.x * blockDim.x;
global_thread_y = threadIdx.y + blockIdx.y * blockDim.y;
index_x = global_thread_x;
index_z = global_thread_y/points_y;
index_y = global_thread_y - index_z*points_y;
if (index_x < points_x && index_y < points_y && index_z < points_z) {
// Point on the 3D-grid
x = x0 + (float) index_x*h; y = y0 + (float) index_y*h; z = z0 + (float) index_z*h;
// Center of each cube
x = x + h/2.f; y = y + h/2.f; z = z + h/2.f;
..
Newton-Rhapson method
..
}

```

Figure 1. CUDA kernel to map cubes from a 3D-grid to threads on a GPU distributed on a 2D-grid.

the cube edges. Evidently, with this procedure we can get the same critical point between adjacent cubes. In the initial stage of this code, we sent 0 (no critical points) or 1 (critical point) to the host, and the CPU applied the NR method again on the cube. However, we decide to perform the whole search of the critical point on each cube and send its coordinates ( $x$ ,  $y$ , and  $z$ ) to the host from the device. The time relative to the data transfer is not relevant for the total time, in particular, for those systems with large 3D-grid. Finally, the characterization of each critical point is done by the CPU, in fact, we reapplied the NR method in the CPU with a smaller tolerance than that used by the GPU, and check if the found critical points are unique. In this sense, the GPU gives good starting points to the NR method within regions where there are surely critical points. The variables defined in the GPU and in the CPU are in single-precision. Although the CPU does the last stage of the critical points search, and it could be implemented in double-precision, we have found that the results do not differ in the digits that commonly are reported. Anyway, in this work, we show the results obtained by single- or double-precision.

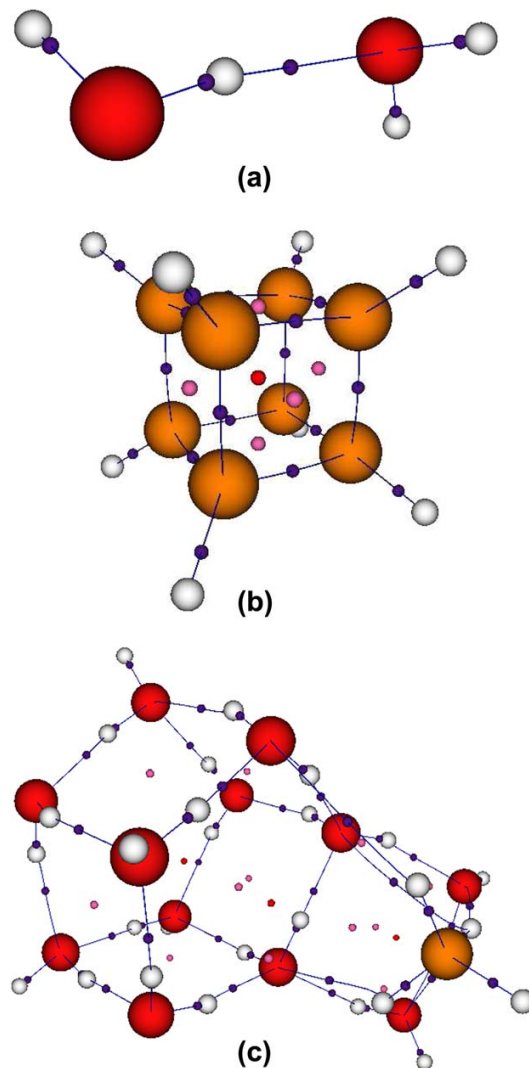
The implementation of our algorithm on CPUs is not the main target of this work. However, we contrast the time used by this way with regard to the time elapsed by GPUs. For that reason, we implemented the algorithm on CPUs using openmp programming techniques,<sup>[24]</sup> to use the code on several cores. The code is free of charge and can be requested to the corresponding author of this report.

## Results and Discussion

In this work, we present the performance of our algorithm on CPUs and GPUs for three systems: (a)  $\text{H}_2\text{O}-\text{H}_2\text{O}$ , (b)  $\text{C}_8\text{H}_8$ , (c) one conformer of the  $(\text{H}_2\text{O})_{12}\text{CH}_4$  cluster (Fig. 2). For the first

system, we use the M06-2x exchange-correlation functional with 10 occupied orbitals and 96 primitive Gaussian functions. For the second system, we use the B3LYP exchange-correlation functional with 28 occupied orbitals and 408 primitive Gaussian functions. The third system was studied with the MP2 method with 495 occupied orbitals and 720 primitive Gaussian functions. Furthermore, the algorithm was tested on two systems that exhibit uncommon critical points; the 1,12-difluorobenzo[c]phenanthrene and one conformer of the  $(\text{H}_2\text{O})_8^-$  cluster. The methods used in these systems were according to the previously published, B3LYP/6-311D(d,p)//B3LYP/6-31G(d) for the first one, and B3LYP/6-31++G(2p,2d) for the second one.<sup>[13,14]</sup> All geometries were optimized with the respective method, with the NWChem v6.3 code. The WFX files were obtained also by NWChem v6.3, except for the MP2 method, which was obtained by G09. We tested the MP2 method because it offers an additional computational effort since this method use natural orbitals, which in number, represents a bigger quantity than that needed for Hartree-Fock or Kohn-Sham methods. The 3D-grid used on each system has cubes with length of 1/6. Thus, 76,055 cubes for  $\text{H}_2\text{O}-\text{H}_2\text{O}$ , 288,923 cubes for  $\text{C}_8\text{H}_8$  and 1,133,237 cubes for  $(\text{H}_2\text{O})_{12}\text{CH}_4$  were used; evidently, this parameter can be easily changed by the user. In our tests, we found that 1/6 or 1/7 is enough to obtain all critical points of the electron density for any system, even for systems that exhibit NNAs. The size of each WFX file is really small, 28 KB for system (a), 256 KB for system (b), and 7138 KB for system (c). For one system with 138 nuclei, 556 electrons, and 2808 primitive Gaussian functions, we need 15,646 KB for the WFX file. Clearly, one GPU with 1 GB in RAM can deal with this kind of files without any problem.

In our attempt to see the performance of the GPUs to search critical points of the electron density, we tested two



**Figure 2.** Systems studied in this work: a) Dimer of water (H<sub>2</sub>O—H<sub>2</sub>O), b) cubane (C<sub>8</sub>H<sub>8</sub>), and c) methane hydrate ((H<sub>2</sub>O)<sub>12</sub>CH<sub>4</sub>). Critical points are represented by small spheres. Bond critical points are connected by bond paths, which start from a bond critical point and finish at a point where the electron density is a maximum.

Table 1. Electron density ( $\rho$ ) and its laplacian ( $\nabla^2\rho$ ) evaluated at some critical points.										
CP	Single-precision					Double-precision				
	$\rho$	$\nabla^2\rho$	$\lambda_1$	$\lambda_2$	$\lambda_3$	$\rho$	$\nabla^2\rho$	$\lambda_1$	$\lambda_2$	$\lambda_3$
H <sub>2</sub> O—H <sub>2</sub> O										
Bond	0.0260	0.1053	0.1758	-0.0349	-0.0357	0.0260	0.1053	0.1758	-0.0349	-0.0357
C <sub>8</sub> H <sub>8</sub>										
Bond	0.2785	-0.9511	0.5506	-0.7507	-0.7510	0.2785	-0.9511	0.5506	-0.7508	-0.7508
Bond	0.2253	-0.4146	0.4146	-0.4142	-0.4150	0.2253	-0.4146	0.4146	-0.4142	-0.4150
Ring	0.0808	0.3205	0.2148	0.2146	-0.1089	0.0808	0.3205	0.2148	0.2146	-0.1089
Cage	0.0343	0.2503	0.0836	0.0834	0.0833	0.0343	0.2503	0.0836	0.0834	0.0833
(H <sub>2</sub> O) <sub>12</sub> CH <sub>4</sub>										
Bond	0.0039	0.0139	0.0200	-0.0029	-0.0031	0.0039	0.0139	0.0200	-0.0029	-0.0031
Bond	0.0022	0.0077	0.0099	-0.0009	-0.0014	0.0022	0.0077	0.0099	-0.0009	-0.0014
Bond	0.0040	0.0157	0.0201	-0.0019	-0.0026	0.0040	0.0157	0.0201	-0.0019	-0.0026
Bond	0.0035	0.0137	0.0182	-0.0021	-0.0024	0.0035	0.0137	0.0182	-0.0021	-0.0024
Bond	0.0035	0.0142	0.0186	-0.0019	-0.0025	0.0035	0.0142	0.0186	-0.0019	-0.0025

$\lambda_i$  represents an eigenvalue of the Hessian of  $\rho$ , and CP the characterization of a critical point.

Table 2. Total time, in s, elapsed for the critical points search of the electron density.			
CPU	H <sub>2</sub> O—H <sub>2</sub> O	C <sub>8</sub> H <sub>8</sub>	(H <sub>2</sub> O) <sub>12</sub> CH <sub>4</sub>
<b>Proc 1</b>			
1	31 (1.00)	1938 (1.00)	254561 (1.00)
2	16 (0.97)	968 (1.00)	128840 (0.99)
4	09 (0.86)	514 (0.94)	67835 (0.94)
<b>Proc 2</b>			
1	50 (1.00)	3175 (1.00)	428877 (1.00)
2	25 (1.00)	1591 (1.00)	215960 (0.99)
4	13 (0.96)	793 (1.00)	108103 (0.99)
8	07 (0.89)	398 (1.00)	54211 (0.99)
16	04 (0.78)	200 (0.99)	27170 (0.99)

platforms. By the CPUs side, we tested the i7-4770@3.40 GHz (Proc 1) and the dual Xeon E5-2670 v2@2.50 GHz (Proc 2) processors, both from Intel™ company. By the GPU side, we tested the GeForce GTX 760 (Card 1) and a Tesla M2090 (Card 2) video cards, both from NVIDIA™ company. We are making this comparison because the combination i7-4770+GeForce GTX 760 is one personal computer (PC) which is mainly used for video games, and the dual Xeon E5-2670 v2+Tesla M2090 corresponds to a HPC devices.

The critical points found for three systems, considered in this work, are reported in Table 1. For H<sub>2</sub>O—H<sub>2</sub>O, there are five critical points, all of them are bond critical points (BCP), but just the one associated with a hydrogen bond is reported. For C<sub>8</sub>H<sub>8</sub>, there are 20 BCPs, 6 ring critical points (RCP), and 1 CCP, but just four are unique due to the system symmetry. For (H<sub>2</sub>O)<sub>12</sub>CH<sub>4</sub>, there are 51 BCPs, 14 RCPs, and 3 CCPs, just those corresponding to weak hydrogen bonds, CH...O interactions, are reported in Table 1.

Results obtained by our code, density and its Laplacian at the critical point coordinates, are in accord with hydrogen bonds of medium or weak strength since they are consistent with results obtained for similar interactions.<sup>[25,26]</sup> Furthermore, the values reported in Table 1 coincide with those obtained by the Multiwfn software.<sup>[27]</sup> From this table, clearly the results obtained by single-precision are the same than those obtained by double-precision, just in two eigenvalues, we found one difference in the fourth figure. Thus, we are confident with the results obtained by our code.

The first thing we are showing is the performance of our algorithm when CPUs are used. In Table 2, we are presenting the time elapsed for the search of critical points in the elec-

Table 3. Total time, in s, elapsed for the critical points search of the electron density.						
GPU	Single-precision			Double-precision		
	H <sub>2</sub> O—H <sub>2</sub> O	C <sub>8</sub> H <sub>8</sub>	(H <sub>2</sub> O) <sub>12</sub> CH <sub>4</sub>	H <sub>2</sub> O—H <sub>2</sub> O	C <sub>8</sub> H <sub>8</sub>	(H <sub>2</sub> O) <sub>12</sub> CH <sub>4</sub>
<b>Card 1</b>						
1	01	21	2083	01	43	4809
<b>Card 2</b>						
1	01	24	2348	01	37	3783
2		15	1381		25	2386
4		13	820		22	1498

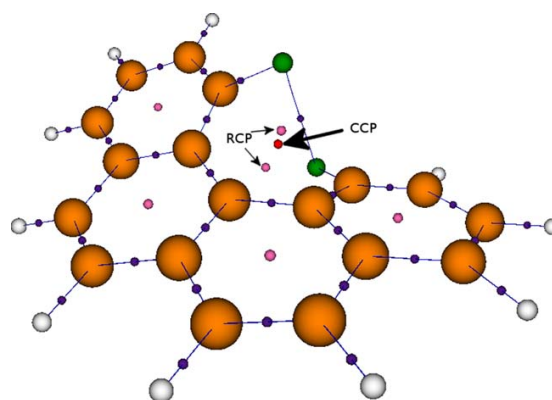


Figure 3. Optimized structure of the 1,12-di-fluorobenzo[c]phenanthrene molecule using the B3LYP/6-31G(d) method. Large arrow indicates the cage critical point (CCP) and small arrows indicate ring critical points (RCP). All critical points were obtained using the B3LYP/6-311++G(d,p)//B3LYP/6-31G(d) method, and they are represented by small spheres. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

tron density, for three systems tested; in parenthesis, the efficiency [time of one processor/(number of processors × elapsed time)] of our algorithm is also reported. We have decided to use three systems with different size since we can appreciate that even for small systems the performance of our algorithm is quite good, evidently our algorithm belongs to embarrassingly parallel problems, which is the idea of GBM. As we mentioned in the introduction, the performance of Rodríguez' algorithm is bad, clearly the restricted search within a cube ensures a good scaling. Additionally to the performance presented by our algorithm, we want to mention the performance of each processor. From these numbers, there is no doubt that a processor used for a common PC consumer compares quite well with that processor used for HPC.

In Table 3, we are reporting the time elapsed by GPUs for three systems considered in this work. By contrasting the time used by GPUs with respect to CPUs, clearly the GPUs show a good performance. For our code, 16 cores in a node built with two Proc 2 spend around 10 times the time used by 1 GPU. In the same table, it can be seen the scaling observed when several GPUs are used, evidently, the scaling observed for our application is not as good as that delivered by CPUs. The main reason of this behavior is the latency involved with GPUs. Evidently, the bigger the system, the better the scaling observed by GPUs. In the same table, we are contrasting the time used by single- and double-precision. It is well known that for double-precision applications, the GPUs show a lower performance than that observed when single-precision is used; this fact is evident in Table 3. However, clearly the reduction on the performance is bigger for the Card 1, which has much sense since the Tesla GPU family works better for double-precision applications. Even with this impact on the GPU performance, the time used by the GPU is much lower than that used by the CPU.



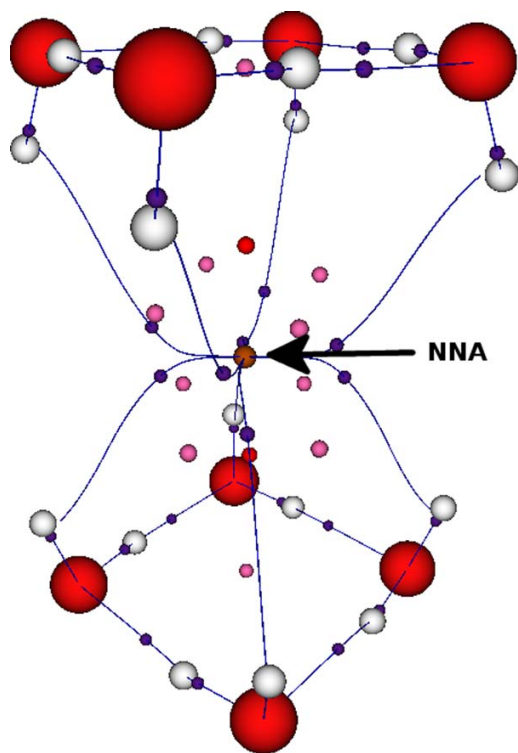


Figure 4. Optimized structure of one conformer of the  $(\text{H}_2\text{O})_8^-$  cluster using the B3LYP/6-31++G(2d,2p) method. Large arrow indicates the non-nuclear attractor. All critical points were obtained by using the B3LYP/6-31++G(2d,2p) method, and they are represented by small spheres. [Color figure can be viewed in the online issue, which is available at [wileyonlinelibrary.com](http://wileyonlinelibrary.com).]

We have showed the performance of our code, now we will discuss two systems that exhibit unusual critical points. In Figure 3, we are presenting the 1,12-difluorobenzo[c]phenanthrene with its corresponding critical points. Castillo et al.<sup>[13]</sup> reported one CCP for this system within a single ring. In this figure, the CCP is indicated by an arrow. Our code found all critical points, using

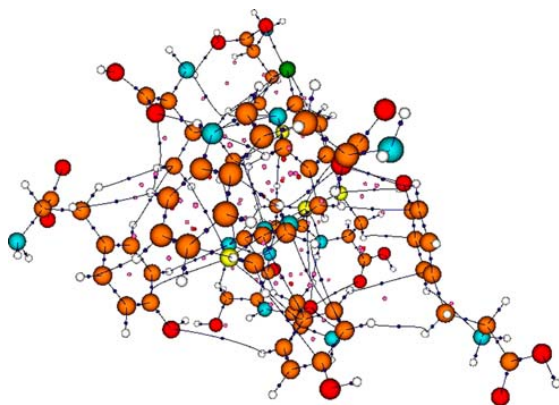


Figure 5. System with 455 occupied orbitals and 4619 primitive Gaussian functions depicted with its corresponding critical points and bond paths. Critical points are represented by small spheres.

one step of  $1/7 \text{ \AA}$ , in 285 s with the Tesla M2090 GPU. In particular, the CCP was found without any modification of the code or an additional option in the input file. For this point, we found  $\rho = 0.008157 \text{ a.u.}$  and  $\nabla^2\rho = 0.052189 \text{ a.u.}$  When the Multiwfn code is used with the same wfx file, we found exactly the same results. Castillo et al. reported for this point  $\rho = 0.0083 \text{ a.u.}$  and  $\nabla^2\rho = 0.0515 \text{ a.u.}$  We attribute the small differences to the geometry and the wfx file generated by different codes, in our case, we used the NWChem code.

Finally, we applied our code on one conformer of the  $(\text{H}_2\text{O})_8^-$  cluster. This system is very interesting because exhibits a NNA.<sup>[14]</sup> Our code spent 123 s to find the 45 critical points involved in this system. The NNA (Fig. 4) has associated  $\rho = 0.001036 \text{ a.u.}$  and  $\nabla^2\rho = -0.000175 \text{ a.u.}$  When the Multiwfn code is used, we obtained same results. Clearly, the convenience of using GBM for finding NNAs is highlighted in this system.

In this work, we have reported just five cases where it is evident the good performance exhibited by GPUs, but we have applied our code on larger systems. In particular, we applied our algorithm, without any problem, on a system with 455 occupied orbitals and 4619 primitive Gaussian functions. This system is presented in Figure 5, which presents BCPs, RCPs, and CCPs, these CPs are depicted with small spheres. The figures generated for bond paths, CPs and nuclei were generated on a GPU with software made by us. Finally, we want to stress the time used by a video games' GPU with regard to that presented by a GPU dedicated to HPC. Although the Tesla M2090 GPU does not belong to the last generation of GPUs (GPU technology is moving rapidly), this is still a card that can be found in an internet store. From this discussion, clearly, we can use a video game platform (many of us have one at home) to perform scientific work.

## Conclusions

In this work, we have reported an implementation of a grid-based critical points search on GPUs. When the time elapsed by GPUs with regard to CPUs is contrasted, clearly the GPUs give the best performance. Doubtless algorithms implemented on GPUs will gain more confidence in the quantum chemistry community and as, we discussed it above, one PC dedicated to video games can be used for large demanding computational tasks. The next step in our laboratory is the implementation of numerical derivatives to analyze any scalar field in a faster way with respect to the time consumed in the evaluation of analytical derivatives, which is very useful for codes where the wave function is represented in reciprocal space or for scalar fields with large oscillations. Thus, we are implementing different scalar and vector fields in our code, such as electrostatic potential,<sup>[28]</sup> electron localization function,<sup>[29]</sup> laplacian,<sup>[1]</sup> kinetic energy density,<sup>[30]</sup> and noncovalent interactions analysis.<sup>[31]</sup> Evidently, the algorithm presented, in this report, will be quite useful to search the critical points in these fields that by their nature present a large number of oscillations. Besides, we are doing efforts to implement population analysis on GPUs to offer one code with several applications

accelerated by GPUs, we hope in the near future report such applications.

### Acknowledgments

The authors thank the facilities provided by the Laboratorio de Supercómputo y Visualización en Paralelo at the Universidad Autónoma Metropolitana-Iztapalapa.

**Keywords:** atoms in molecules · electron density · parallel computing · graphics processing units

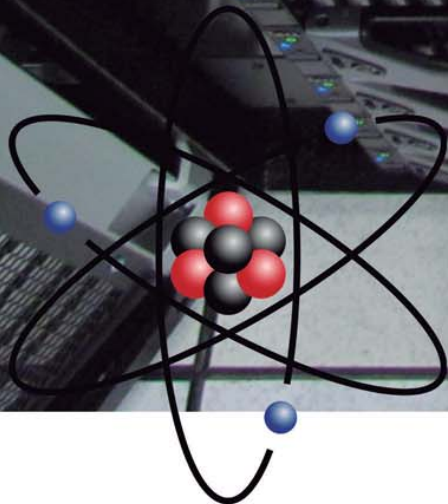
How to cite this article: R. Hernández-Esparza, S. M. Mejía-Chica, A. D. Zapata-Escobar, A. Guevara-García, A. Martínez-Melchor, J. M. Hernández-Pérez, R. Vargas, J. Garza. *J. Comput. Chem.* **2014**, *35*, 2272–2278. DOI: 10.1002/jcc.23752

- [1] R. F. W. Bader, *Phys. Rev. B* **1994**, *49*, 13348–13356.
- [2] R. F. W. Bader, *Atoms in Molecules: A Quantum Theory*, Oxford University Press: New York, **1990**.
- [3] C. F. Matta, R. J. Boyd, Eds. *The Quantum Theory of Atoms in Molecules: From Solid State to DNA and Drug Design*, Wiley-VCH: Weinheim, **2007**.
- [4] J. Cioslowski, A. Nanayakkara, *Chem. Phys. Lett.* **1994**, *219*, 151–154.
- [5] P. L. A. Popelier, *Chem. Phys. Lett.* **1994**, *228*, 160–164.
- [6] P. L. A. Popelier, *Comput. Phys. Commun.* **1996**, *93*, 212–240.
- [7] P. Balanarayan, S. R. Gadre, *J. Chem. Phys.* **2003**, *119*, 5037–5043.
- [8] M. Leboeuf, A. M. Köster, K. Jug, D. R. Salahub, *J. Chem. Phys.* **1999**, *111*, 4893–4905.
- [9] P. Rabiller, M. Souhassou, C. Katan, C. Gatti, C. Lecomte, *J. Phys. Chem. Solids* **2004**, *65*, 1951–1955.
- [10] G. Henkelman, A. Arnaldsson, H. Jónsson, *Comput. Mater. Sci.* **2006**, *36*, 354–360.
- [11] E. Sanville, S. D. Kenny, R. Smith, G. Henkelman, *J. Comput. Chem.* **2007**, *28*, 899–908.
- [12] J. I. Rodríguez, *J. Comput. Chem.* **2013**, *34*, 681–686.
- [13] N. Castillo, C. F. Matta, R. J. Boyd, *Chem. Phys. Lett.* **2005**, *409*, 265–269.
- [14] A. Taylor, C. F. Matta, R. J. Boyd, *J. Chem. Theory Comput.* **2007**, *3*, 1054–1063.
- [15] W.-M. W. Hwu, *GPU Computing Gems*, Emerald Edition, Morgan Kaufmann: Boston, **2011**.
- [16] J. Sanders, E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, NVIDIA Corporation: Boston, **2010**.
- [17] R. Farber, *CUDA Application Design and Development*, Elsevier: Boston, **2011**.
- [18] R. G. Parr, W. Yang, *Density-Functional Theory of Atoms and Molecules*, Oxford University Press: Oxford, **1989**.
- [19] A. Szabo, N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, Dover Publications: New York, **1996**.
- [20] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, N. J. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, Ö. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, D. J. Fox, Gaussian 09, Revision D.01, Gaussian, Inc., Wallingford CT, **2009**.
- [21] M. Valiev, E. J. Bylaska, N. Govind, K. Kowalski, T. P. Straatsma, H. J. J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T. L. Windus, W.A. de Jong, *Comput. Phys. Commun.* **2010**, *181*, 1477–1489.
- [22] R. L. Burden, J. D. Faires, *Numerical analysis*, 9th ed., Cengage Learning: Boston, **2010**.
- [23] B. W. Kernighan, D. M. Ritchie, *C Programming Language*, 2nd ed., Prentice Hall: New Jersey, **1988**.
- [24] B. Chapman, G. Jost, R. Van der Pas, *Using OpenMP: Portable Shared Memory Parallel Programming*, MIT Press: Massachusetts, **2007**.
- [25] A.-M. Navarrete-López, J. Garza, R. Vargas, *J. Phys. Chem. A* **2007**, *111*, 11147–11152.
- [26] U. Koch, P. L. A. Popelier, *J. Phys. Chem.* **1995**, *99*, 9747–9754.
- [27] T. Lu, F. Chen, *J. Comput. Chem.* **2012**, *33*, 580–592.
- [28] J. N. Murray, P. Politzer, *Comput. Mol. Sci.* **2011**, *1*, 153–163.
- [29] A. D. Becke, K. E. Edgecombe, *J. Chem. Phys.* **1990**, *92*, 5397–5403.
- [30] A.-M. Navarrete-López, J. Garza, R. Vargas, *J. Chem. Phys.* **2008**, *128*, 104110(8).
- [31] E. R. Johnson, S. Keinan, P. Mori-Sánchez, J. Contreras-García, A. J. Cohen, W. Yang, *J. Am. Chem. Soc.* **2010**, *132*, 6498–6506.

Received: 3 July 2014  
Revised: 14 August 2014  
Accepted: 14 September 2014  
Published online in Wiley Online Library

Química cuántica, supercómputo  
y tarjetas gráficas (GPUs).

Raymundo Hernández-Esparza, Andy D. Zapata-Esparza, Apolinar Martínez-Melchor, María Cristina Ortega-Moo, Rubicelia Vargas y Jorge Garza  
Departamento de Química  
Universidad Autónoma Metropolitana-Iztapalapa





### Resumen

En este reporte se presentan brevemente los métodos de la química cuántica que dominan en estos días para el estudio de la estructura electrónica de átomos, moléculas, superficies o sólidos y las exigencias que tienen del uso del supercómputo. Se presentan algunos detalles que muestran la necesidad de cómputo que procese una gran cantidad de datos y haga millones de operaciones en tiempos razonables. Se discute cómo las tarjetas gráficas (GPU por el término en inglés Graphics Processing Unit) han incursionado en el supercómputo, como aceleradores de procesos que son paralelizables. Finalmente, presentamos una aplicación que se ha generado, en la Universidad Autónoma Metropolitana-Iztapalapa, alrededor de los GPU que sirve para el análisis del enlace químico en moléculas de tamaño considerable.

### Abstract

This report briefly presents the quantum chemistry methods which are used currently, to study the electronic structure of atoms, molecules, surfaces or solids, and their necessity to use supercomputers. Some details are discussed to show the large amount of information, and operations, involved in such methods and consequently why powerful computers are required to obtain answers in reasonable times. We also discuss how graphics processing units (GPU) are taking part in high performance computing in order to accelerate processes in parallel. Finally, we present an application that has been developed at the Universidad Autónoma Metropolitana-Iztapalapa, around GPU, which is used to analyze the chemical bond in molecules with considerable size.

### 1. Introducción

En la actualidad, es difícil concebir un trabajo de investigación en química donde no esté involucrado el trabajo experimental y teórico. Por parte del trabajo teórico, se ha inclinado la balanza al trabajo computacional, donde los métodos de la química cuántica pueden ser aplicados de manera casi rutinaria haciendo uso de cómputo conocido como supercómputo, o también como cómputo de alto rendimiento. Naturalmente, este tipo de cómputo requiere de una gran inversión y por esa razón solamente universidades o empresas pueden instalar laboratorios de cómputo con este tipo de arquitectura

computacional. Sin embargo, en los últimos años ha aparecido en escena una arquitectura de cómputo, basada en tarjetas gráficas, que solía usarse solamente en las consolas de videojuegos o en computadoras personales dedicadas a videojuegos o a aplicaciones que requieren de alta definición en imágenes. Ahora, es posible hacer cálculos relacionados con la química computacional haciendo uso de tarjetas gráficas, también conocidas como GPU (por las siglas en inglés de Graphics Processing Units), abatiendo así costos. En este reporte haremos un recuento del impacto que han tenido los GPU en la química cuántica y el desarrollo que ha realizado el Departamento de Química de la Universidad Autónoma Metropolitana-Iztapalapa (UAMI) alrededor de esta nueva forma de hacer cómputo científico. Para este fin en la sección 2 hacemos una revisión de los métodos de la química cuántica usados en estos días. Sobre todo, se pone énfasis en la cantidad de operaciones que se requieren. En la sección 3 se discute cómo el supercómputo ha ayudado a la implementación de los métodos de la química cuántica para que los cálculos involucrados estén al alcance de cualquier investigador, teórico o experimental. En esta sección se menciona la incursión de los GPU en el cómputo científico. Finalmente, en la sección 4 se presenta una aplicación, desarrollada en el Departamento de Química de la UAMI, basada en GPU y cómo computadoras personales dedicadas a videojuegos pueden ser usadas para cómputo científico.

### 2. Métodos de la química cuántica

Muchas de las aplicaciones de la química cuántica se centran en la solución de la ecuación de Schrödinger independiente del tiempo (Atkins y de Paula, 2008; Levine, 2001),

$$\hat{H}\Psi = E\Psi. \quad (1)$$

Resolver este problema significa que dado el operador hamiltoniano,  $\hat{H}$ , se deben de encontrar sus funciones propias y sus valores propios. En los cursos básicos de mecánica o química cuántica se tratan problemas donde el hamiltoniano contiene fuerzas que actúan sobre una partícula. Sin embargo, cuando se tratan átomos o moléculas, el hamiltoniano está compuesto por fuerzas relacionadas con la interacción electrón-electrón, electrón-núcleo y núcleo-núcleo. Consecuentemente, la solución de la ecuación 1 se vuelve extremadamente compleja y

son necesarias varias aproximaciones.

A continuación daremos un resumen de las teorías más usadas hoy en día para el estudio de la estructura electrónica de átomos, moléculas, superficies y sólidos. Si el lector desea entrar en detalles de estas teorías entonces debe de leer la siguiente sección. Sin embargo, la intención de este reporte es mostrar el impacto que ha tenido el supercómputo en la química cuántica y por lo tanto no se necesitan los detalles de los métodos involucrados, por lo tanto se puede ir a la sección 3 y cuando se requiera regresar a la sección 2.

### Aproximaciones implicadas en los métodos de la química cuántica para sistemas de varios electrones

1. En una molécula, la función de onda depende explícitamente de las coordenadas de los núcleos y de los electrones. Además, existe el principio de exclusión de Pauli que tiene como consecuencia que cada electrón dependa de una variable más, esta variable es conocida como espín. Así, si nuestra molécula de estudio está formada por 12 núcleos y 42 electrones entonces la función de onda dependerá de 204 variables, 36 por parte de los núcleos y 168 para los electrones. Max Born y Robert Oppenheimer (Born y Oppenheimer, 1927), desacoplaron el movimiento de los núcleos con el de los electrones, esencialmente, la aproximación de Born-Oppenheimer se basa en que el movimiento de los núcleos es mucho más lento que el exhibido por los electrones y por lo tanto se considera que los núcleos están fijos. De esta manera, la función de onda tendrá dependencia explícita en las coordenadas de los electrones y de manera implícita en las coordenadas de los núcleos.
2. Para un sistema con dos electrones o más, la ecuación de Schrödinger no admite una solución analítica y por lo tanto, solamente se tienen aproximaciones a la función de onda. Muchas de estas aproximaciones están sustentadas sobre el principio variacional, el cual dentro de la mecánica cuántica enuncia que si la energía del sistema es estimada por una función de onda que no corresponde a la función de onda exacta entonces su valor será mayor al de la energía exacta (Levine, 2001). Este principio que tie-

ne mucho sentido, y es intuitivo, es una herramienta básica para generar aproximaciones de la función de onda.

### 2.1. Método de Hartree-Fock

El método de Hartree-Fock (HF) es uno de los métodos más usados hoy en día para el estudio de la estructura electrónica de átomos y moléculas. (Levine (2001)) Este método está basado en la aproximación de Born-Oppenheimer y en el principio variacional. (Levine (2001)) La función de onda que se propone es escrita como un determinante. Por ejemplo, para un sistema de tres electrones se tiene,

$$\Psi^{HF}(\vec{x}_1, \vec{x}_2, \vec{x}_3) = \frac{1}{\sqrt{3!}} \begin{vmatrix} \chi_i(\vec{x}_1) & \chi_j(\vec{x}_1) & \chi_k(\vec{x}_1) \\ \chi_i(\vec{x}_2) & \chi_j(\vec{x}_2) & \chi_k(\vec{x}_2) \\ \chi_i(\vec{x}_3) & \chi_j(\vec{x}_3) & \chi_k(\vec{x}_3) \end{vmatrix} \quad (3)$$

Es claro que el determinante se contruye con las funciones  $\chi_l$  en este caso cada una de estas funciones representa a un electrón del sistema, y depende de las coordenadas  $\vec{x}$  que representan a las coordenadas  $x, y, z$  y la coordenada de espín  $\omega$ . Estas funciones son conocidas como orbitales, por que describen a cada electrón, y se construyen como el producto de una función que depende de la parte espacial,  $\phi(\vec{r})$ , por una función que depende de la parte de espín,  $\sigma(\omega) = \alpha(\omega), \beta(\omega)$ . Para un sistema de  $N$  electrones el determinante será de  $N \times N$ , a este determinante se le conoce como determinante de Slater ya que John Slater propuso esta forma de la función de onda (Levine, 2001). La razón de que sea un determinante es por que así la función de onda satisface el principio de exclusión de Pauli. Siguiendo con el principio variacional, lo que se hace es estimar a la energía con la función de onda representada por un determinante de Slater, de esta manera, la expresión de la energía resultante queda en términos del conjunto de orbitales de la siguiente forma

$$E_{HF} = \sum_{i=1}^N \langle \chi_i | -\frac{1}{2} \nabla^2 + v | \chi_i \rangle + \frac{1}{2} \sum_{a=1}^N \sum_{b=1}^N (\langle ab | ab \rangle - \langle ab | ba \rangle). \quad (3)$$

donde

$$\langle \chi_i | -\frac{1}{2} \nabla^2 + v | \chi_i \rangle = \int d\vec{x} \chi_i^*(\vec{x}) \left( -\frac{1}{2} \nabla^2 + v \right) \chi_i(\vec{x}), \quad (4)$$

con

$$v(\vec{r}) = - \sum_{A=1}^M \frac{Z_A}{|\vec{R}_A - \vec{r}|}. \quad (5)$$

Estos dos últimos términos representan la contribución a la energía cinética y a la interacción núcleo electrón de cada orbital. El segundo término de la ecuación 3 se obtiene al reconocer que

$$\langle ab | ab \rangle = \iint d\vec{x} d\vec{x}' \frac{\chi_a^*(\vec{x}) \chi_b^*(\vec{x}') \chi_a(\vec{x}) \chi_b(\vec{x}')}{|\vec{r} - \vec{r}'|}, \quad (6)$$

$$\langle ab | ba \rangle = \iint d\vec{x} d\vec{x}' \frac{\chi_a^*(\vec{x}) \chi_b^*(\vec{x}') \chi_b(\vec{x}) \chi_a(\vec{x}')}{|\vec{r} - \vec{r}'|}. \quad (7)$$

Ambas integrales contribuyen a la interacción entre electrones, la primera tiene una interpretación clásica y es conocida como integral coulombica y la otra es conocida como integral de intercambio. Finalmente, los orbitales de HF que se utilizan para evaluar la energía de la ecuación 3 son aquellos que la minimiza, para que suceda esto se de cumplir que

$$\left( -\frac{1}{2} \nabla^2 + v + v^{HF} \right) \phi_i = \epsilon_i \phi_i \quad (8)$$

donde  $v^{HF}$  representa el potencial que contiene la interacción de un electrón con los restantes electrones promedia da por la parte coulombica y de intercambio.

### 2.1.1. Funciones de base

Para obtener los orbitales que minimizan la energía de HF (solución de la ecuación 8) se utiliza el método de combinación lineal de funciones de base. Este método está fundamentado en representar a una función, digamos  $\phi(\vec{r})$ , en término de un conjunto

de funciones conocidas, digamos  $\{f_\mu(\vec{r})\}$ .

$$\phi_i(\vec{r}) = \sum_{\mu=1}^K c_\mu^{(i)} f_\mu(\vec{r}). \quad (9)$$

En este caso, las funciones  $f_\mu$  son conocidas. Comúnmente, para átomos y moléculas estas funciones son gaussianas de la forma

$$f_\mu(\vec{r}) = (x - X)^l (y - Y)^m (z - Z)^n e^{-\alpha_\mu [(x-X)^2 + (y-Y)^2 + (z-Z)^2]} \quad (10)$$

donde  $X, Y, Z$  representa la posición de un núcleo donde la gaussiana está centrada. Para encontrar los coeficientes  $c_\mu$  se procede a construir una ecuación algebraica.

La ecuación 8 se puede escribir en forma algebraica como

$$\mathbf{F} \vec{c}_i = \epsilon_i \mathbf{S} \vec{c}_i \quad (11)$$

donde los vectores  $\vec{c}_i$  tienen como componentes a los coeficientes  $c_\mu$ 's. Los elementos de matriz de las matrices S y F se obtienen de

$$(\mathbf{S})_{\mu\nu} = (\mu | \nu) = \int d\vec{r} f_\mu^*(\vec{r}) f_\nu(\vec{r}) \quad (12)$$

y para el caso donde se tienen pares de electrones

$$(\mathbf{F})_{\mu\nu} = (\mu | -\frac{1}{2} \nabla^2 + v | \nu) + \sum_{\sigma, \lambda=1}^K 2 \sum_{a=1}^{N/2} c_{\lambda a} c_{\sigma a}^* \left[ (\mu \nu | \sigma \lambda) - \frac{1}{2} (\mu \lambda | \sigma \nu) \right] \quad (13)$$

con

$$(\mu | -\frac{1}{2}\nabla^2 + v | \nu) = \int d\vec{r} f_{\mu}^*(\vec{r}) \left( -\frac{1}{2}\nabla^2 + v \right) f_{\nu}(\vec{r}) \quad (14)$$

y

$$(\mu \lambda | \sigma \nu) = \iint d\vec{r}_2 d\vec{r}_1 \frac{f_{\mu}(\vec{r}_1) f_{\lambda}(\vec{r}_2) f_{\sigma}(\vec{r}_1) f_{\nu}(\vec{r}_2)}{|\vec{r}_2 - \vec{r}_1|}. \quad (15)$$

A continuación hacemos un resumen del método de Hartree-Fock:

1. Elegir un conjunto de  $K$  funciones de base para representar a los orbitales de Hartree-Fock, ecuaciones 9 y 10.

2. Evaluar las integrales:

- a) de la matriz  $S$  (son  $K \times K$  integrales), ecuación 12.
- b)  $(\mu | -\frac{1}{2}\nabla^2 + v | \nu)$  que aparecen en la matriz  $F$  (son  $K \times K$ ), ecuación 13.
- c)  $(\mu \lambda | \sigma \nu)$  que aparecen en la matriz  $F$  (son  $K^4$  integrales!), ecuación 13.

3. Construir la matriz  $F$ , para este fin son necesarios los coeficientes  $c_{\mu}$ 's. En el primer paso de este proceso podemos suponer que los coeficientes son cero (no hay interacción electrón-electrón).

4. Resolver el problema algebraico de la ecuación 11 para encontrar los coeficientes  $c_{\mu}$ 's.

5. Determinar la energía total.

6. Regresar al paso 5 y continuar el proceso hasta que la energía total no tenga cambios apreciables, teniendo así un proceso iterativo.

## 2.2. Métodos correlacionados

El método de Hartree-Fock es una buena aproximación a la función de onda exacta. Sin embargo, no deja de ser una aproximación y es difícil de creer que la función de onda representada por un solo determinante contenga toda la información de un sistema polieletrónico. Esto motiva a que la función de onda sea representada en muchas ocasiones por más de un determinante. A la diferencia entre

la energía exacta y la energía de Hartree-Fock se le conoce como energía de correlación (Levine, 2001). El uso de métodos correlacionados lleva en muchas ocasiones la evaluación de integrales del tipo

$$(abcd) = \iint d\vec{r}_1 d\vec{r}_2 \frac{\phi_a^*(\vec{r}_1) \phi_b^*(\vec{r}_2) \phi_c(\vec{r}_1) \phi_d(\vec{r}_2)}{|\vec{r}_1 - \vec{r}_2|} \\ = \sum_{\mu=1}^K \sum_{\nu=1}^K \sum_{\lambda=1}^K \sum_{\sigma=1}^K c_{\mu}^{(a)*} c_{\nu}^{(b)*} c_{\lambda}^{(c)} c_{\sigma}^{(d)} \iint d\vec{r}_1 d\vec{r}_2 \frac{f_{\mu}^*(\vec{r}_1) f_{\nu}^*(\vec{r}_2) f_{\lambda}(\vec{r}_1) f_{\sigma}(\vec{r}_2)}{|\vec{r}_1 - \vec{r}_2|} \quad (16)$$

A esta operación se le conoce como transformación de 4 índices e implica dos tareas demandantes de cómputo: 1) Evaluar integrales bielectrónicas con las funciones de base ( $\approx K^4$ ). 2) Realizar las cuatro sumas, donde cada una contiene  $K$  elementos, por lo tanto se tienen  $K^4$  términos. Por ejemplo, si se tienen mil funciones de base,  $K = 1000$ , entonces hay que evaluar del orden de  $10^{12}$  integrales y realizar una suma con  $10^{12}$  términos. Estas operaciones están asociadas a solamente una transformación de 4 índices. Sin embargo, en los métodos correlacionados hay que hacer un gran número de estas transformaciones, por lo tanto los métodos correlacionados o post Hartree-Fock son métodos muy demandantes computacionalmente hablando, debido a que se necesita una gran cantidad de memoria temporal (RAM) y permanente (disco duro) ya que muchas de las integrales deben de residir en disco duro por la insuficiencia de RAM. Evidentemente, se requieren de procesadores con gran velocidad para procesar rápidamente las operaciones involucradas en la transformación de 4 índices y en ocasiones en el problema de valores propios de matrices enormes.

## 2.3. Teoría de Funcionales de la Densidad

La teoría de funcionales de la densidad (TFD) es una alternativa para el estudio de la estructura electrónica y que no está basada en la función de onda, a cambio, esta teoría se basa en la densidad electrónica. Aún cuando es una teoría que comenzó a ser de amplio uso en la década de los noventa, ésta ha probado ser una buena alternativa para el estudio de la estructura electrónica de átomos, moléculas de tamaño considerable, sólidos o superficies (Sholl, 2009). En su versión más exitosa, esta teoría divide a la energía de la siguiente manera (Levine, 2001)

$$E = T_s + V_{ne} + J + E_{xc}. \quad (17)$$

El primer término estima la energía cinética del sistema, simplemente como la suma de energía cinética de cada uno de los electrones. El segundo término está relacionado con la interacción entre los electrones y los núcleos. Los dos últimos términos están relacionados con la interacción entre los electrones. El término  $J$  toma en cuenta la interacción electrónica como una interacción electrostática y  $E_{xc}$  considera lo que  $T_S$  y  $J$  no pueden describir. A esta cantidad se le conoce como el funcional de intercambio y correlación. En principio, el planteamiento lleva a una teoría exacta. Sin embargo, el funcional de intercambio y correlación no se conoce y por lo tanto se han hecho muchas aproximaciones a esta cantidad (Perdew, 2013). Lo atractivo de esta teoría es que su costo computacional es similar al de Hartree-Fock, y de hecho toda la maquinaria que se usa para el método de Hartree-Fock es usado por la TFD, en particular, la evaluación de integrales como las de las ecuaciones 6 y 7 también están involucradas en esta teoría.

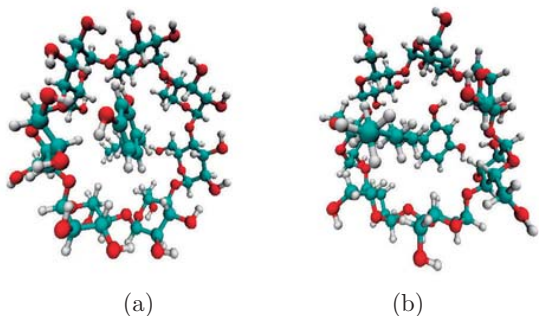


Figura 1: Interacción entre la  $\beta$ -CD y el 4-hexilresorcinol. a) Vista frontal, b) Vista posterior.

### 3. Uso de supercómputo

De la sección anterior es indiscutible que se requiere de una computadora para llevar a cabo el procedimiento involucrado en los métodos de la química cuántica. Claramente, uno de los pasos donde se demanda de cómputo potente es en la evaluación de las integrales bielectrónicas, ecuación 15. Por ejemplo, en un cálculo donde se usen 100 funciones de base ( $K = 100$ ), se requieren aproximadamente de  $K^4$  integrales, o sea, aproximadamente  $10^8$  integrales. La evaluación de estas integrales para átomos en un código de uso amplio, como *Mathematica*®, se publicó en esta revista para apoyo a los cursos

de química cuántica a nivel básico (Garza, 2011). Solamente para ser enfáticos sobre este problema, en los días en que se estaba preparando este artículo se estaba trabajando en nuestro laboratorio con la interacción de la  $\beta$ -ciclodextrina ( $\beta$ CD) con el 4-hexilresorcinol (4-HR), vea se la Figura 1. En ese caso se tienen 2257 funciones de base. Muchas de las integrales son iguales entre sí o algunas de ellas son cercanas a cero. Entonces, usando todas las formas de evitar la evaluación de integrales resulta que se deben de evaluar aproximadamente  $3 \times 10^{12}$  integrales. Al problema de la evaluación de integrales, debemos de añadirle la solución de la ecuación 11, el cual es conocido como un problema de valores propios generalizado, y que en este caso ya no es despreciable. De aquí, no hay duda de que se requiere de una computadora, con el software adecuado, que permita realizar este trabajo en un tiempo razonable.

En nuestros días existe un tipo de cómputo que sobresale de lo común y se le conoce como supercómputo. Este tipo de computadoras sobresale por que puede realizar el mayor número de operaciones, entre números reales, en su época. A las operaciones entre números reales, en una computadora, se le conoce como operaciones de punto flotante. Con la mención anterior, en 1993 se creó una lista de las 500 supercomputadoras más rápidas en el mundo (Top500, 2014). Ciertamente, una supercomputadora de 1993 es superada fácilmente por una computadora de escritorio de ahora. Naturalmente, no podemos decir que esa computadora de escritorio es una supercomputadora por que no es la que hace más operaciones de punto flotante en su época. Por cierto, la UAMI ha instalado supercomputadoras que han estado en la lista de las 500 supercomputadoras más rápidas del mundo, la última fue instalada en el año 2008 y fue la computadora más rápida de América Latina (La Jornada, 2008). No es difícil imaginar el costo de instalación de una supercomputadora y por esa razón se instalan en universidades o en empresas con las instalaciones apropiadas y con la infraestructura para darles un mantenimiento adecuado.

El supercómputo ha tenido un impacto importante en la química computacional y es ahí donde más software se ha desarrollado. Cuando los desarrolladores de software, dentro del ámbito de la química cuántica, tuvieron la posibilidad de contar con muchos CPU (por las siglas en inglés de Central Processing Units) se preguntaban si era mejor contar con un procesador muy rápido en lugar de contar

con varios no tan rápidos. Esto surgió por que muchas aplicaciones estaban implementadas sobre el uso de un solo procesador y el cambio a usar cómputo en paralelo no es trivial por que se usan diferentes técnicas de programación. Ahora, esa discusión ha sido superada por que se implementan las aplicaciones sobre muchos CPU cuando es posible, por supuesto, sobre CPU lo más rápidos que se tengan al alcance. Naturalmente, la tecnología ha jugado un papel crucial en el supercómputo por que en lugar de usar un procesador muy poderoso, las compañías han decidido construir los procesadores con varios núcleos (cores en inglés) para reducir el gasto energético. Por ese motivo vemos que el mercado oferta procesadores multicore. Por la parte del cómputo en paralelo aplicado a la química computacional, el paquete computacional NWChem es un buen ejemplo, ya que ha sido diseñado para utilizar de manera eficiente este tipo de cómputo (Valiev, et al.). Para la interacción  $\beta$ CD-4HR, el programa NWChem instalado en la supercomputadora de la UAMI promedia por iteración, dentro del ciclo iterativo del método de Hartree-Fock,<sup>1</sup> el tiempo reportado en la Tabla 1. En este caso los cálculos se realizaron sobre procesadores Xeon E5-2670 v2@2.50GHz.

Tabla 1: Tiempo promedio usado por iteración, en segundos, por NWChem para obtener la energía total de la  $\beta$ CD interaccionando con el 4-HR, usando el método  $HF/6-311G^{**}$ .

Procesadores	Tiempo (s)
10	6201
20	3097
40	1572
80	814

Es claro de esta tabla que NWChem + (sistema instalado en la UAMI) proveen una buena opción para estudiar la estructura electrónica de sistemas de tamaño considerable, debido a que permiten un buen escalamiento, ya que cuando se dobla el número de procesadores el tiempo invertido se disminuye a la mitad. En muchas ocasiones, se cuenta con un gran número de procesadores pero no con el software que los use correctamente, y en otras ocasiones se cuenta con el software pero no con un gran número de procesadores. Por fortuna en la UAMI se cuenta con

<sup>1</sup> Usaremos el método de Hartree-Fock para comparar los resultados obtenidos con NWChem y otros códigos.

ambas cosas.

#### 4. Tarjetas gráficas como ayuda al supercómputo

En los últimos años, las tarjetas gráficas han sido utilizadas para el cómputo científico. La razón de esto es muy simple, esta tecnología fue diseñada, desde su origen, para ejecutar tareas en paralelo. Tomando en cuenta que las tarjetas gráficas fueron diseñadas para hacer despliegue gráfico, es claro que hacer una gráfica, por ejemplo, un automóvil, debe de hacerse de manera simultánea. De no ser así veríamos primero las llantas, luego un parabrisas, después el cofre y así sucesivamente. Naturalmente, sería visualmente muy desagradable. Por ese motivo estas tarjetas fueron hechas para trabajo que pareciera ser elaborado en un paso (aunque es claro que requiere esfuerzo desplegar una llanta, un cofre, un parabrisas, etc.) y de manera intrínseca está involucrado el cómputo en paralelo. Uno de los problemas que se tenía para el uso de tarjetas gráficas en problemas científicos, fue que la manera de programar no era de alto nivel. Por fortuna ya se tienen dos lenguajes de programación de propósito general: a) CUDA, (Farber, 2011; Sanders y Kandrot, 2010) b) OpenCL (Scarpino, 2012). CUDA ha sido diseñado y mantenido por la empresa NVIDIA®, naturalmente está pensado para que sea usado en tarjetas gráficas producidas por esa compañía, lo cual representa una limitante ya que existen en el mercado otras dos compañías bien consolidadas que ofrecen tarjetas gráficas; AMD® e Intel®. Lo interesante de CUDA es que se han desarrollado varias herramientas y compiladores robustos para los desarrolladores de software. Por otro lado, OpenCL es el lenguaje para la programación heterogénea (CPU-GPU), y está respaldado por el grupo KHRONOS (Khronos, 2014), en cuyo consorcio se encuentra NVIDIA, Intel, AMD, entre otros. Este lenguaje de programación puede ser utilizado en casi cualquier tarjeta gráfica, incluso en aquellas que traen los teléfonos celulares o tabletas de uso comercial. Infortunadamente, siendo un nuevo estándar de programación aun no existe soporte similar al de CUDA y tampoco se han generado aplicaciones y librerías que ayuden a los desarrolladores de software.

Por parte de CUDA se han generados códigos que han implementado métodos de la química cuántica para ser ejecutados sobre tarjetas gráficas. Terachem es el código que sobresale de los que se han reportado hasta el momento, fue creado por el pro-



fesor Todd Martínez de la Universidad de Stanford y ha sido diseñado con el concepto de ser ejecutado sobre tarjetas gráficas (Umtsev et al., 2008). En la Tabla 2 se reporta el tiempo usado, en promedio, por cada iteración del sistema  $\beta$ CD+4-HR en Terachem. En este caso los resultados fueron obtenidos con tarjetas gráficas Tesla M-2090.

Tabla 2: Tiempo usado, en segundos, por Terachem para obtener la energía total de la  $\beta$ CD interactuando con el 4-HR, usando el método  $HF/6-311G^{**}$ .

GPU	Tiempo (s)
1	1389
2	732
4	395

De acuerdo a esta tabla un GPU Tesla M-2090 realiza el trabajo de aproximadamente 50 procesadores Xeon E5-2670 v2@2.50GHz usados por NWChem. Posiblemente esta comparación no es del todo justa ya que la evaluación de las integrales bielectrónicas se realizan de diferente forma en cada aplicación, sobre todo Terachem juega con la precisión numérica y por esa razón evalúa más rápido este tipo de integrales. Por lo tanto, existe una diferencia numérica entre los resultados generados por Terachem y NWChem. Por ejemplo, para nuestro sistema  $\beta$ CD+4-HR, y el método  $HF/6-31++G^{**}$ , NWChem computa una energía de -4866.138491417190 unidades atómicas mientras que el Terachem da como resultado -4866.1331308678 unidades atómicas, esta diferencia representa alrededor de 3 kcal/mol, mostrando así que la diferencia de energía es importante cuando se comparan energías totales. Sin embargo, cuando se comparan diferencias de energía, por ejemplo en la evaluación de la energía de interacción entre la  $\beta$ CD y el 4-HR, NWChem predice un cambio de energía de -1.3 kcal/mol, mientras que el Terachem predice -0.5 kcal/mol. En este caso la diferencia de 3 kcal/mol entre los dos códigos se ve reducida a 0.8 kcal/mol. Dicha diferencia es menor a la precisión de muchos de los métodos de la química cuántica con los que se predicen energías de enlace, sobre todo en interacciones débiles.

NWChem ha entrado también en escena para el uso de GPU. Recientemente se ha reportado la aceleración de un método correlacionado basado en la función de onda (cúmulos acoplados) dentro del NWChem (DePrince y Hammond, 2011). El desem-

peño que se muestra en la aplicación es alentador para pensar que los GPU llegaron para quedarse en aplicaciones de química cuántica. No será sorprendente que en algunos años los GPU estén involucrados en el desarrollo de muchos códigos de química cuántica, como lo están ahora GAMESS (Gordon et al., 2014) o Gaussian (Gomperts et al., 2014). Naturalmente, el departamento de Química de la UAMI ha contribuido a generar código sobre GPU para la transformación de 4 índices para acelerar métodos que involucran a éstas (Carmona-Espíndola et al., 2013; García-Hernández et al., 2013, 2014).

## 5. Análisis de la densidad electrónica

### 5.1. Evaluación de campos escalares

Naturalmente, el estudio de la estructura electrónica no se queda solamente en el cálculo de la energía o de la función de onda. Por ejemplo, si estamos interesados en saber los sitios ricos en electrones, o de cientos de ellos, dentro de una molécula, lo primero que se puede hacer es una gráfica de esta cantidad. En general, los métodos de estructura electrónica evalúan a la densidad electrónica,  $\rho(\vec{r})$ , de la forma

$$\rho(\vec{r}) = \sum_{i=1}^{noc} \omega_i \phi_i^*(\vec{r}) \phi_i(\vec{r}) \quad (18)$$

donde  $\omega_i$  representa la ocupación del orbital  $i$  y  $noc$  representa el número de orbitales ocupados.

Típicamente, para hacer una gráfica de esta cantidad se recurre a una malla. Esto es, el sistema de estudio se envuelve por un cubo o un paralelepípedo y se hacen cubos dentro de él. Sobre cada vértice de los cubos internos se evalúa a la densidad electrónica. Para nuestro ejemplo,  $\beta$ CD+4-HR, si se usa una malla mediana (5 puntos por cada unidad atómica) entonces el número de puntos donde será evaluada la densidad electrónica es igual a 4,752,000. Veámos el número de operaciones que debemos de realizar por cada punto de la malla. Para nuestro ejemplo,  $noc = 354$  y  $K = 3616^2$ . De la ecuación 9, vemos que debemos de evaluar una suma con 3616 términos, al resultado lo elevamos al cuadrado (de acuerdo a la ecuación 18). El penúltimo paso es multiplicar por la ocupación, en nuestro caso es igual a 2. Finalmente se suma sobre los orbitales ocupados (354), de

<sup>2</sup> No se está usando ningún esquema de contracción, se usan directamente funciones primitivas.

acuerdo a lo que indica la ecuación 18. Por lo tanto, se deben hacer  $(3616 \text{ sumas}) \times (2 \text{ multiplicaciones}) \times (354 \text{ sumas}) = 2,560,128$  operaciones de punto flotante. Este número debe de ser multiplicado por el número de puntos en la malla. En total se tienen que realizar  $12,165,728,256,000$  operaciones de punto flotante. A este resultado tendríamos que agregarle las operaciones que están involucradas dentro de la evaluación de  $f_\mu$  (ecuación 10). Evidentemente, se requiere de una computadora muy rápida para hacer estas operaciones. Un proyecto que estamos desarrollando en la

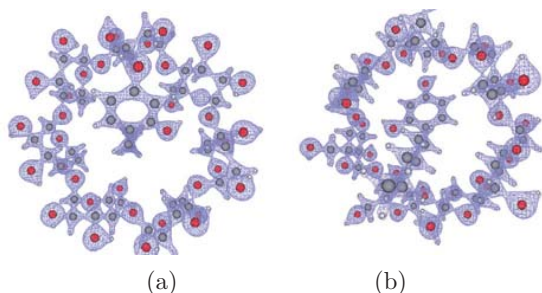


Figura 2: Densidad electrónica del sistema  $\beta$ -CD + 4-HR. a) Vista frontal, b) Vista posterior.

UAMI es el de evaluar campos escalares y vectoriales de la química cuántica, la densidad electrónica es uno de esos campos, sobre GPU. La idea del proyecto es la de evaluar orbitales, el gradiente y el laplaciano de la densidad electrónica, la función de localización electrónica (Becke y Edgecombe, 1990), la densidad de energía cinética (Navarrete-López et al., 2008) y el potencial electrostático (Murray et al., 2011), entre otros. En la Figura 2 presentamos la densidad electrónica de nuestro sistema  $\beta$ CD+4-HR. Es importante mencionar que el despliegue gráfico está realizado con nuestra aplicación, que también está desarrollada para que se ejecute sobre GPU. Las esferas rojas representan átomos de oxígeno, las grises a átomos de carbono y las blancas a átomos de hidrógeno. La manera en que estamos graficando este campo es sin suponer una posible conexión entre los átomos, para representar los enlaces que hay en el sistema. La conectividad es un invento de los químicos, y aún cuando muchos de ellos son muy inteligentes, en ocasiones su intuición puede fallar y por eso han sido reportados modelos matemáticos para tener una conectividad fundamentada en teoremas y no en la intuición química, como lo veremos en la siguiente sección. Aún con la limitación de no presentar la conectividad entre los átomos, es evidente de la Figura 2 que a partir

de la densidad electrónica es posible rescatar lo que la intuición química nos indicaba desde un inicio, los enlaces químicos que conforman la  $\beta$ CD+4-HR, pero ahora de manera no trivial y sustentada en un observable.

## 5.2. Analizando el enlace químico

Existe un método relacionado con el análisis de la densidad electrónica que también involucra el uso de supercómputo. Este método frecuentemente es utilizado para encontrar los posibles enlaces en una molécula y se conoce como el método de átomos en moléculas (AIM, del inglés atoms in molecules) propuesto por el profesor Richard Bader (Bader, 1990, 1994). Este método está basado en encontrar los puntos críticos de la densidad electrónica. En esta revista uno de los autores de este reporte ha publicado una manera de encontrar los puntos críticos de una función de varias variables (Garza, 2007). En este caso la densidad electrónica,  $\rho(\vec{r})$ , depende de tres coordenadas,  $x, y, z$ . Los puntos críticos,  $\vec{r}_c$ , de esta cantidad son aquellos donde

$$\left. \frac{\partial \rho(\vec{r})}{\partial x} \right|_{\vec{r}_c} = 0, \quad \left. \frac{\partial \rho(\vec{r})}{\partial y} \right|_{\vec{r}_c} = 0, \quad \left. \frac{\partial \rho(\vec{r})}{\partial z} \right|_{\vec{r}_c} = 0, \quad (19)$$

o en forma condensada

$$\nabla \rho(\vec{r})|_{\vec{r}_c} = \vec{0}. \quad (20)$$

Parte del proyecto que hemos emprendido sobre el uso de GPU en la UAMI, es el de encontrar los

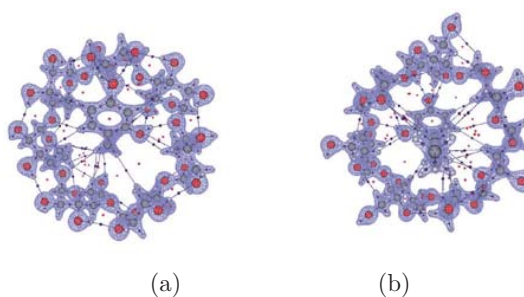


Figura 3: Densidad electrónica y trayectorias de enlace entre la  $\beta$ -CD y el 4-HR. a) Vista frontal, b) Vista posterior.

puntos críticos de la densidad electrónica (Hernández-Esparza et al., 2014). Nuevamente, usando nuestro sistema  $\beta$ CD+4-HR, hemos encontrado los



puntos críticos de la densidad electrónica asociados a esta molécula. En la Figura 3, se muestran los puntos críticos (representados por esferas pequeñas) exhibidos en este sistema. Las esferas azules pequeñas representan puntos críticos de enlace. De acuerdo a estos puntos, en la Figura 3a, en la parte de arriba, se pueden apreciar puentes de hidrógeno ya que se tiene un punto crítico de enlace entre los átomos O-H-O. La caracterización de este tipo de interacciones ha sido discutida por A. Navarrete y colaboradores (Navarrete-López et al., 2007).

Además de los puntos críticos, se presentan las trayectorias de enlace en la Figura 3, las cuales se obtienen al trazar una línea de un punto crítico de enlace hasta el valor máximo que puede alcanzar la densidad electrónica. Esta trayectoria está involucrada en la solución de una ecuación diferencial. En el caso del sistema  $\beta$ CD+4-HR se encontraron 225 trayectorias de enlace, lo cual significa que se tuvieron que resolver 225 ecuaciones diferenciales. La solución de estas ecuaciones es lo que hemos trazado con líneas delgadas. Estas líneas muestran que el 4-HR está formando enlaces (interacciones) con la  $\beta$ CD. Así, la conectividad entre átomos no la dejamos a nuestra imaginación de químicos sino que lo hacemos a través del modelo matemático involucrado con el método de AIM.

El costo computacional del método usado en (Hernández-Esparza et al., 2014) para la búsqueda de puntos críticos en la densidad electrónica se discute en esa referencia y la parte importante que deseamos mencionar es que una tarjeta gráfica diseñada para videojuegos, como la GForce GTX 760® puede ser usada para esta aplicación y compete muy bien con respecto al desempeño que exhibe una tarjeta gráfica dedicada al supercómputo. En ese sentido, los GPU están dando la posibilidad de realizar cómputo científico con costos que están muy por debajo de lo que significa instalar servidores basados exclusivamente en núcleos de CPU.

## 6. Conclusiones

Se ha presentado la forma en que el cómputo forma parte de la investigación rutinaria dentro de la química cuántica, resumiendo las metodologías comúnmente empleadas y haciendo especial hincapié en la enorme cantidad de datos a calcular y en el porqué el empleo del supercómputo. Además, como parte fundamental de este trabajo, se presenta el uso actual de los GPU en la química cuántica y el trabajo efectuado en la UAMI para la generación de programas que hagan uso de estas nuevas tec-

nologías de procesamiento concurrente, como una alternativa viable al empleo de servidores basados en múltiples CPU. Dentro de esta línea de trabajo, se analiza el enlace químico dentro del marco de la teoría de Átomos en Moléculas, donde la densidad electrónica, su gradiente y sus segundas derivadas son evaluadas sobre GPU. Mostrando así el poder de cómputo que ofrecen los GPU y sobre todo que, GPU dedicados a videojuegos pueden ser utilizados para el cómputo científico.

## Agradecimientos

Agradecemos al Laboratorio de Supercómputo y Visualización en Paralelo el acceso a sus instalaciones. A CONACYT por el financiamiento a través de los proyectos 154784, 155070 y 155698, y las becas 283251, 283645 y 371984.

## Referencias

- Atkins, P. y de Paula, J., *Química Física, 8a Edición*, Médica Panamericana, Madrid, 2008.
- Bader, R. F. W., *Atoms in molecules: A Quantum Theory*, Oxford University Press, New York, 1990.
- Bader, R. F. W., Principle of stationary action and the definition of a proper open system, *Phys. Rev. B.*, **49**, pp. 13348-13356, 1994.
- Becke, A.D. y Edgecombe, K. E., A simple measure of electron localization in atomic and molecular systems, *J. Chem. Phys.*, **92**, pp. 5397-5403, 1990.
- Born, M. y Oppenheimer, J. R., Quantum theory of molecules, *Ann. Physik*, **84**, pp. 457-484, 1927.
- Carmona-Espíndola, J., Alcalde-Segundo, I., Vargas, R. y Garza, J., Many-body perturbation theory to second order applied on condensed helium like atoms, In *COMPUTATIONAL AND EXPERIMENTAL CHEMISTRY: Developments and applications*. CRC Press, New Jersey, 2013.
- DePrince, A. E. y Hammond, J. R., Coupled cluster theory on graphics processing units I. The coupled cluster doubles method, *J. Chem. Theory Comput.*, **7**, pp. 1287-1295, 2011.
- Farber, R., *CUDA Application Design and Development*. Elsevier: Boston, 2011.
- García-Hernández, E., Díaz-García, C., Vargas, R.

- y Garza, J., Four-index integral transformation in many-body perturbation theory and electron propagator to second order on GPUs for condensed atoms, *AIP Conf. Proc.*, **1558**, pp. 1528-1531, 2013.
- García-Hernández, E., Díaz-García, C., Vargas, R. y Garza, J., Implementation of the electron propagator to second order on GPUs to estimate the ionization potentials of condensed atoms. *J. Phys. B: At. Mol. Opt. Phys.*, **47**, pp. 185007-185014, 2014.
- Garza, J., Búsqueda de puntos críticos y elaboración de gráficos de funciones de varias variables con MATHEMATICA®. *Contactos*, **63**, pp. 22-27, 2007.
- Garza, J., Usando MATHEMATICA® para la evaluación de Integrales bielectrónicas con orbitales hidrogenoides, *Contactos*, **81**, pp. 5-10, 2011.
- Gomperts, R., Frisch, M., Scalmani, G., Lebeck, B., *Current status of the project to enable Gaussian 09 on GPGPUS*, Ponencia presentada en la GPU Technology Conference 2014, Disponible en <http://on-demand.gputechconf.com/gtc/2014/presentations/S4613-enabling-gaussian-09-on-gpgpus.pdf>
- GAMESS, The General Atomic and Molecular Electronic Structure System. En *Mark Gordon's Quantum Theory Group, Ames Laboratory/Iowa State University*. Última consulta el 27 de noviembre de 2014, Disponible en <http://www.msg.ameslab.gov>
- Hernández-Esparza, R., Mejía-Chica, S.- M., Zapata-Escobar, A. D., Guevara-García, A., Martínez-Melchor, A., Hernández-Pérez, J.- M., Vargas, R. y Garza, J., Grid-based algorithm to search critical points, in the electron density, accelerated by graphics processing units, *J. Comput. Chem.*, **35**, pp. 2272-2278, 2014.
- Khronos, OpenCL - The open standard for parallel programming of heterogeneous systems. En Khronos Group Connecting software to silicon. Última consulta el 27 de noviembre de 2014, Disponible en <https://www.khronos.org>
- La Jornada, La UAM pone en marcha la computadora más potente de América Latina. Última consulta el 27 de noviembre de 2014, Disponible en <http://www.jornada.unam.mx/2008/11/29/index.php?section=ciencias&article=a02n1cie>.
- Levine, I. N., *Química Cuántica, 5a Edición*. Pearson Educación, Madrid, 2001.
- Murray, J. N., Politzer, P., The electrostatic potential: an overview, *Comput. Mol. Sci.*, **1**, pp. 153-163, 2011.
- Navarrete-López, A.-M., Garza, J., Vargas, R., Relationship between the critical points found by the electron localization function and atoms in molecular approaches in adducts with hydrogen bonds. *J. Phys. Chem. A.*, **111**, pp. 11147-11152, 2007.
- Navarrete-López, A.-M., Garza, J., Vargas, R., The Kohn-Sham kinetic energy density as indicator of the electron localization: Atomic shell structure, *J. Chem. Phys.*, **128**, pp. 104110-104118, 2008.
- Perdew, J. P., Climbing the ladder of density functional approximations, *MRS Bulletin*, **38**, pp. 743-750, 2013.
- Rys, J., Dupuis, M., King, H. F., Computation of electron repulsion integrals using the Rys quadrature method, *J. Comput. Chem.*, **4**, pp. 154-157, 1983.
- Sanders J. y Kandrot, E., *CUDA by Example: An Introduction to General-Purpose GPU Programming*, NVIDIA Corporation: Boston, 2010.
- Sholl, D. S. y Steckel, J. A. *Density Functional Theory: A practical introduction*. Wiley, New York, 2009.
- Scarpino, M., *OpenCL in Action. How to accelerate graphics and computations*. Manning Publications Co. Shelter Island, 2012.
- Top500, The list. Última consulta el 27 de noviembre de 2014, Disponible en <http://www.top500.org>.
- Umtsev, I. S., Martínez, T. J., Graphical processing units for quantum chemistry, *Comput. Sci. Eng.*, **10**, pp. 26-34, 2008.
- Valiev, M., Bylaska, E. J., Govind, N., Kowalski, K., Straatsma, T. P., Van Dam, H. J. J., Wang, D., Nieplocha, J., Apra, E., Windus, T. L., de Jong, W. A., NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations, *Comput. Phys. Commun.*, **181**, pp. 1477-1489, 2010.



Contents lists available at ScienceDirect

International Journal of Adhesion and Adhesives

journal homepage: [www.elsevier.com/locate/ijadhadh](http://www.elsevier.com/locate/ijadhadh)

## Oligolignols within lignin-adhesive formulations drive their Young's modulus: A ReaxFF-MD study



Pablo López-Albarrán<sup>a,\*</sup>, Antonio Pizzi<sup>b</sup>, Pedro Navarro-Santos<sup>a,c</sup>,  
Raymundo Hernández-Esparza<sup>d</sup>, Jorge Garza<sup>d</sup>

<sup>a</sup> Facultad de Ingeniería en Tecnología de la Madera, Universidad Michoacana de San Nicolás de Hidalgo, Edificio D, C. U., Fco. J. Múgica S/N, 58030 Morelia, Mexico

<sup>b</sup> Ecole Nationale Supérieure des Technologies et Industries du Bois (ENSTIB)<sup>®</sup>, Université de Lorraine, France

<sup>c</sup> CONACYT-Universidad Michoacana de San Nicolás de Hidalgo, Edif. B-1, C.U., Fco. J. Múgica S/N, 58030 Morelia, Mexico

<sup>d</sup> Universidad Autónoma Metropolitana – Iztapalapa, Departamento de Química, División de Ciencias Básicas e Ingenierías, San Rafael Atlixco 186, Col. Vicentina, Iztapalapa, CP 09340 Ciudad de México, Mexico

### ARTICLE INFO

#### Keywords:

Adhesives for wood (A)  
Structure property relations (C)  
Lignin-Adhesive composition  
ReaxFF Oligolignol-Cellulose complexes

### ABSTRACT

Adhesive formulations based on depolymerized lignins were studied to investigate the contribution of their oligolignols content in the determination of its Young's modulus towards the contact with a cellulose surface. Four lignins with low molar mass, tested and reported in an earlier study, were employed to model oligolignol molecular sub-structures. Over these oligolignols, several Reactive Force Field (ReaxFF) molecular dynamic simulations were considered to elucidate the contribution of each molecular structure to the adhesive capacity of the lignin mixture, modelling the interaction of each oligolignol over one model of cellulose I- $\beta$ . One important result obtained from these simulations is, that coupling energies obtained for the oligolignol-cellulose complexes correlate quite well with experimental values of Young's modulus, yielding the insight to describe how oligolignols within lignin-adhesive formulations drive their Young's modulus from an atomistic point of view. This study opens a new perspective to design novel wood adhesive formulations based on lignins, considering the oligolignols present in their composition.

### 1. Introduction

The need of eco-friendly wood adhesives has led to intense research on adhesives derived from natural and non-toxic materials. Many studies about this topic have been reported in recent years [1–9] where lignin is one of the materials at the forefront of such studies.

Considering the fact that lignin is the second major component of wood and its main biological function is to participate as a cementing agent among wood cells [10], chemists and engineers have reported several studies on the interaction between cellulose and lignin in the field of wood adhesives to develop new formulations that reproduce or mimic lignin properties [2–9]. Recently, adhesives that include lignin in their formulation have shown several properties similar to phenol-formaldehyde resins [11–14]. Their role in the wood industry is quite relevant, since they are extensively utilized in plywood, particleboard, fiberboard, or in laminated wood products [15–17]. Lignin-based adhesives have been studied mainly from an empirical point of view; [2,13] consequently the coupling between cellulose and the structural cross-linked 4-hydroxyphenylpropanoids of lignin (oligolignols) that

drive the adhesion phenomenon remain still unclear. In particular, the lignin structure is based on the assumption that this is built from simple units of 4-hydroxyphenylpropane, typically three hydroxycinnamyl alcohols are involved; *p*-coumarol, coniferol, and sinapol (Fig. 1), [2,3] which are commonly named monolignols [18].

Lignin in solution has a different composition with respect to the native lignin (protolignin) found in wood, due to the extraction process used commonly. In general, coniferol units are more abundant in softwood than in hardwood; the later one presents a similar proportion between coniferol and sinapol units [19]. Even with these differences, the three following linkages are the most abundant ones found between lignin units:  $\beta$ -O-4 ( $\beta$ O4),  $\beta$ -5 ( $\beta$ 5) and  $\beta$ - $\beta$  ( $\beta\beta$ ) [20].

Pizzi et al. have reported four low molar mass lignins in new adhesive proposals, where oligolignols are recognized as the principal components in the tested adhesive formulations, identifying the molecular masses of fragments by using MALDI-TOF (matrix assisted laser desorption ionization time-of-flight) mass spectroscopy. Additionally, the values of Young's modulus ( $Y_M$ ) of such adhesives were distributed among the identified oligolignols from their mass percent in each

\* Corresponding author.

E-mail address: [plopez@umich.mx](mailto:plopez@umich.mx) (P. López-Albarrán).

<http://dx.doi.org/10.1016/j.ijadhadh.2017.08.003>

Accepted 31 July 2017

Available online 04 August 2017

0143-7496/© 2017 Elsevier Ltd. All rights reserved.

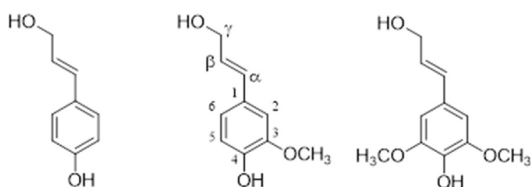


Fig. 1. Lignin precursors: *p*-coumarol, coniferol, and sinapyl, respectively, shown in their predominant (E)-monolignol configuration. Numbering is shown here only on the coniferol structure.

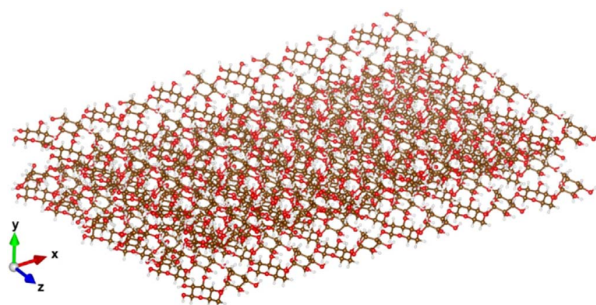


Fig. 2. Cellulose I- $\beta$  model used to form complexes with oligolignols [24,25].

mixture [21]. In that work, a computational molecular docking study was carried out in order to correlate the interaction energy between oligolignols and a cellulose I- $\beta$  crystallite model. These interaction energies were directly related with the experimental  $Y_M$  obtained from thermomechanical analysis. The correlations reported in that study were reasonably good, however, few adhesive mixtures showed low correlations, even though their  $Y_M$  values suggested an efficient adhesion. Naturally, a molecular docking methodology is a first step to understand oligolignol-cellulose complexes, due to limitations of the computational model [22].

Although the weighted distribution of oligolignols developed by Pizzi et al. [21], was suitable to assign the participation of each oligolignol-cellulose complex in the  $Y_M$  of the lignin mixtures suggesting the additivity of the individual  $Y_M$  studies, the molecular structures elucidated by MALDI-TOF results did not yield good correlations concerning the interaction energies from the molecular docking studies. Thus, a molecular dynamics study is desirable to evaluate energetically such oligolignols-cellulose complexes. Indeed, improving the molecular structures suggested from the MALDI-TOF spectroscopy for the oligolignols used in our previous molecular interactions study [23], could verify that the weighted oligolignol distribution of Pizzi et al. [21], supports the postulate of addition in the  $Y_M$  determination, which may be extended to new lignin mixtures for the enhancement of lignin-adhesive formulations.

In this work, a molecular modeling study is presented to characterize the complexes formed between oligolignols of two and three units over a cellulose I- $\beta$  crystallite. The corresponding coupling energies are estimated by using molecular dynamics simulations through a reactive force field. Evidently, the aim of this work is to evidence which oligolignols from the lignin mixtures drive the adhesivity of lignin-based formulations, through the knowledge and enhancement of the molecular structures of oligolignols in the adhesive formulations previously reported [21,23]. We think that; this study will help to describe the molecular interactions for the design of adhesive formulations based on the oligolignols characterized from depolymerized lignins.

To the best of our knowledge, there are no theoretical reports employing the reactive molecular dynamics methodologies available to elucidate and evaluate the interaction between oligolignols from

adhesive formulations and cellulose. In our opinion, this topic is central to obtain guides on the discovering of new eco-friendly adhesive formulations for wood.

## 2. Methodology

Molecular dynamics (MD) simulations were employed to estimate the interaction energy of oligolignols on a cellulose I- $\beta$  crystallite model. Considering that the cellulose I- $\beta$  is the most common form of cellulose found in higher plants [10], the cellulose model used in this work was built from crystalline structures characterized by X-ray and neutron fiber diffraction measurements [24,25]. Our cellulose I- $\beta$  crystallite model consists of 12 D-glucose molecules along the cellulosic axle, 6 D-glucose molecules transverse to the x axis and 3 stacked layers along the y axis ( $6 \times 6 \times 3$  model). This model is presented in Fig. 2.

A simulation box of ( $85 \times 50 \times 60$ ) Å was considered to move the oligolignols over the cellulose I- $\beta$  crystallite model. The molecular structures of oligolignols elucidated by our MALDI-TOF experimental results and improved from our molecular interactions study, were built through the ChemBioOffice ultra software version 11.0.1 [26]. These oligolignols were considered to interact with the cellulose I- $\beta$  crystallite. The set of oligolignols studied in this work is presented in Figs. 3 and 4, where it is clear the relevance of dilignols and trilignols in the proposed models. We have used a general nomenclature for oligolignols: A<sub>m</sub>B<sub>n</sub>C where A, B or C indicates a coniferyl (CA) or sinapyl (SA) alcohol linked by one m or n linkage (for instance,  $\beta$ O4,  $\alpha$  $\beta$ ,  $\beta$  $\beta$ ,  $\gamma$  $\beta$  and  $\beta$ 5 were the considered linkages).

Molecular dynamics simulations were carried out by the reactive force field (ReaxFF) [27], as this is implemented in the open source code LAMMPS [28], with the C/H/O atomic parameters developed by Chenoweth et al. [29] Although there are force fields that include terms to model molecules coupled to charge equilibration [30,31], the ReaxFF applies a bond energy - bond order (BEBO) [32] parameter that depends on the force field, providing accurate descriptions of formation or breaking of bonds in molecular systems. In addition, the ReaxFF has a non-bond interaction term to consider van der Waals and Coulomb interactions. Some works have successfully applied the ReaxFF to describe adequately the reactivity of systems, pathways, organic reactions as well as thermal decomposition and catalytic formation of carbon nanotubes [32,33]. Furthermore, the ReaxFF has been applied recently inside cellulose and lignin studies [34–38], and for the adhesion description of epoxy resins [39], which support the reliability of the parameter set involved in the force field applied in our MD calculations.

At the beginning of each simulation, one oligolignol was placed randomly on the surface of the cellulose model performing updates of positions and velocities for each oligolignol molecule, considering constant numbers of particles, volume, and temperature (NVT). Each trajectory was followed up to 2 ns with a time step of 0.2 fs. After the MD simulation, an energy minimization was performed within a convergence threshold of  $1 \times 10^{-6}$  eV for the energy and  $1 \times 10^{-6}$  eV Å<sup>-1</sup> for the forces. The coupling energy ( $E_c$ ) between each oligolignol and the cellulose surface was estimated in a simulation box, as the total energy difference between the minimized cellulose-oligolignol complex and the sum of the energies of the optimized cellulose surface and the gas-phase oligolignol molecule:

$$E_c = E_{\text{COMPLEX}} - E_{\text{CELLULOSE}} - E_{\text{OLIGOLIGNOL}}$$

The results of the coupling of each oligolignol over the cellulose I- $\beta$  crystallite model from the MD simulations were grouped according the lignin mixture characterized in our previous study [21]. Therefore for each type of lignin mixture, we built four groups: 1) Ln<sub>India</sub>, an industrial organosolv grass lignin containing the following units: C-A $\beta$ O4\_SA $\beta$  $\beta$ \_SA, CA $\beta$  $\beta$ \_CA, SA $\beta$ O4\_SA $\beta$  $\beta$ \_SA and CA $\beta$ O4\_SA $\beta$ 5\_CA, 2) Ln<sub>miscanthus</sub>, a laboratory organosolv lignin from miscanthus (*Miscanthus giganteus*) grass, containing the following oligolignols: C-A $\beta$ O4\_SA, SA $\beta$ O4\_SA, CA $\gamma$ b\_CA, SA $\alpha$  $\beta$ \_SA, CA $\beta$ O4\_CA and

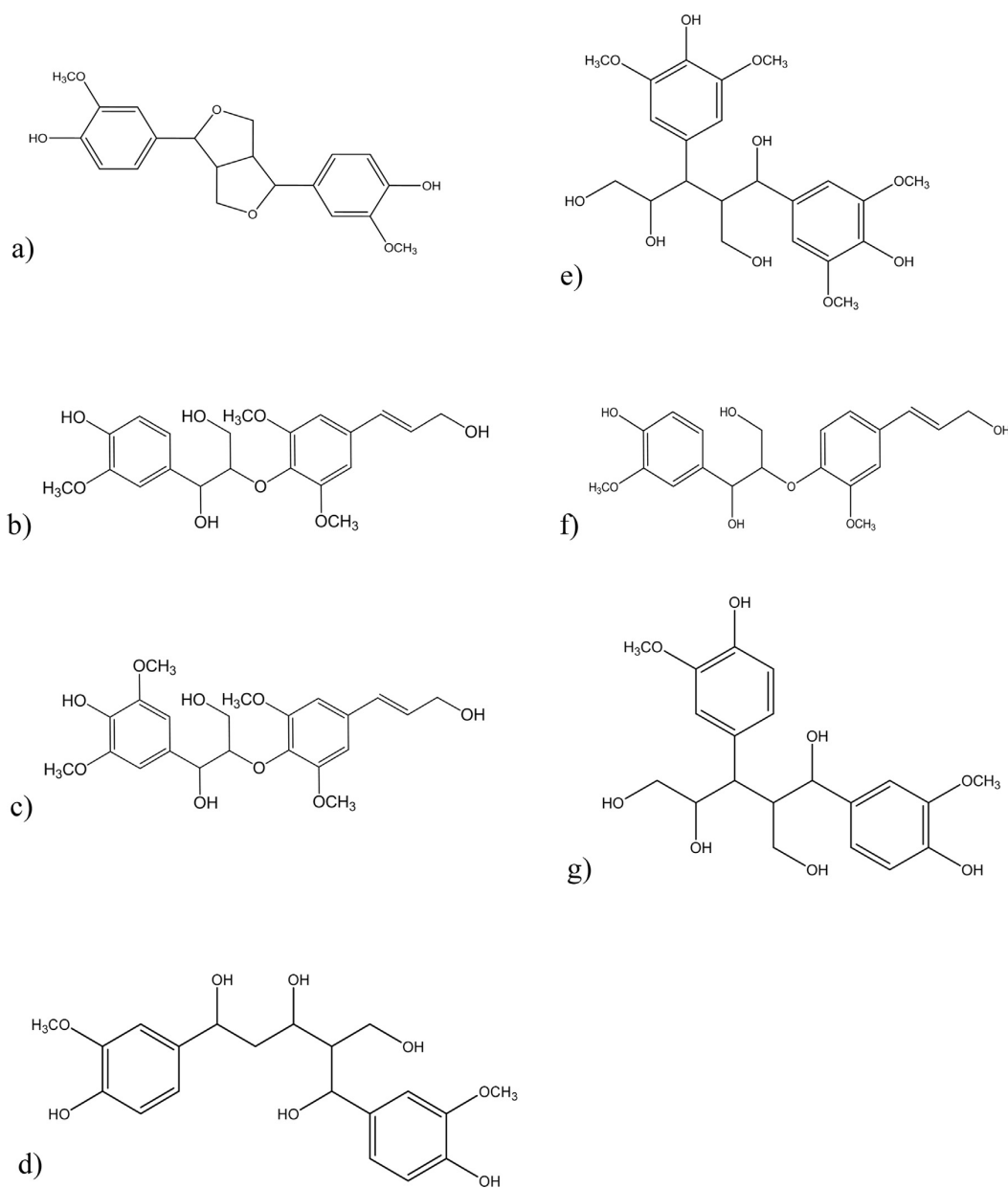


Fig. 3. Molecular structures of dilignols. a) CA<sub>ββ</sub>CA, b) CA<sub>βO4</sub>SA, c) SA<sub>βO4</sub>SA, d) CA<sub>γβ</sub>CA, e) SA<sub>αβ</sub>SA, f) CA<sub>βO4</sub>CA and g) CA<sub>αβ</sub>CA.

CA<sub>βO4</sub>SA<sub>ββ</sub>SA, 3) Ln<sub>CO2</sub>, a depolymerized kraft lignin, containing the following units: CA<sub>αβ</sub>CA, CA<sub>βO4</sub>SA<sub>ββ</sub>SA, SA<sub>βO4</sub>SA<sub>ββ</sub>SA, CA<sub>βO4</sub>CA, CA<sub>γβ</sub>CA, CA<sub>βO4</sub>SA<sub>β5</sub>CA, CA<sub>β5</sub>CA<sub>γβ</sub>SA and CA<sub>βO4</sub>CA<sub>β5</sub>CA, and 4) Ln<sub>Depol</sub>, a depolymerized Indian lignin from the mixture (1) containing the units: CA<sub>γβ</sub>CA, CA<sub>αβ</sub>CA, CA<sub>βO4</sub>SA and SA<sub>βO4</sub>SA oligolignols. Details about the characterization of the lignin mixtures are in reference [21].

### 3. Results and discussion

From our results it was observed that all complexes formed between oligolignols on the cellulose I-β crystallite surface have similar coupled conformations than those achieved in a previous molecular docking study. [21] Throughout each simulation, the period of time was long

enough to obtain complexes where the oligolignols yield stable complexes with the cellulose layer. Indeed, in a short simulation time (< 100 ps), the total energy of each complex arrives to one stationary state.

Curiously, after the minimization performed on the complexes, no significant conformational variations were observed on their geometries, which means that the structures are near a local energy minimum, as it is shown in Fig. 5 for the case of the components of the lignin mixture Ln<sub>India</sub>. From the optimized geometries, it is observed that phenyl rings of the dilignols, with linkage βO4, αβ, and γβ or at least one phenyl ring of the trilignols in the conformation CA<sub>m</sub>B<sub>n</sub>CA lies on one plane parallel to the cellulose surface. It is worth noting that these conformations are in good agreement with Raman microprobe studies [40–42], where there is evidence of the orientation of lignin



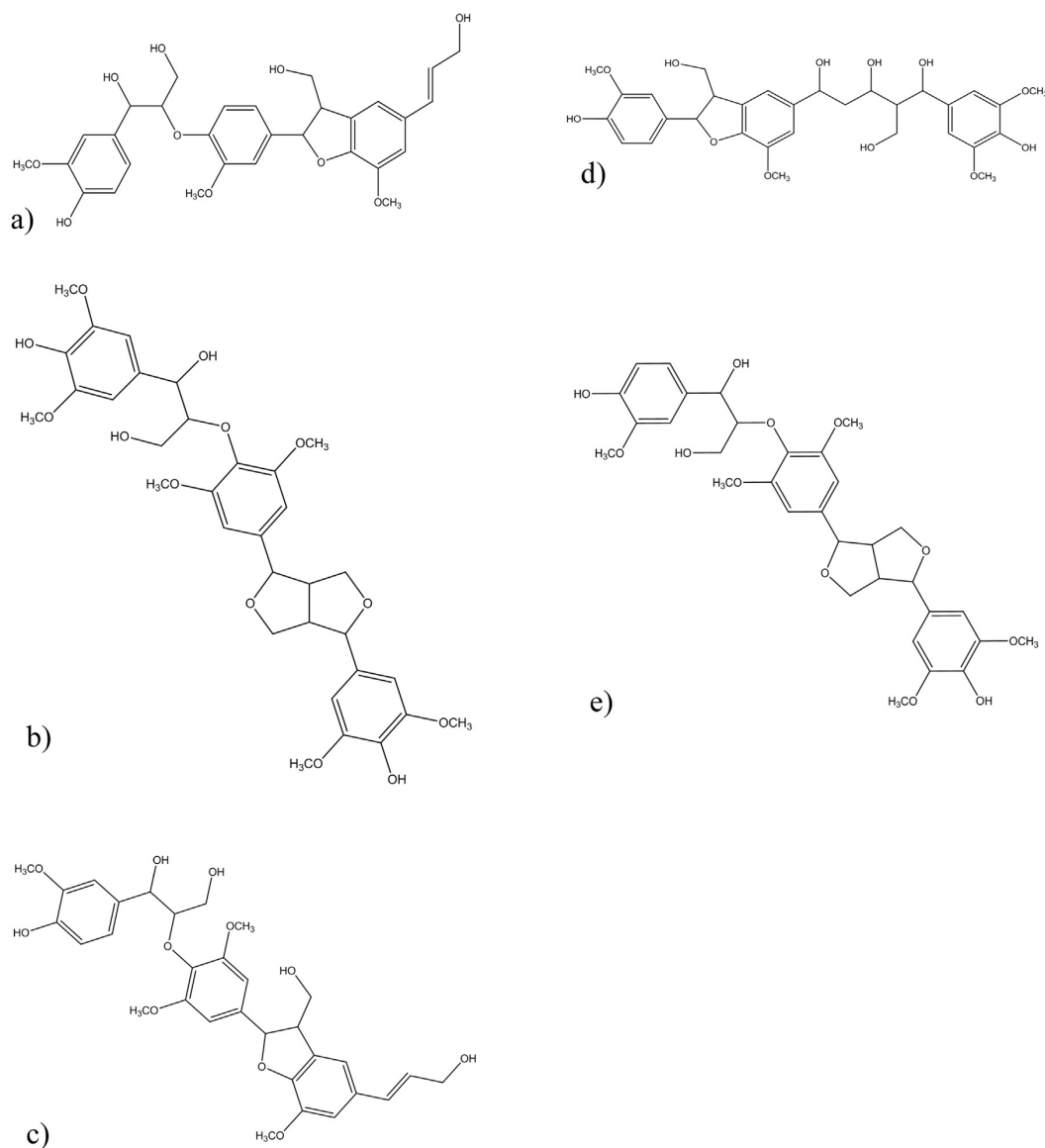


Fig. 4. Molecular structures of the trilignols: a) CA $_{\beta}O4$ CA $_{\beta}5$ CA, b) SA $_{\beta}O4$ SA $_{\beta\beta}$ SA, c) CA $_{\beta}O4$ SA $_{\beta}5$ CA, d) CA $_{\beta}5$ CA $_{\gamma\beta}$ SA, e) CA $_{\beta}O4$ SA $_{\beta\beta}$ SA.

relative to the plane of the cell wall. In most instances, the aromatic rings of the phenyl propane structural units are parallel to the plane of the cell-wall surface [42].

On the other hand, analyzing the coupling energies,  $E_C$ , of the corresponding complexes, we found that those formed between the cellulose and the dilignols with linkage  $\beta O4$  formed with units CA\_SA and SA\_SA are the most energetically stable complexes with respect to other linkages, finding the following stability ordering of complexes formed with dilignols:  $\beta O4 > \gamma\beta > \alpha\beta$ ; an exception of this trend is the CA $_{\beta}O4$ CA dilignol. For trilignols, the highest  $E_C$  of the complexes formed on cellulose corresponds to the SA $_{\beta}O4$ SA $_{\beta\beta}$ SA trilignol (see third column of Table 1). In this conformation, the phenyl ring of the trilignol from the side of the  $\beta\beta$  linkage is slightly tilted with respect to the plane of the cellulose surface. We think that, in an adhesive formulation, these kinds of complexes may provide functional groups where they are able to interact with other cellulose cells. Consequently, the corresponding formulations could be improved, taking

advantage of this conformational behavior. However, to clarify this point is necessary MD studies involving at least two layered cellulose model. Therefore, oligolignols with  $\beta O4$  and  $\beta\beta$  linkages seem to be useful substructures of lignin to form complexes energetically stable which may improve the interaction over cellulose.

We also tried to correlate the calculated coupling energies ( $E_C$ ) with the experimental  $Y_M$  from the previous study of Pizzi et al. [21]. However, such correlation can be established only if the  $E_C$  of each oligolignol is multiplied by the mass percent obtained from the experimental composition of the mixture, namely  $E_{MIX}$ ; it could be shown that  $E_{MIX}$  correlates quite well with respect to the experimental  $Y_M$ . The mass composition of the oligolignols of the four different lignins, the experimental  $Y_M$ , and the obtained  $E_{MIX}$  are summarized in Table 1.

We can observe from Table 1 a trend among the  $E_{MIX}$  results with the experimental  $Y_M$  values, pointing out which substructures of the lignin mixture contributes mainly to the adhesion. Obviously, the relationship between the experimental value of  $Y_M$  and  $E_{MIX}$  does not fit

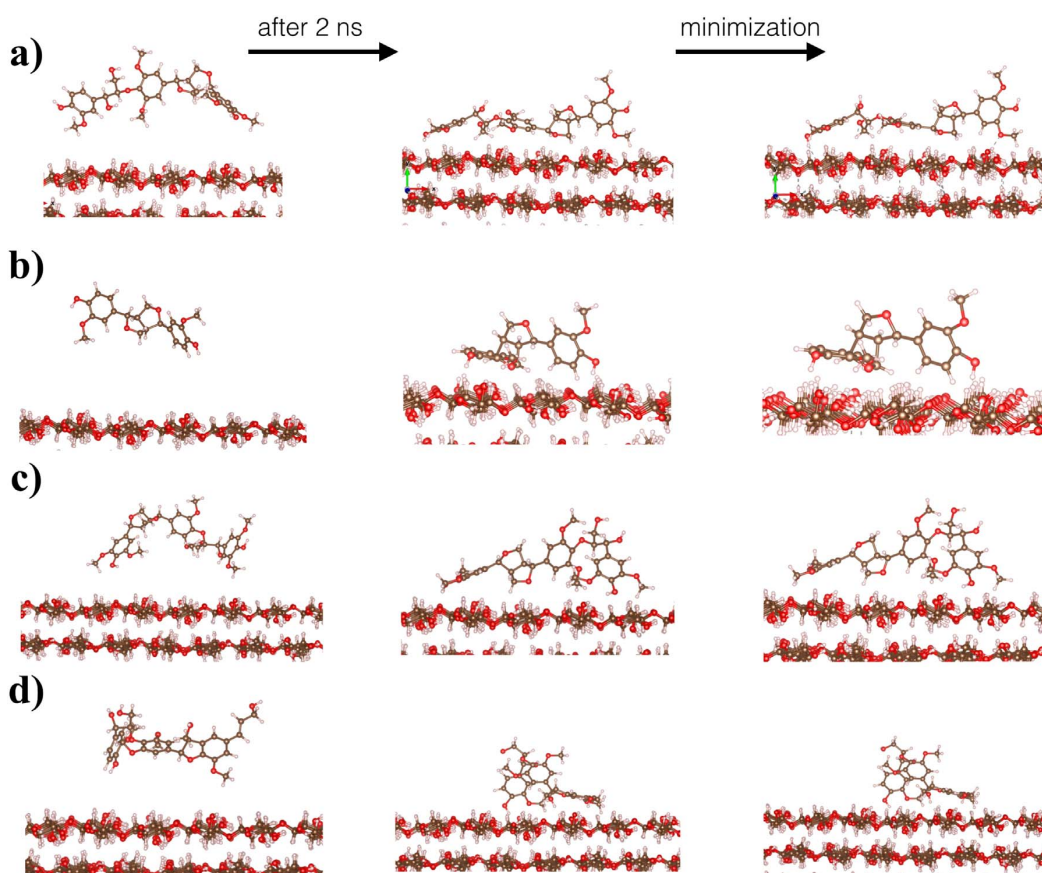


Fig. 5. Initial conformation, last snapshot after 2 ns of MD simulation and the relaxed geometries after the minimization process of the complexes formed between the cellulose I-β and (a) CA\_β04\_SA\_ββ\_SA, (b) CA\_ββ\_CA, (c) SA\_β04\_SA\_ββ\_SA and (d) CA\_β04\_SA\_β5\_CA oligolignols.

perfectly because of the sense of both magnitudes. However, when the  $E_{MIX}$  is plotted towards the experimental  $Y_M$  values, are found good linear correlations for each lignin mixture, observing individually which oligolignols maximize the adhesion in the studied samples. From Table 1 it is shown that, the oligolignols of two units with β04 and αβ linkages are the components that maximize the adhesion of the studied samples, because oligolignols of two and three units with β04 and ββ linkages. In particular, the oligolignol CA\_β04\_SA\_ββ\_SA and the C-A\_ββ\_CA one. It is evident that such information contributes strongly to improve formulations of wood adhesive materials (Fig. 6).

From the regression found in the four lignin mixtures studied in this work, we calculated the theoretical Young's modulus of the mixtures according to the follow equation:

$$\widehat{E}_C = m * Y_M$$

where  $\widehat{E}_C$  is the value of the average coupling energy ( $E_{mix} - b$ ) when the intercept “b” goes to zero, taken through the correlation of each lignin. Thus, it is possible to estimate the theoretical value of Young's modulus of each component of the studied mixtures and for each lignin mixture. The maximum deviation between the calculated and the experimental  $Y_M$  was found in the lignin Ln\_Miscanthus (~10.0%), and the best correlation was observed for the lignin Ln\_india, as shown in Fig. 7. The obtained values for theoretical  $Y_M$  for each oligolignol and their additivity are described in Supplementary material.

Without a doubt, the values for  $E_C$  found by ReaxFF have a physical sense since they correlate well with the experimental information, by means of the mass composition of the oligolignols. It is interesting to observe that our results of the  $E_{mix}$  fit mixtures of lignins up to 8

oligolignols such as the Ln\_CO<sub>2</sub> lignin. Comparing our results with those found in the literature, [13] it is possible to correlate theoretical values of the coupling energy of oligolignols on cellulose with experimental adhesion of oligolignol compositions by employing the ReaxFF.

We must bear in mind that the stability of the complexes is discussed in terms of the interaction of one molecule of oligolignol on cellulose, while the  $Y_M$  was calculated sorting the contribution of each oligolignol in the mixture; thereby it remained unclear if the mass percentage of the mixtures depends on the nature of molecular interaction or on the chemical behavior of the species, we are preparing a further work to address this point. Obviously, experimental values of  $Y_M$  depend on the total composition of the depolymerized lignins and on the cellulose contributions; however, Fig. 7 points out that the main contributions in the adhesive formulations are supported by the studied oligolignol-cellulose complexes, built accordingly the elucidated MALDI-TOF molecular structures and their enhancement from molecular interactions studies.

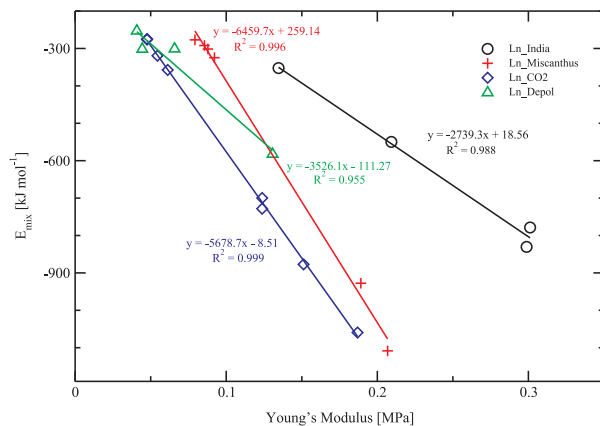
#### 4. Conclusions

By using molecular dynamics simulations, employing the ReaxFF force field, it is possible to correlate oligolignol-cellulose coupling energies with the Young's modulus of structures suggested experimentally from MALDI-TOF data, and enhanced by molecular interactions insights. This conclusion was obtained when oligolignols were designed and built with two and three units of monolignols, to interact on a model of cellulose I-β. According to the correlation obtained from molecular dynamics, the triligol CA\_β04\_SA\_ββ\_SA and the dilignol

**Table 1**  
Experimental composition (mass percent), [21] interaction energy of complexes, and measured  $Y_M$  (MPa) of lignins [21] compared with the  $Y_M$  calculated in this work.

Oligolignols in lignin mixture	Mass percent (%)	$E_C$ (kJ/mol)	$E_{MIX}$ (kJ/mol)	Experimental $Y_M$ (MPa) <sup>a</sup>
<b>Ln_India</b>				
CA_βO4_SA_ββ_SA	6.69	-11636.1	-778.5	0.3011
CA_ββ_CA	6.64	-12505.0	-830.3	0.2988
SA_βO4_SA_ββ_SA	4.65	-11824.9	-549.9	0.2093
CA_βO4_SA_β5_CA	2.99	-11787.1	-352.4	0.1346
<b>Ln_Miscanthus</b>				
CA_βO4_SA	6.21	-17849.6	-1108.5	0.2068
SA_βO4_SA	5.68	-16331.0	-927.6	0.1891
CA_γβ_CA	2.77	-11720.6	-324.7	0.0922
SA_αβ_SA	2.64	-11421.3	-301.5	0.0879
CA_βO4_CA	2.57	-11361.0	-292.0	0.0856
CA_βO4_SA_ββ_SA	2.38	-11636.1	-276.9	0.0793
<b>Ln_CO2</b>				
CA_αβ_CA	9.3	-11391.8	-1059.4	0.1869
CA_βO4_SA_ββ_SA	7.52	-11663.9	-877.1	0.1512
SA_βO4_SA_ββ_SA	6.16	-11824.9	-728.4	0.1238
CA_βO4_CA	6.16	-11361.0	-699.8	0.1238
CA_γβ_CA	3.05	-11720.6	-357.5	0.0613
CA_βO4_SA_β5_CA	2.71	-11787.1	-319.4	0.0545
CA_β5_CA_γβ_SA	2.37	-11583.7	-274.5	0.0476
CA_βO4_CA_β5_CA	2.37	-11647.5	-276.0	0.0476
<b>Ln_Depol</b>				
CA_γβ_CA	4.97	-11720.6	-582.5	0.1307
CA_αβ_CA	2.5	-12043.2	-301.1	0.0658
CA_βO4_SA	1.69	-17849.6	-301.7	0.0444
SA_βO4_SA	1.55	-16331.0	-253.1	0.0408

<sup>a</sup> Values for the experimental  $Y_M$  used to develop a correlation with  $E_M$ . They were taken from our previous MALDI-TOF study applying the weighted distribution of oligolignols developed by Pizzi et al. [21].

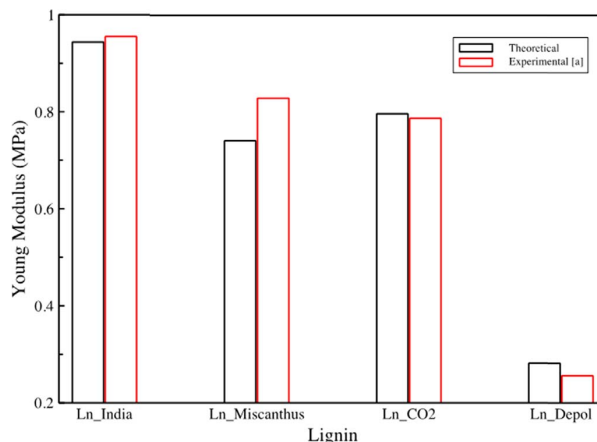


**Fig. 6.** Energetic contribution of oligolignols in each mixture ( $E_{MIX}$ ) vs the experimental  $Y_M$  of the four different types of lignins studied in this work.

CA\_ββ\_CA were the lignin substructures that contribute mostly to adhesion; in such way that, it can be described how oligolignols within lignin-adhesive formulation drive their Young's modulus from an atomistic point of view. Our results and methodology may improve the selection of the lignin source in base of its composition, after depolymerization, for wood lignin-adhesive formulations because it was observed which oligolignols promotes adhesion in wood adhesive formulations.

**Acknowledgements**

Authors thanks to CONACyT for financial support under grant



**Fig. 7.** Calculated Young's modulus with respect to the experimental value of the four lignins studied in this work.

numbers 252239, 194912 and 153125 (Fondo SEP-CONACyT).

**Appendix A. Supplementary material**

Supplementary data associated with this article can be found in the online version at <http://dx.doi.org/10.1016/j.ijadhadh.2017.08.003>.

**References**

- [1] Pizzi A. Advanced wood adhesives technology. CRC Press; 1994.
- [2] Houtman CJ, Atalla RH. Cellulose-lignin interactions (A computational study). *Plant Physiol* 1995;107:977–84.
- [3] Nimz H. Lignin-based wood adhesives. In: Pizzi A, editor. Wood adhesion chemistry and technology. New York: Dekker; 1983. p. 247–88.
- [4] Pizzi A. Recent developments in eco-efficient bio-based adhesives for wood bonding: opportunities and issues. *J Adhes Sci Technol* 2006;20:829–46.
- [5] Imam SH, Gordon SH, Mao L, Chen L. Environmentally friendly wood adhesive from a renewable plant polymer: characteristics and optimization. *Polym Degrad Stab* 2001;73:529–33.
- [6] Desai SD, Patel JV, Sinha VK. Polyurethane adhesive system from biomaterial-based polyol for bonding wood. *Int J Adhes Adhes* 2003;23:393–9.
- [7] Li K, Geng X, Simonsen J, Karchesy J. Novel wood adhesives from condensed tannins and polyethylenimine. *Int J Adhes Adhes* 2004;24:327–33.
- [8] Kim S. Environment-friendly adhesives for surface bonding of wood-based flooring using natural tannin to reduce formaldehyde and TVOC emission. *Bioresour Technol* 2009;100:744–8.
- [9] Passoni V, Scarica C, Levi M, Turri S, Griffini G. Fractionation of industrial softwood kraft lignin: solvent selection as a tool for tailored material properties. *ACS Sustain Chem Eng* 2016;4:2232–42.
- [10] Fengel D, Wegener G. Wood: chemistry, ultrastructure, reactions. Walter de Gruyter; 1983.
- [11] Pizzi A, Mittal KL. Wood adhesives. CRC Press; 2011.
- [12] Jin Y, Cheng X, Zheng Z. Preparation and characterization of phenol-formaldehyde adhesives modified with enzymatic hydrolysis lignin. *Bioresour Technol* 2010;101:2046–8.
- [13] Pizzi A. Bioadhesives for wood and fibres. *Rev Adhes Adhes* 2013;1:88–113.
- [14] Zhao M, Jing J, Zhu Y, Yang X, Wang X, Wang Z. Preparation and performance of lignin-phenol-formaldehyde adhesives. *Int J Adhes Adhes* 2016;64:163–7.
- [15] Mansouri H, Navarrete P, Pizzi A, Tapin-Lingua S, Benjelloun-Mlayah B, Pasch H, et al. Synthetic-resin-free wood panel adhesives from mixed low molecular mass lignin and tannin. *Eur J Wood Wood Prod* 2011;69:221–9.
- [16] Navarrete P, Mansouri H, Pizzi A, Tapin-Lingua S, Benjelloun-Mlayah B, Pasch H, et al. Wood panel adhesives from low molecular mass lignin and tannin without synthetic resins. *J Adhes Sci Technol* 2010;24:1597–610.
- [17] Pradyawong S, Qi G, Li N, Sun XS, Wang D. Adhesion properties of soy protein adhesives enhanced by biomass lignin. *Int J Adhes Adhes* 2017;75:66–73.
- [18] Freudenberg K. Lignin: its constitution and formation from p-hydroxycinnamyl alcohols. *Science* 1965;148:595–600.
- [19] Higuchi T. Biosynthesis and biodegradation of wood components. Elsevier; 2012.
- [20] Sjöström E, Alen R. Analytical methods in wood chemistry, Translated by Mir shokravi Ahmad and Sadeghifar H. Tehran. Payame Noor University; 1999.
- [21] Sedano-Mendoza M, Lopez-Albarran P, Pizzi A. Natural lignans as adhesives for cellulose: computational interaction energy vs experimental results. *J Adhes Sci Technol* 2010;24:1769–86.
- [22] Huey R, Morris GM, Olson AJ, Goodsell DS. A semiempirical free energy force field with charge-based desolvation. *J Comput Chem* 2007;28:1145–52.



- [23] Martínez C, Sedano M, Munro A, Lopez P, Pizzi A. Evaluation of some synthetic oligolignols as adhesives: a molecular docking study. *J Adhes Sci Technol* 2010;24:1739–51.
- [24] Nishiyama Y, Langan P, Chanzy H. Crystal structure and hydrogen-bonding system in cellulose I $\beta$  from synchrotron X-ray and neutron fiber diffraction. *J Am Chem Soc* 2002;124:9074–82.
- [25] Nishiyama Y, Sugiyama J, Chanzy H, Langan P. Crystal structure and hydrogen bonding system in cellulose I $\alpha$  from synchrotron X-ray and neutron fiber diffraction. *J Am Chem Soc* 2003;125:14300–6.
- [26] Ultra C. , 11.0. 1, CambridgeSoft Corporation, Cambridge; 2007.
- [27] Van Duin AC, Dasgupta S, Lorant F, Goddard WA. ReaxFF: a reactive force field for hydrocarbons. *J Phys Chem A* 2001;105:9396–409.
- [28] Plimpton S. Fast parallel algorithms for short-range molecular dynamics. *J Comput Phys* 1995;117:1–19.
- [29] Chenoweth K, Van Duin AC, Goddard WA. ReaxFF reactive force field for molecular dynamics simulations of hydrocarbon oxidation. *J Phys Chem A* 2008;112:1040–53.
- [30] Rappé AK, Casewit CJ, Colwell K, Goddard Iii W, Skiff W. UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *J Am Chem Soc* 1992;114:10024–35.
- [31] Mayo SL, Olafson BD, Goddard WA. DREIDING: a generic force field for molecular simulations. *J Phys Chem* 1990;94:8897–909.
- [32] Rahman R, Foster J, Haque A. Molecular dynamics simulation and characterization of graphene–cellulose nanocomposites. *J Phys Chem A* 2013;117:5344–53.
- [33] Warshel A, Levitt M. Theoretical studies of enzymic reactions: dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *J Mol Biol* 1976;103:227–49.
- [34] Chen B, Wei X-Y, Yang Z-S, Liu C, Fan X, Qing Y, et al. ReaxFF reactive force field for molecular dynamics simulations of lignite depolymerization in supercritical methanol with lignite-related model compounds. *Energy Fuels* 2012;26:984–9.
- [35] Zhang T, Li X, Qiao X, Zheng M, Guo L, Song W, et al. Initial mechanisms for an overall behavior of lignin pyrolysis through large-scale ReaxFF molecular dynamics simulations. *Energy Fuels* 2016;30:3140–50.
- [36] Beste A. ReaxFF study of the oxidation of softwood lignin in view of carbon fiber production. *Energy Fuels* 2014;28:7007–13.
- [37] Beste A. ReaxFF study of the oxidation of lignin model compounds for the most common linkages in softwood in view of carbon fiber production. *J Phys Chem A* 2014;118:803–14.
- [38] Zhang XM, Shi SQ, Cao J. Elastic properties of cellulose by molecular dynamics simulation. *Appl Mech Mater* 2013;1726–30.
- [39] Odegard GM, Jensen BD, Gowtham S, Wu J, He J, Zhang Z. Predicting mechanical response of crosslinked epoxy using ReaxFF. *Chem Phys Lett* 2014;591:175–8.
- [40] Agarwal U, Atalla R. In-situ Raman microprobe studies of plant cell walls: macromolecular organization and compositional variability in the secondary wall of *Picea mariana* (Mill.) BSP. *Planta* 1986;169:325–32.
- [41] Agarwal UP. Raman imaging to investigate ultrastructure and composition of plant cell walls: distribution of lignin and cellulose in black spruce wood (*Picea mariana*). *Planta* 2006;224:1141.
- [42] Atalla RH, Agarwal UP. Raman microprobe evidence for lignin orientation in the cell walls of native woody tissue. *Science* 1985;227:636–8.


Received: 27 February 2018 | Revised: 30 April 2018 | Accepted: 7 May 2018

DOI: 10.1002/qua.25671

WILEY  INTERNATIONAL JOURNAL OF  
QUANTUM CHEMISTRY

## REVIEW

# GPUs as boosters to analyze scalar and vector fields in quantum chemistry

Raymundo Hernández-Esparza<sup>1</sup> | Álvaro Vázquez-Mayagoitia<sup>2</sup> |  
Luis-Antonio Soriano-Agueda<sup>1</sup> | Rubicelia Vargas<sup>1</sup> | Jorge Garza<sup>1</sup> <sup>1</sup>División de Ciencias Básicas e Ingeniería,  
Departamento de Química, Universidad  
Autónoma Metropolitana-Iztapalapa, México  
City, Mexico<sup>2</sup>Computational Sciences Division, Argonne  
National Laboratory, Lemont, Illinois**Correspondence**Jorge Garza, División de Ciencias Básicas e  
Ingeniería, Departamento de Química,  
Universidad Autónoma Metropolitana-  
Iztapalapa, San Rafael Atlixco 186, Col.  
Vicentina, Iztapalapa, C. P. 09340, México  
City, MX.

Email: jgo@xanum.uam.mx

**Funding information**Consejo Nacional de Ciencia y Tecnología,  
Grant/Award Number: 155070; CONACYT,  
Grant/Award Numbers: 265471, 283251;  
Office of Science, Grant/Award Number: DE-  
AC02-06CH11357**Abstract**

The analysis of scalar and vector fields in quantum chemistry is an essential task for the computational chemistry community, where such quantities must be evaluated rapidly to perform a particular study. For example, the atoms in molecules approach proposed by Bader has become popular; however, this method demands significant computational resources to compute the involved tasks in short times. In this article, we discuss the importance of graphics processing units (GPU) to analyze electron density, and related fields, implementing several scalar, and vector fields within the *graphics processing units for atoms and molecules* (GPUAM) code developed by a group of the Universidad Autónoma Metropolitana in México City. With this application, the quantum chemistry community can perform demanding computational tasks on a desktop, where CPUs and GPUs are used to their maximum capabilities. The performance of GPUAM is tested in several systems and over different GPUs, where a GPU installed in a workstation converts it to a robust high-performance computing system.

**KEYWORDS**

GPUs, Hartree-Fock, Kohn-sham, large systems, QTAIM, semiempirical methods, visualization, wave-function analysis

## 1 | INTRODUCTION

In these days, wave-function techniques and density functional theory are cornerstones of the quantum chemistry to study the electronic structure of atoms and molecules. However, for large systems, both of these approaches demand substantial computational resources, many times unreachable with personal computers. By the side of wave-function techniques, the Hartree-Fock (HF) theory is relevant since some sophisticated methods use it as a starting point.<sup>[1]</sup> Without a doubt, the HF method was essential to establish the foundations of first principles computational chemistry, showing as its initial stages the need for parallel programming techniques to reduce overall the time to solution. Indeed, Density Functional Theory Kohn-Sham (DFT KS), which implicitly adds exchange and correlation effects and uses as a fundamental variable the electron density,<sup>[2]</sup> uses many algorithms proposed originally by HF implementations.<sup>[3]</sup> In the last three decades, the number of publications with molecular simulations has ramped up geometrically, which coincides with the release of efficient DFT KS, HF, and semiempirical parallel codes. In quantum chemistry, orbitals usually are represented in a basis set, as a linear combination of atomic localized functions like Slater-type orbitals (STO) or Gaussian-type orbitals (GTO).<sup>[4]</sup> Many modern codes that use STOs and GTOs were designed, as its conception, or modified to be executed in parallel.<sup>[5-12]</sup>

In the last years, the design of microprocessors has reached limits in miniaturization and speed, which is a consequence of the intrinsic nature of their components. On the other hand, to get faster computers adding accelerators and increasing the number of core computing units has been the most viable alternative. Accelerators like graphics processing units (GPU) have gained popularity for scientific computing. The origin of such GPUs comes from video cards designed to enhance the performance while displaying video games. GPU industry has impacted some science fields amazingly, and quantum chemistry is one of them since GPUs act as parallel computers. The availability of APIs developed by NVIDIA and Khronos Group through CUDA<sup>[13]</sup> and OpenCL,<sup>[14]</sup> respectively, has been the key to the GPUs success to gain adepts in general purpose programming. Thus, with these tools, number crunching codes can take advantage of heterogeneous architectures. Currently, some quantum chemistry

codes use GPUs to accelerate the self-consistent process involved in the solution of KS or HF methods.<sup>[5,7,8,15-18]</sup> Furthermore, there are several efforts to increase the performance of post-HF algorithms.<sup>[19-25]</sup> There are also efforts to use GPUs for nonlocalized functions in the basis set, although for this case such efforts have been devoted mainly to the solution of KS equations with nonlocal or semilocal exchange-correlation functionals, in such a way that the performance of these codes shows significant improvements.<sup>[26-29]</sup>

In summary, there are several reports demonstrating that the wave-function or electron density can be obtained efficiently through heterogeneous computing for large systems. However, the analysis or postprocessing of the electron density of related properties can be demanding tasks. An example of such computational effort is the study of the acetylcholine receptor 2 K58 proteins with 590 atoms and 2158 electrons using the 6-31G basis set, where we want to analyze either its electron density or the corresponding electrostatic potential to find some reactive sites. However, many times these properties are not enough to examine this system and additional quantum chemistry fields are needed to obtain well-supported conclusions. Thus, for this analysis, heterogeneous hardware must be used appropriately to achieve responses in reasonable times, in particular, if we want to perform such tasks on a desktop. The aim of this article is precisely the evaluation of several quantum chemistry scalar and vector fields by using CPUs and GPUs.

## 2 | EVALUATION OF QUANTUM CHEMISTRY FIELDS BY USING GPUS

In molecular simulations, the evaluation of a scalar field is a routinary activity for the visualization of the density and related properties. This evaluation is carried out over a set of points in a grid defined in real space, and after it, visualization programs allow an analysis of the corresponding field. Let us think in the evaluation of the electron density, as

$$\begin{aligned} \rho(\mathbf{r}) &= \sum_{i=1}^T \omega_i \psi_i^*(\mathbf{r}) \psi_i(\mathbf{r}), \\ &= \sum_{i=1}^T \omega_i \left( \sum_{\mu=1}^T c_{\mu,i} \phi_{\mu}(\mathbf{r}) \right)^2, \end{aligned} \quad (1)$$

$$= \sum_{\mu,\nu=1}^T \sum_{i=1}^T \omega_i c_{\mu,i}^* c_{\nu,i} \phi_{\mu}^*(\mathbf{r}) \phi_{\nu}(\mathbf{r}). \quad (2)$$

In this equation,  $\psi_i(\mathbf{r})$  represents the orbital  $i$  and  $\omega_i$  its occupancy. In this case, the basis set is built with  $\phi_{\mu}(\mathbf{r})$  functions and  $T$  represents the number of these functions. The coefficients  $c_{\mu,i}$  are those that minimize HF or KS energy, or they are obtained by diagonalizing the first-order density matrix for correlated methods.

Equations (1) and (2) provide two ways to evaluate the electron density at the point  $\mathbf{r}$ , although both of them give same results. However, Equation (1) needs fewer operations and therefore this equation must be in a code. It means, that Equation (1) involves  $2(T+1) \times \text{occ} - 1$  float point operations, where  $\text{occ}$  represents the number of occupied orbitals. However, this number of operations corresponds to each point in the grid, if there are  $\text{pts}$  points then the total number of operations is around  $2T \times \text{occ} \times \text{pts}$  to evaluate the electron density over the whole grid. Evidently, we have not taken into account the operations involved to evaluate each function,  $\phi_{\mu}$ , in the basis set. From here it is evident that the evaluation of this quantum chemistry field is an example of embarrassingly parallel workload as the computation of this property at  $\mathbf{r}$  is entirely independent of other point in the grid. Therefore, this kind of work is an excellent candidate to implement over GPUs.

Implementation of algorithms on GPUs requires some topics that a developer must bear in mind, in this article we describe some of them. In particular, the access to physical memory is a crucial issue in the implementation of codes, as the cores installed in a GPU cannot access the volatile memory provided by a CPU. In initial stages of general purpose GPUs, the RAM attached to this device was limited to few hundreds of megabytes, and consequently, the programmer had severe restrictions on the use of GPUs for high performance computing. Fortunately, at the time of this article, NVIDIA offers GPUs with several gigabytes and high rate data transfers. The work distribution is the key to use a GPU as these devices have available thousands of threads to be accessed. In practical terms, the GPU has two kinds of RAM; shared and global. Whereas shared memory gives access to few threads collected in a block, global memory provides access to all threads in the GPU. The main difference between these two types of memory stems on the rate to be accessed; shared memory is quite small but with small latency. Global memory is as big as the physical memory of the GPU, although it exhibits a high latency. There are additional issues to be considered in the use of a GPU; however, the points mentioned in this paragraph are enough to the evaluation of a quantum chemistry field by using this device.

In our laboratory, we have developed the *Graphics Processing Units for Atoms and Molecules* (GPUAM) code to evaluate several scalar and vector quantum chemistry fields coupled with the Atoms in Molecules (AIM) analysis.<sup>[30]</sup> It is worth to mention that there are several codes with some of these capabilities.<sup>[31-36]</sup> However, GPUAM has been designed to use heterogeneous computational resources. The design of this code is simple since for the evaluation of a field we associate one thread of a GPU with one point in the grid. This approach requires that all threads have access to the coefficients,  $c_{\mu,i}$ , of Equation (1) and the work to evaluate the corresponding field (Equation (1) for the electron density) is performed on each thread. Therefore, the bigger the number of threads, the faster the evaluation of the field. In this article, the evaluation of several quantum chemistry fields is the central aim, however, in initial stages of this project the critical points searching of the electron density was the primary target, which has been shown by Hernández-Esparza et al.<sup>[30]</sup> and consequently this is not discussed here.

For the last version of GPUAM, NWChem, G09, GAMESS, or MOPAC<sup>[5-7,37]</sup> give the coefficients  $c_{\mu,i}$  of Equation (1), and the Molden2AIM code also can be used to obtain WFN or WFX files.<sup>[38]</sup> Evidently, there is one step for the data arrangement since GPUAM requires of nuclei coordinates, exponents, coefficients and type of basis set because MOPAC uses STOs instead of GTOs. In this sense, GPUAM deals with STOs as well as GTOs. After the load of the corresponding data in the CPU, GPUAM sends this information to the GPU and keep it within this device as much as possible along of the execution. When GPUAM finish with the evaluation of the corresponding field, it moves the buffer to the CPU, and finally, it stores the information in a disk file. We sketch this procedure in Figure 1.

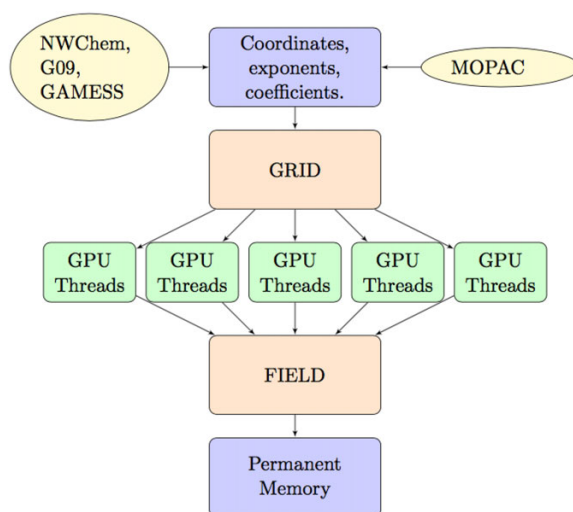
GPUAM uses a grid with a constant step over the three Cartesian axes, and the total points of this grid define the work distribution of our code, which is a crucial step. For this task, GPUAM distributes the points in the grid over the available GPUs in the system, where a CPU thread controls a GPU. It is worth to note that GPUAM does not build a buffer for the grid since with the initial point  $(x_0, y_0, z_0)$  and the step defined to create the grid, GPUAM assigns one GPU thread to one grid point by using the following CUDA kernel:

```
- global_thread_x = threadIdx.x + blockIdx.x * blockDim.x;
- global_thread_y = threadIdx.y + blockIdx.y * blockDim.y;
- sign_x = global_thread_x;
- sign_y = global_thread_y/pts_z;
- sign_z = global_thread_y*pts_z;
- sign_f = global_thread_x*pts_y*pts_z + global_thread_y;
- if (sign_x > pts_x && sign_y > pts_y && sign_z > pts_z) {
- x = x0 + (float) sign_x*step;
- y = y0 + (float) sign_y*step;
- z = z0 + (float) sign_z*step;
- buffer_field[sign_f] = field(x,y,z,coordinates,exponents,coefficients);
- }
```

In the previous code `threadIdx.x`, `blockIdx.x`, and `blockDim.x` are reserved words by CUDA to identify thread, block and the dimension of this block along x, respectively, in the GPU where GPUAM is performing the execution. Analogously, there are reserved words for the y-axis. Additionally, the number of points are denoted by `pts_x`, `pts_y`, and `pts_z`. This code is used to transform a 3D grid to a 2D grid, where the threads are executed efficiently over a GPU. For the case of several GPUs, the code replicates nuclei coordinates, exponents, and coefficients on each graphics card.

The approach sketched above has been used in GPUAM to implement scalar and vector quantum chemistry fields, for GTOs, few of these fields are:

- Molecular orbitals and their derivatives,
- electron density and its derivatives,
- reduced density gradient,<sup>[39]</sup>
- noncovalent interaction index,<sup>[40]</sup>
- electrostatic potential, and
- electron localization function.<sup>[41]</sup>



**FIGURE 1** General sketch of the design of GPUAM to evaluate a quantum chemistry field

**TABLE 1** GPUs tested in this article. The first two GPUs are designed to be installed in desktops and the Tesla family for high performance computing

GPU	Release date	Memory (MB)	Threads	Clock rate (MHz)
Quadro GP100	2016	16 273	114 688	1442
Titan V	2017	12 066	163 840	1455
Tesla K20Xm	2012	5699	28 672	732
Tesla K40	2013	11 439	30 720	745
Tesla K80	2014	11 439	26 624	823
Tesla P100	2016	16 276	114 688	1328
Tesla V100	2017	16 152	163 840	1380

Some of these fields involve more operations than others. For example, the noncovalent interaction index (NCI)<sup>[40]</sup> is useful for quantum chemistry people to study weak interactions. The NCI is based on the reduced density gradient,

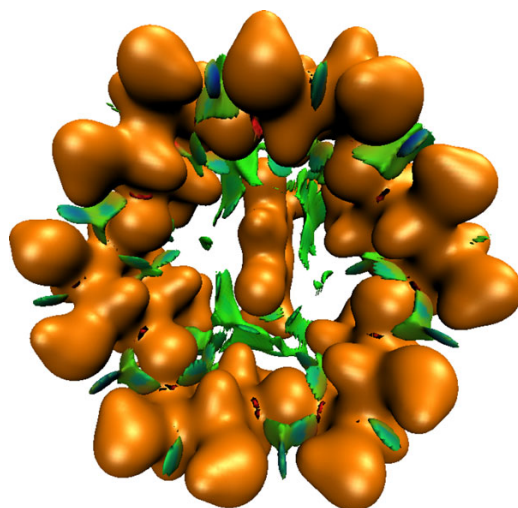
$$s(\mathbf{r}) = \frac{1}{2(3\pi^2)^{1/3}} \frac{|\nabla\rho(\mathbf{r})|}{\rho(\mathbf{r})^{4/3}}, \quad (3)$$

and on the curvature of  $\rho(\mathbf{r})$  (eigenvalues of the electron-density Hessian), it means that electron density, its first and second derivatives must be evaluated over each point of the grid. This is a good test to see the GPUs performance, although it is important to mention that the functions implemented in GPUAM for CPUs have been optimized and we think they are giving the best performance over several threads within a CPU. In our implementation, the CPU version takes advantage of the CPU architecture and the tools offered by Intel compilers.

### 3 | EXAMPLES

Continuously, the GPUAM is under scrutiny by checking its performance over different GPUs and CPUs. In this article, we present results delivered by two desktop GPUs (Quadro GP100 and Titan V) and five high-performance computing GPUs (Tesla family). Specifications of each GPU are in Table 1, where it is evident the evolution of the NVIDIA technology through the years; in particular, the number of threads in a graphics card is increased notably.

The examples discussed in this article were done with GPUAM compiled with CUDA Toolkit 8.0 or 9.0 and Intel compilers V16-V18. The first system to be analyzed by GPUAM is presented in Figure 2, where  $\rho(\mathbf{r})$ ,  $\nabla\rho(\mathbf{r})$ , and the Hessian of the electron density were evaluated to compute the NCI. In this example, an  $\alpha$ -cyclodextrin inclusion compound contains 146 nuclei, 301 orbitals, and 2680 primitive functions to represent Kohn-Sham orbitals. For this case, a methylthymine derivative (5-Methyl-1-(2-propyn-1-yl)-2,4(1H,3H)-pyrimidinedione) is included within the  $\alpha$ -cyclodextrin. Electron density and NCI depicted in Figure 2 were evaluated over a grid of 3 873 096 points and visualized with the VMD code.<sup>[42]</sup> The number of points corresponds to 5 points per atomic unit along each axis, and this is a parameter that is defined by the user.

**FIGURE 2** Electron density (dark orange) and noncovalent interaction index (mainly green and some parts blue) for an  $\alpha$ -cyclodextrin inclusion compound with 146 nuclei, 301 orbitals, and 2680 primitive functions over a grid of 3 873 096 points

**TABLE 2** Time, in seconds, used to evaluate four quantum chemistry fields for an  $\alpha$ -cyclodextrin inclusion compound with 301 orbitals, 2680 primitives, and 3 873 096 points in the grid

GPU	$\rho(r)$	$ \nabla\rho(r) $	NCI	ELF
Quadro GP100	34	37	96	37
Titan V	13	14	50	14
Tesla K20	184	203	617	201
Tesla K40	170	187	568	185
Tesla K80	112	126	340	124
Tesla P100	41	44	113	43
Tesla V100	16	16	47	16

The time used for GPUs to evaluate  $\rho(r)$ ,  $|\nabla\rho(r)|$ , and the NCI is reported in Table 2. For the NCI evaluation, GPUAM does not compute  $\rho(r)$ ,  $|\nabla\rho(r)|$ , and Hessian in separate processes, instead of this approach, GPUAM evaluates on the fly these three fields on each point of the grid. Evidently, some parts of the code are reused to avoid duplicate work.

The electron localization function, ELF,<sup>[41]</sup> is another example where GPUs show their convenience to analyze quantum chemistry fields. The definition of this scalar field is

$$\text{ELF} = \frac{1}{1 + \chi_{\sigma}^2}, \quad (4)$$

with

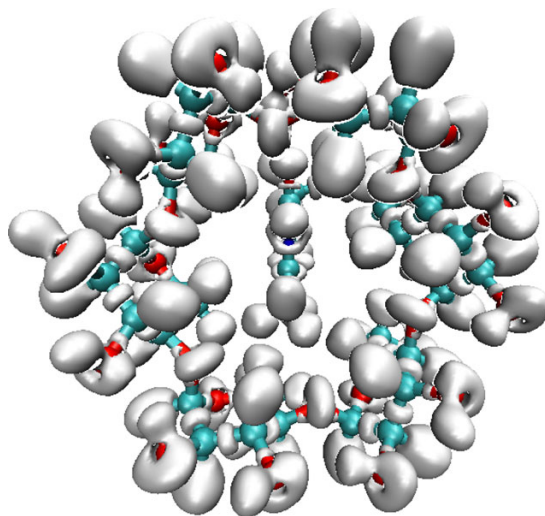
$$\chi_{\sigma} = \frac{D_{\sigma}}{D_{\sigma}^0}, \quad (5)$$

$$D_{\sigma}^0 = \frac{3}{5} (6\pi^2)^{2/3} \rho_{\sigma}^{5/3}, \quad (6)$$

and

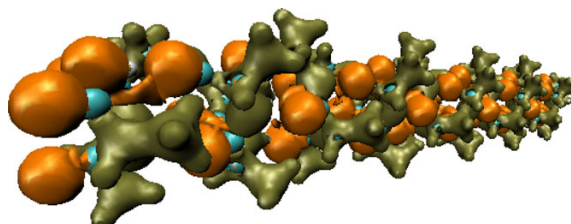
$$D_{\sigma} = \sum_{i=1}^{\text{occ}_{\sigma}} |\nabla\psi_i|^2 - \frac{1}{4} \frac{|\nabla\rho_{\sigma}|^2}{\rho_{\sigma}}, \quad (7)$$

where  $\sigma$  denotes the spin component. For this case, the ELF needs derivatives of all orbitals over each point in the grid. This scalar field for an  $\alpha$ -cyclodextrin inclusion compound is presented in Figure 3, and the time consumed to evaluate this field is reported in Table 2, where it is clear that the computational effort to compute this quantity is quite similar to that required to evaluate  $|\nabla\rho(r)|$ . From this table, it is clear how the NVIDIA technology has increased its capabilities since last generations GPUs, like Pascal (Quadro GP100, Tesla P100) and Volta (Titan V, Tesla V100), show impressive performance, in particular when these GPUs are compared with the K20 GPU.

**FIGURE 3** Electron localization function for an  $\alpha$ -cyclodextrin inclusion compound with 146 nuclei, 301 orbitals, and 2680 primitive functions over a grid of 3 873 096 points

**TABLE 3** Time, in seconds, used to evaluate four quantum chemistry fields for an  $\alpha$ -cyclodextrin inclusion compound with 301 orbitals, 2680 primitives, and 3 873 096 points in the grid

Threads	Threads per core	$\rho(r)$	$ \nabla\rho(r) $	NCI	ELF
Xeon E5-2670 v2 at 2.50 GHz					
01	1	1766	2661	3614	2661
02	1	885	1338	1813	1338
04	1	444	670	911	670
08	1	223	336	460	336
10	1	181	272	373	273
20	2	91	136	194	137
Xeon E5-2690 v2 at 3.00 GHz					
01	1	1229	1880	2518	1858
02	1	634	965	1295	957
04	1	335	494	664	507
08	1	169	254	349	254
10	1	137	206	282	206
20	2	71	108	153	108


**FIGURE 4** Laplacian of the density for a poly(L-alanine) with 35 residues. Positive values in orange and negative values in green

GPUAM was created to be also used over CPUs, at this moment there is one version optimized for Intel compilers. The results obtained by two Xeon processors of the Ivy Bridge family are reported in Table 3. These processors have 10 Cores and 20 Threads, that means that results between 1 and 10 cores use only one thread by core and for 20 cores the hyper-threading technology is used. From here, GPUAM shows an impressive performance by using the CPU since when the number of threads is increased, the efficiency measured as  $(1 \text{ thread time})/(\text{number of threads} \times \text{total time})$  is good. The lowest value of this quantity (0.82) is obtained when the NCI is evaluated with 20 threads of the Xeon E5-2690 v2 processor. From this table, if 20 cores of the Ivy Bridge family are used to execute GPUAM, then the time used by CPUs is better than that obtained by a K80 GPU. In particular, the NCI evaluation is quite good over the CPU.

The GPUAM powerful has been shown above to evaluate quantum chemistry fields when Gaussian functions are used to represent KS or HF orbitals. Besides, GPUAM supports STOs, and with it gives support to semiempirical methods, in particular, GPUAM uses MGF or AUX files from MOPAC 2016.<sup>[37]</sup> Some of the fields evaluated by GPUAM for the semiempirical method are:

- Molecular orbitals,
- electron density,
- Laplacian of the electron density,

**TABLE 4** Time, in seconds, used to evaluate four quantum chemistry fields for a poly(L-alanine) with 35 residues

GPU	$\rho(r)$	$ \nabla\rho(r) $	$\nabla^2\rho(r)$
Quadro GP100	73	106	180
Titan V	23	50	106
Tesla K20	526	659	971
Tesla K40	481	612	911
Tesla K80	297	414	683
Tesla P100	67	119	209
Tesla V100	25	46	92
10 threads Xeon E5-2670 v2 at 2.50 GHz	162	295	209
10 threads Xeon E5-2690 v2 at 3.00 GHz	124	153	160

**TABLE 5** Summary of properties computed by GPUAM for each basis set type

Properties	GTO	STO
Molecular orbitals	X	X
Molecular orbitals derivatives	X	X
Electron density	X	X
Electron density derivatives	X	X
Reduced density gradient	X	X
Noncovalent interaction index	X	X
Electrostatic potential	X	-
Electron localization function	X	-
Kinetic energy density	X	-
Search of critical points of the density	X	X
Search of critical points of the Laplacian of the density	X	-

- gradient of the electron density,
- reduced density gradient, and
- noncovalent interaction index.

The main reason to implement derivatives of the electron density obtained by semiempirical methods stems from reports, where these methods evaluate the NCI<sup>[35,36]</sup> and consequently short times are needed to study large systems. Thus, GPUAM can be a useful tool for this purpose. The evaluation of quantum chemistry fields from semiempirical methods could be a challenge since the size of the corresponding system, in general, is extended over vast regions. For example, the Laplacian of the electron density of a poly(L-alanine) with 35 residues (Figure 4) is a demanding computational task. For this case, there are 353 nuclei, 494 orbitals, and 881 primitives. The structure presented in this figure is the challenge of this system since there are 6 898 940 points over the grid; this grid has the same number of points by atomic unit of the first example. For this poly(L-alanine),  $\rho$ ,  $\nabla\rho$ , and  $\nabla^2\rho$  were evaluated by using GPUs and CPUs, the time required by each system is recorded in Table 4.

When a Quadro GP100 GPU is used to compute the electron density, the time elapsed is of 73 seconds and this time is reduced drastically when Titan V is used to obtain this field. From this table, it is evident the GPU effort to evaluate properties that involve derivatives, as the cost to get the Laplacian is higher than that used to evaluate the electron density. Such deterioration is not observed for CPUs where there are no appreciable differences between the times used for gradient and Laplacian.

From Tables 2 and 4, we conclude that desktop GPUs (Quadro GP100, Titan V) exhibit a remarkable performance and convert desktops in high-performance computers. In summary, the evaluation of quantum chemistry fields demands many threads as although the clock rate of GPUs is small, when we use many of them the execution time is reduced dramatically.

## 4 | PERSPECTIVES

For the electron density analysis and related quantum chemistry fields, integration methods are relevant, and at this moment, GPUAM has no implementation of a robust method to obtain atomic properties; this is an issue with high priority in our laboratory.<sup>[43,44]</sup> With reliable integration methods, the next step for GPUAM is the implementation of interacting quantum atoms approach to obtain a decomposition of the total energy for intraatomic and interatomic components.<sup>[45,46]</sup> Also, additional critical points search algorithms will be included in our code to offer several options to improve execution times.

## 5 | CONCLUSIONS

In this article, the Graphics Processing Units for Atoms in Molecules (GPUAM) code shows the importance to use GPUs to evaluate quantum chemistry scalar and vector fields, as this task is an example of embarrassingly parallel workload. A strategy used in this code is to assign one GPU thread to one point in the grid to evaluate the corresponding field. In Table 5, there is a list of the properties obtained by GPUAM, as a summary of the capabilities offered by this code, which uses GTOs and STOs functions. The GPUAM shows excellent performance when it computes electron density and some fields related to this property for large systems. Without a doubt, the GPUs give a great opportunity to quantum chemistry community to analyze scalar or vector fields at reasonable times by using personal desktops with at least one installed GPU. At this moment GPUAM is distributed as binary by request to the corresponding author.



## ACKNOWLEDGMENTS

The authors thank the facilities provided by the Laboratorio de Supercómputo y Visualización en Paralelo at the Universidad Autónoma Metropolitana-Iztapalapa and NVIDIA Corporation, as all results related to GPUs were provided by the NVIDIA PSG Cluster. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. R. H.-E. and L.-A. S.-A thank CONACYT, México, the scholarships 283251 and 265471, respectively.

## ORCID

Jorge Garza  <http://orcid.org/0000-0003-4249-6078>

## REFERENCES

- [1] A. Szabo, N. S. Ostlund, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, Dover, New York 1996.
- [2] R. G. Parr, W. Yang, *Density-Functional Theory of Atoms and Molecules*, Oxford University Press, Oxford 1994.
- [3] B. G. Johnson, P. M. W. Gill, J. A. Pople, *J. Chem. Phys.* **1993**, *98*, 5612.
- [4] T. Helgaker, P. Jørgensen, J. Olsen, *Molecular Electronic Structure Theory*, John Wiley & Sons, LTD, Chichester 2000.
- [5] M. Valiev, E. J. Bylaska, N. Govind, K. Kowalski, T. P. Straatsma, H. J. J. Van Dam, D. Wang, J. Nieplocha, E. Apra, T. L. Windus, W. A. de Jong, *Comput. Phys. Comm.* **2010**, *181*, 1477.
- [6] M. W. Schmidt, K. K. Baldrige, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jense, S. Koseki, N. Matsunaga, K. A. Nguyen, S. J. Su, T. L. Windus, M. Dupuis, J. A. Montgomery, *J. Comput. Chem.* **1993**, *14*, 1347.
- [7] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox. Gaussian16 Revision A.03, 2016. Gaussian Inc. Wallingford CT.
- [8] I. S. Ufimtsev, T. J. Martinez, *J. Chem. Theory Comput.* **2008**, *4*, 222.
- [9] G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. van Gisbergen, J. G. Snijders, T. Ziegler, *J. Comput. Chem.* **2001**, *22*, 931.
- [10] F. Neese, *Wiley Interdiscip. Rev.-Comput. Mol. Sci.* **2012**, *2*, 73.
- [11] L. Visscher, H. J. A. Jensen, R. Bast, and T. Saue, 2017. DIRAC, a relativistic ab initio electronic structure program, Release DIRAC17. DIRAC (see <http://www.diracprogram.org>).
- [12] Y. Shao, Z. Gan, E. Epifanovsky, A. T. B. Gilbert, M. Wormit, J. Kussmann, A. W. Lange, A. Behn, J. Deng, X. Feng, D. Ghosh, M. Goldey, P. R. Horn, L. D. Jacobson, I. Kaliman, R. Z. Khaliullin, T. Ku, A. Landau, J. Liu, E. I. Proynov, Y. Min Rhee, R. M. Richard, M. A. Rohrdanz, R. P. Steele, E. J. Sundstrom, H. Lee Woodcock III, P. M. Zimmerman, D. Zuev, B. Albrecht, E. Alguire, B. Austin, G. J. O. Beran, Y. A. Bernard, E. Berquist, K. Brandhorst, K. B. Bravaya, S. T. Brown, D. Casanova, C. -M. Chang, Y. Chen, S. Hung Chien, K. D. Closser, D. L. Crittenden, M. Diederhofen, R. A. DiStasio Jr., H. Do, A. D. Dutoi, R. G. Edgar, S. Fatehi, L. Fusti-Molnar, A. Ghysels, A. Golubeva-Zadorozhnaya, J. Gomes, M. W. D. Hanson-Heine, P. H. P. Harbach, A. W. Hauser, E. G. Hohenstein, Z. C. Holden, T. -C. Jagau, H. Ji, B. Kaduk, K. Khistyayev, J. Kim, J. Kim, R. A. King, P. Klunzinger, D. Kosenkov, T. Kowalczyk, C. M. Krauter, K. Un Lao, A. D. Lauder, K. V. Lawler, S. V. Levchenko, C. Yeh Lin, F. Liu, E. Livshits, R. C. Lochan, A. Luenser, P. Manohar, S. F. Manzer, S. -P. Mao, N. Mardirossian, A. V. Marenich, S. A. Maurer, N. J. Mayhall, E. Neuscamman, C. M. Oana, R. Olivares-Amaya, D. P. O'Neill, J. A. Parkhill, T. M. Perrine, R. Peverati, A. Prociuk, D. R. Rehn, E. Rosta, N. J. Russ, S. M. Sharada, S. Sharma, D. W. Small, A. Sodt, T. Stein, D. Stuck, Y. -C. Su, A. J. W. Thom, T. Tsuchimochi, V. Vanovschi, L. Vogt, O. Vydrov, T. Wang, M. A. Watson, J. Wenzel, A. White, C. F. Williams, J. Yang, S. Yeganeh, S. R. Yost, Z.-Q. You, I. Ying Zhang, X. Zhang, Y. Zhao, B. R. Brooks, G. K. L. Chan, D. M. Chipman, C. J. Cramer, W. A. Goddard III, M. S. Gordon, W. J. Hehre, A. Klamt, H. F. Schaefer III, M. W. Schmidt, C. D. Sherrill, D. G. Truhlar, A. Warshel, X. Xu, A. Aspuru-Guzik, R. Baer, A. T. Bell, N. A. Besley, J.-D. Chai, A. Dreuw, B. D. Dunietz, T. R. Furlani, S. R. Gwaltney, C.-P. Hsu, Y. Jung, J. Kong, D. S. Lambrecht, W. Liang, C. Ochsenfeld, V. A. Rassolov, L. V. Slipchenko, J. E. Subotnik, T. Van Voorhis, J. M. Herbert, A. I. Krylov, P. M. W. Gill, M. Head-Gordon, *Mol. Phys.* **2015**, *113*, 184.
- [13] J. Sanders, E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Tsinghua University Press, Boston 2010.
- [14] M. Scarpino, *OpenCL in Action: How to Accelerate Graphics and Computation*, Manning, Shelter Island 2012.
- [15] I. S. Ufimtsev, T. J. Martinez, *J. Chem. Theory Comput.* **2009**, *5*, 3138.
- [16] A. Asadchev, M. S. Gordon, *J. Chem. Theory Comput.* **2012**, *8*, 4166.
- [17] M. Rodriguez-Bautista, C. Diaz-García, A. M. Navarrete-López, R. Vargas, J. Garza, *J. Chem. Phys.* **2015**, *143*, 034103.
- [18] M.-A. Martínez-Sánchez, M. Rodríguez-Bautista, R. Vargas, J. Garza, *Theor. Chem. Acc.* **2016**, *135*, 207.
- [19] L. Vogt, R. Olivares-Amaya, S. Kermes, Y. Shao, C. Amador-Bedolla, A. Aspuru-Guzik, *J. Phys. Chem. A* **2008**, *112*(10), 2049.
- [20] W. Ma, S. Krishnamoorthy, O. Villa, K. Kowalski, *J. Chem. Theory Comput.* **2011**, *7*, 1316.
- [21] A. Eugene, I. I. DePrince, R. Jeff, *J. Chem. Theory Comput.* **2011**, *7*, 1287.
- [22] A. Asadchev, M. S. Gordon, *J. Chem. Theory Comput.* **2013**, *9*, 3385.
- [23] A. E. DePrince, III, M. R. Kennedy, B. G. Sumpter, and C. D. Sherrill. *Mol. Phys.* **2014**, *112*, 844.
- [24] J. J. Eriksen, *Mol. Phys.* **2017**, *115*, 2086.
- [25] I. A. Kaliman, A. I. Krylov, *J. Comput. Chem.* **2017**, *38*, 842.
- [26] X. Gonze, B. Amadon, P. M. Anglade, J. M. Beuken, F. Bottin, P. Boulanger, F. Bruneval, D. Caliste, R. Caracas, M. Cote, T. Deutsch, L. Genovese, P. Ghosez, M. Giantomassi, S. Goedecker, D. R. Hamann, P. Hermet, F. Jollet, G. Jomard, S. Leroux, M. Mancini, S. Mazevet, M. J. T. Oliveira, G. Onida, Y. Pouillon, T. Rangel, G. M. Rignanese, D. Sangalli, R. Shaltaf, M. Torrent, M. J. Verstraete, G. Zerah, J. W. Zwanziger, *Comput. Phys. Commun.* **2009**, *180*, 2582.
- [27] J. Enkovaara, C. Rostgaard, J. J. Mortensen, J. Chen, M. Dulak, L. Ferrighi, J. Gavnholt, C. Glinsvad, V. Haikola, H. A. Hansen, H. H. Kristoffersen, M. Kuisma, A. H. Larsen, L. Lehtovaara, M. Ljungberg, O. Lopez-Acevedo, P. G. Moses, J. Ojanen, T. Olsen, V. Petzold, N. A. Romero, J. Stausholm-Moller, M. Strange, G. A. Tritsarlis, M. Vanin, M. Walter, B. Hammer, H. Hakkinen, G. K. H. Madsen, R. M. Nieminen, J. K. Nørskov, M. Puska, T. T. Rantala, J. Schiøtz, K. S. Thygesen, K. W. Jacobsen, *J. Phys.-Condens. Matter* **2010**, *22*, 253202.

- [28] P. Giannozzi, O. Andreussi, T. Brumme, O. Bunau, M. Buongiorno Nardelli, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, M. Cococcioni, N. Colonna, I. Carnimeo, A. Dal Corso, S. de Gironcoli, P. Delugas, R. A. DiStasio Jr., A. Ferretti, A. Floris, G. Fratesi, G. Fugallo, R. Gebauer, U. Gerstmann, F. Giustino, T. Gorni, J. Jia, M. Kawamura, H.-Y. Ko, A. Kokalj, E. Kucukbenli, M. Lazzeri, M. Marsili, N. Marzari, F. Mauri, N. L. Nguyen, H.-V. Nguyen, A. Otero-de-la Roza, L. Paulatto, S. Ponce, D. Rocca, R. Sabatini, B. Santra, M. Schlipf, A. P. Seitsonen, A. Smogunov, I. Timrov, T. Thonhauser, P. Umari, N. Vast, X. Wu, S. Baroni, *J. Phys.-Condes. Matter* **2017**, *29*, 465901
- [29] G. Kresse, J. Furthmuller, *Phys. Rev. B* **1996**, *54*, 11169.
- [30] R. Hernández-Esparza, S. M. Mejía-Chica, A. D. Zapata-Escobar, A. Guevara-García, A. Martínez-Melchor, J. M. Hernández-Pérez, R. Vargas, J. Garza, *J. Comput. Chem.* **2014**, *35*, 2272.
- [31] P. L. A. Popelier, *Comput. Phys. Commun.* **1996**, *93*, 212.
- [32] T. Lu, F. Chen, *J. Comput. Chem.* **2012**, *33*, 580.
- [33] J. M. Solano-Altamirano, J. M. Hernández-Pérez, *Comput. Phys. Commun.* **2015**, *196*, 362.
- [34] G. Hermann, V. Pohl, J. C. Tremblay, B. Paulus, H. C. Hege, A. Schild, *J. Comput. Chem.* **2016**, *37*, 1511.
- [35] G. Rubez, J.-M. Etancelin, X. Vigouroux, M. Krajecki, J.-C. Boisson, E. Henon, *J. Comput. Chem.* **2017**, *38*, 1071.
- [36] C. Barberot, J. C. Boisson, S. Gerard, H. Khartabil, E. Thiriot, G. Monard, E. Henon, *Comput. Theor. Chem.* **2014**, *1028*, 7.
- [37] Stewart Computational Chemistry, Colorado Springs, CO, USA. MOPAC. <https://OpenMOPAC.net>.
- [38] W. Zou, D. Nori-Shargh, J. E. Boggs, *J. Phys. Chem. A* **2013**, *117*, 207.
- [39] J. P. Perdew, K. Burke, Y. Wang, *Phys. Rev. B* **1996**, *54*, 16533.
- [40] J. Contreras-García, E. R. Johnson, S. Keinan, R. Chaudret, J.-P. Piquemal, D. N. Beratan, W. Yang, *J. Chem. Theory Comput.* **2011**, *7*, 625.
- [41] A. D. Becke, K. E. Edgecombe, *J. Chem. Phys.* **1990**, *92*, 5397.
- [42] W. Humphrey, A. Dalke, K. Schulten, *J. Mol. Graph.* **1996**, *14*, 33.
- [43] P. L. A. Popelier, *Theor. Chem. Acc.* **2001**, *105*, 393.
- [44] G. Henkelman, A. Arnaldsson, H. Jonsson, *Comput. Mater. Sci.* **2006**, *36*, 354.
- [45] M. A. Blanco, A. Martín Pendás, E. Francisco, *J. Chem. Theory and Comput.* **2005**, *1*, 1096.
- [46] E. Francisco, A. Martín Pendás, M. A. Blanco, *J. Chem. Theory and Comput.* **2006**, *2*, 90.

**How to cite this article:** Hernández-Esparza R, Vázquez-Mayagoitia A, Soriano-Agueda L-A, Vargas R, Garza J. GPUs as boosters to analyze scalar and vector fields in quantum chemistry. *Int J Quantum Chem.* 2018;e25671. <https://doi.org/10.1002/qua.25671>



## Analyzing ZnO clusters through the density-functional theory

Irineo-Pedro Zaragoza<sup>1</sup> · Luis-Antonio Soriano-Agueda<sup>2</sup> · Raymundo Hernández-Esparza<sup>2</sup> · Rubicelia Vargas<sup>2</sup> · Jorge Garza<sup>2</sup>

Received: 14 March 2018 / Accepted: 22 May 2018 / Published online: 16 June 2018  
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

### Abstract

The potential energy surface of  $Zn_nO_n$  clusters ( $n = 2, 4, 6, 8$ ) has been explored by using a simulated annealing method. For  $n = 2, 4$ , and  $6$ , the CCSD(T)/TZP method was used as the reference, and from here it is shown that the M06-2X/TZP method gives the lowest deviations over PBE, PBE0, B3LYP, M06, and MP2 methods. Thus, with the M06-2X method we predict isomers of  $Zn_nO_n$  clusters, which coincide with some isomers reported previously. By using the atoms in molecules analysis, possible contacts between Zn and O atoms were found for all structures studied in this article. The bond paths involved in several clusters suggest that  $Zn_nO_n$  clusters can be obtained from the zincite (ZnO crystal), such an observation was confirmed for clusters with  $n = 2 - 9, 18$  and  $20$ . The structure with  $n = 23$  was obtained by the procedure presented here, from crystal information, which could be important to confirm experimental data delivered for  $n = 18$  and  $23$ .

**Keywords** ZnO clusters · DFT · Exchange-correlation functionals · Simulated annealing

### Introduction

Clusters of ZnO, small or large, are quite interesting since they have many technological applications [1–6]. Quantum chemistry methods have been used to interpret some experimental results and to propose molecular structures. However, the study of these systems offers a big challenge because they contain many degrees of freedom and therefore the global minimum for each cluster is elusive. Even

for small  $Zn_nO_n$  clusters, the number of electrons is of considerable size, and consequently, the density functional theory (DFT) has been used in several reports as the principal tool to analyze these systems. In such reports, some variants of the exchange-correlation (XC) functional, of the Kohn–Sham (KS) method, have been used to obtain the corresponding electronic structure. After a detailed review of previous works related to the global minimum searching for these clusters, just three XC functionals have been used for this task. Two of them are based on the generalized-gradient approximation (GGA), they are PBE [7] and BLYP. [8, 9] The third one is the popular B3LYP [10, 11], which has one contribution of the exact-exchange. Localized and non-localized functions, as basis sets, have been coupled with these XC functionals. Gaussian functions and numerical representations of the KS orbitals are within the localized functions group [12–30], and plane-waves belong to the non-localized functions group [31]. For both groups of functions, sometimes effective core potentials (ECP) are used to reduce the number of functions in the basis set and to take into account relativistic effects.

Some conclusions obtained in these works are contradictory since some authors show that one-ring structures for clusters with  $n = 2 - 7$  represent global minima, [12, 14, 21, 22, 24, 31] while others present different conclusions: they mention that these structures are global minima only for  $n < 6$ . [17, 27, 32]

This paper belongs to Topical Collection International Conference on Systems and Processes in Physics, Chemistry and Biology (ICSPPCB-2018) in honor of Professor Pratim K. Chattaraj on his sixtieth birthday

**Electronic supplementary material** The online version of this article (<https://doi.org/10.1007/s00894-018-3691-y>) contains supplementary material, which is available to authorized users.

✉ Jorge Garza  
jgo@xanum.uam.mx

<sup>1</sup> División de Estudios de Posgrado e Investigación, Tecnológico Nacional de México Campus Instituto Tecnológico de Tlalnepantla, Av. Mario Colín S/N, La Comunidad, Tlalnepantla de Baz, C.P. 54070, Estado de México, México

<sup>2</sup> Departamento de Química, División de Ciencias Básicas e Ingeniería, Universidad Autónoma Metropolitana-Iztapalapa, San Rafael Atlixco 186, Col. Vicentina, Iztapalapa, C. P. 09340, Ciudad de México, México

In our best of knowledge, the first work related to the global minimum searching of these clusters was published by Matxain et al., where they used the B3LYP XC functional [12]. In the same work, the authors compare results from this XC functional with those obtained by CCSD(T)/TZ2P method for the  $Zn_2O_2$  system, which exhibits just two isomers. For  $Zn_3O_3$  they performed this comparison by using the CCSD(T)/SKBJ(d) and B3LYP/SKBJ(d) methods, where relativistic effects are incorporated through the ECP coupled with the corresponding basis set. From this work, it is clear that the B3LYP XC functional does not exhibit a good performance, although such a comparison was made to show possible differences between this XC functional and correlated methods based on the wavefunction. Precisely in this work, the authors affirm that the one-ring structure represents the global minimum for  $n < 8$ . Other reports give support to this conclusion [14, 21, 22, 24, 31] and others use these structures for additional analysis. There is other research where several XC functionals were tested for the  $Zn_{12}O_{12}$  cluster; unfortunately the authors did not search minima for this system. One conclusion made by these authors indicates that the M06-2X XC functional is not appropriate to study ZnO clusters [25]. Very recently, Chen et al. have reported minima of  $Zn_nO_n$  with  $n \leq 168$ , in their study the B3LYP was used to search the corresponding minimum for each cluster, and over some of these minima, nine additional XC functionals were tested without a geometry relaxation [30].

At this point, it is evident that there are no strong arguments to declare which XC functional offers reliable relative energies for  $Zn_nO_n$  isomers. In this way, one aim of this work is to show the performance of several XC functionals to describe the potential energy surface (PES) of  $Zn_nO_n$  clusters with  $n = 2, 4, 6, 8$ . For all these clusters, the global minimum searching was carried out by using the Kohn–Sham method, and for  $n = 2, 4, 6$  the CCSD(T)/TZP method was used as the reference for several isomers. Also, we have found a tight relationship between clusters of this oxide and its structure in a crystalline phase. For this observation,  $n$  was tested for 1–9, 18 and 20, giving for all of them the corresponding global minimum and in the same sense we propose one structure for  $Zn_{23}O_{23}$ .

## Methods

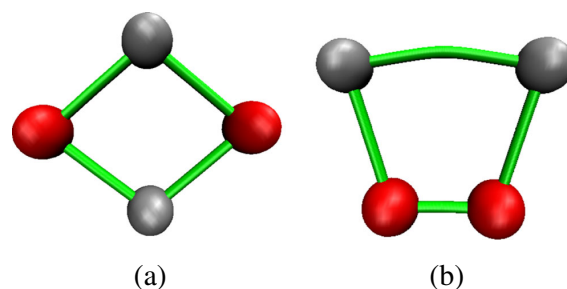
The exploration of the PES for  $Zn_nO_n$ , with  $n = 2, 4, 6$  and 8, was done by using a modified version of the Monte Carlo method through a simulated annealing process, as this is implemented in the ASCEC code [33, 34]. For this case, the objective function was provided by the total energy delivered by the PM7 method developed in the MOPAC suite code [35]. Our methodology is based on the

procedure used by García-Hernández et al. [36] to search isomers in a stochastic way. For all clusters considered in this study, the origin was the initial position of all atoms, which were moved randomly. The objective function was evaluated around 360,000 times in the stochastic process. From here, each structure obtained from the simulated annealing method was optimized by the Kohn–Sham (KS) method. Thus, all candidates proposed by ASCEC were optimized with the exchange-correlation functionals: PBE, [7] PBE0, [37] B3LYP, [10, 11] M06, [38] M06-2X, [39] as well as second-order many-body perturbation theory (MP2) [40]. All these methods were coupled with the all-electron TZP basis set [41, 42]. The CCSD(T) coupled-cluster method [43–45] was applied for  $n = 2, 4$  and 6 over MP2 and M06-2X structures. Only for  $n = 2$ , the cc-pVDZ, cc-pVTZ, and cc-pVQZ [46] basis sets were employed with M06-2X, MP2, and CCSD(T) methods. For  $n \geq 18$ , the initial optimization process was done by using the LANL2DZ basis set and its corresponding ECP [47] from here, the TZP basis set was used to obtain the final optimized structures and the corresponding frequency analysis to confirm minima over the potential energy surface. The NWChem suite code v6.6 performed all these calculations [48]. The ionization potential was obtained for each structure found by the M06-2X/TZP method. The possible bonds formed in each  $Zn_nO_n$  cluster were studied through the atoms in molecules (AIM) [49] approach by using a code developed under graphics processing units [50, 51].

## Results

### Relative energies

Before to analyze results obtained for all  $Zn_nO_n$  clusters, small clusters of  $Zn_2O_2$ ,  $Zn_4O_4$ , and  $Zn_6O_6$  are studied in detail to provide the adequate method used within the KS method to describe these systems and contrast the corresponding results with other reports. The five XC



**Fig. 1** Two isomers of the  $Zn_2O_2$  cluster. Red spheres represent oxygen atoms and gray spheres represent zinc atoms

**Table 1** Relative energy between the two isomers found for  $Zn_2O_2$ . All quantities are in kJ/mol

	PBE	PBE0	B3LYP	M06	M06-2X	MP2	CCSD(T)
Optimized Geometry	132.5	161.7	144.8	119.1	185.4	239.9	–
MP2 Geometry	150.0	–392.1	–374.1	129.7	210.9	239.9	208.1
M06-2X Geometry	131.0	161.8	144.9	120.9	185.4	250.8	187.6

functionals mentioned in the last section and the MP2 method were used to optimize the two molecular geometries exhibited by the  $Zn_2O_2$  cluster in Fig. 1. Such structures coincide with those reported by Matxain et al. [12]. For all figures presented in this article, the connectivity between atoms is provided by bond paths obtained from the AIM approach.

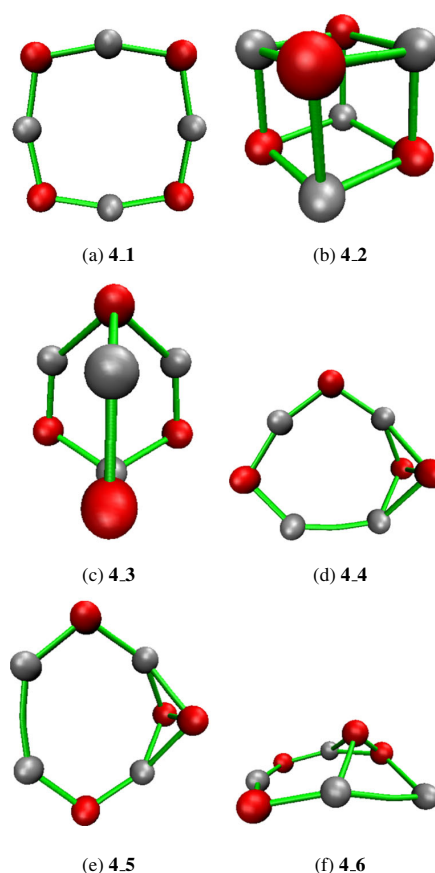
The relative energies obtained by these methods are reported in Table 1, where all methods used in this work give different predictions among themselves. For this reason, we use the CCSD(T)/TZP method as the reference. In the first instance, MP2/TZP geometries were used to estimate the correlation energy through the CCSD(T) method. The relative energy estimated for all methods, including CCSD(T), over MP2 geometries are reported in the same Table 1. From here, there are several interesting results: First, the relative energy obtained by MP2/TZP is quite different than that obtained by the five XC functionals tested in this work. In particular, when the MP2 geometry is used, PBE0 and B3LYP exhibit a different ordering than that obtained for the optimized geometries. Second, the MP2/TZP method predicts larger relative energies than those delivered by the CCSD(T)/TZP method. Finally, the M06-2X/TZP gives a relative energy quite similar to that predicted by CCSD(T). From this observation, the M06-2X geometry was used to estimate the relative energy for all methods considered in this article over these two isomers. The corresponding results from this procedure are also in Table 1, where it is evident that all XC functionals give similar results than those obtained from their corresponding molecular geometries, suggesting that the XC functionals tested in this work deliver similar geometries, at least for this cluster. Besides, it is worth noting the predictions made

by CCSD(T)/TZP and M06-2X/TZP methods; they are almost the same. Even for this structure, MP2/TZP predicts higher relative energy than CCSD(T)/TZP. Therefore, for this small cluster, the M06-2X exhibits the best performance when the reference is the CCSD(T)/TZP method.

The basis set is an important ingredient within quantum chemistry methods. There is one reference where the TZ2P was used to study  $Zn_nO_n$  clusters [12]. Currently, the high-performance computing and the software developed around this technology, calculations with high accuracy

**Table 2** Relative energy between the two isomers found for  $Zn_2O_2$ . All quantities are in kJ/mol

Basis set	M06-2X	MP2	CCSD(T)
TZP	185.4	250.8	187.6
cc-pVDZ	169.4	236.1	169.1
cc-pVTZ	185.0	260.5	194.6
cc-pVQZ	197.7	277.4	211.7


**Fig. 2** Six isomers of the  $Zn_4O_4$  cluster. Red spheres represent oxygen atoms and gray spheres represent zinc atoms

**Table 3** Relative energies ( $\Delta E_i = E_i - E_1$ ) for the isomers found for  $Zn_4O_4$  clusters. All quantities are in kJ/mol

	PBE	PBE0	B3LYP	M06	M06-2X	MP2	CCSD(T)
$\Delta E_2$	206.9	180.9	219.3	206.8	111.2	155.4	117.6 <sup>a</sup> 118.4 <sup>b</sup>
$\Delta E_3$	266.5	284.2	293.7	281.0	267.9	268.8	255.5 <sup>a</sup> 254.8 <sup>b</sup>
$\Delta E_4$	321.7	341.6	345.1	321.7	334.1	377.9	337.5 <sup>a</sup> 333.4 <sup>b</sup>
$\Delta E_5$	329.2	343.6	337.9	329.1	348.0	392.3	349.8 <sup>a</sup> 345.4 <sup>b</sup>
$\Delta E_6$	363.7	380.8	377.0	355.1	364.9	428.9	374.7 <sup>a</sup> 372.7 <sup>b</sup>
$\Delta E_7$	372.3	392.2	380.8	365.5	381.1	451.5	391.1 <sup>a</sup> 388.4 <sup>b</sup>
$\Delta E_8$	465.1	490.1	487.3	439.2	442.1	500.2	445.7 <sup>a</sup> 445.3 <sup>b</sup>
$\Delta E_9$	434.4	477.9	462.5	440.4	449.0	509.5	446.2 <sup>a</sup> 447.4 <sup>b</sup>
$\Delta E_{10}$	453.0	474.5	486.1	460.8	431.8	492.1	458.8 <sup>a</sup> 461.2 <sup>b</sup>

<sup>a</sup> M06-2X geometry. <sup>b</sup> MP2 geometry

can be performed for systems of small and medium size. However, for large systems, some methods demand high computational resources. For this reason, in this article the TZP basis set was employed to study  $Zn_nO_n$  clusters of reasonable size. For the  $Zn_2O_2$  system over the geometry predicted by the M06-2X/TZP method, three consistent-correlated basis sets were used with M06-2X, MP2, and CCSD(T) methods; their results are presented in Table 2.

From these results, some conclusions can be established for the  $Zn_2O_2$  cluster: (1) the MP2 method overestimates the relative energy concerns the CCSD(T) method; (2) for large basis sets, the M06-2X method underestimates the relative energy; (3) Virtual orbitals have an important impact on MP2 and CCSD(T) methods to predict the correlation energy; and (4) the TZP basis set predicts relative energy between the cc-pVTZ and cc-pVQZ basis sets for the

**Table 4** Relative energies ( $\Delta E_i = E_i - E_1$ ) for the isomers found for  $Zn_6O_6$  clusters. All quantities are in kJ/mol

	PBE	PBE0	B3LYP	M06	M06-2X	MP2	CCSD(T)
$\Delta E_2$	0. 78.0	0. 79.9	0. 92.1	0. 369.4	69.1 69.1	77.4 74.5	67.4
$\Delta E_3$	-42.2 -41.2	-1.1 0.5	-68.4 -66.0	-96.8 -200.5	123.8 123.8	44.3 46.6	96.2
$\Delta E_4$	34.4 29.5	66.2 66.3	19.7 20.4	-10.8 -104.2	140.6 140.6	94.6 94.1	119.7
$\Delta E_5$	88.5 86.4	122.7 122.6	101.5 101.5	36.2 -22.1	168.3 168.3	138.8 138.8	150.9
$\Delta E_6$	106.8 103.6	133.6 132.4	111.1 110.7	69.8 23.8	178.6 178.6	147.5 146.8	160.2
$\Delta E_7$	120.8 119.9	150.4 150.8	132.9 133.1	0. 31.9	185.1 185.1	161.8 161.8	170.6
$\Delta E_8$	126.9 126.6	155.3 155.8	138.3 138.6	-96.6 44.6	191.5 191.5	166.0 166.3	176.8
$\Delta E_9$	90.3 94.3	139.6 150.1	98.0 114.5	10.6 -74.4	228.1 228.1	167.4 180.	202.4
$\Delta E_{10}$	127.3 122.8	170.6 169.8	151.6 151.4	-96.6 3.5	229.7 229.7	190.8 190.4	204.3

Second row, M06-2X geometry



M06-2X method, and for MP2 or CCSD(T) methods, this basis set gives predictions between the cc-pVDZ and cc-pVTZ basis sets.

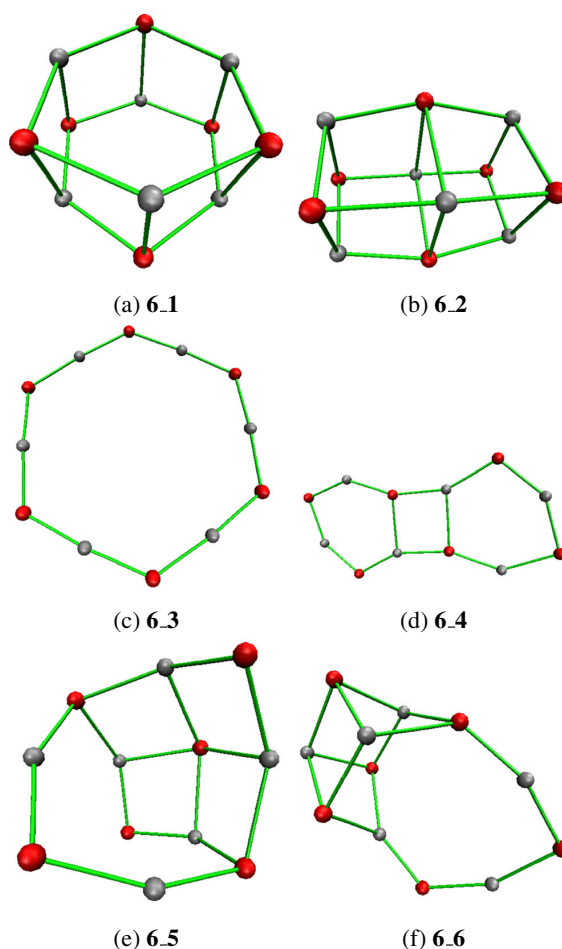
Thus, for the M06-2X/TZP method, we expect underestimations of relative energies among isomers with a similar quality to the cc-pVTZ basis set. Furthermore, for this case, the MP2 will give overestimations of these energy differences.

At this moment, one system with only two isomers has been discussed for all methods. Now, the  $Zn_4O_4$  system is analyzed in the same way. For this case, we will discuss the 11 low-lying energy isomers provided by the procedure delineated in the [Methods](#) section, which are reported in the Electronic Supplementary Information (ESI). The six most stable isomers for this cluster are presented, in ascending order, in Fig. 2. In this article, the notation  $n_j$  indicates the isomer  $j$  of the  $Zn_nO_n$  cluster.

For this cluster, our prediction coincides with other authors in the sense that the one-ring structure is the most stable isomer, and the first three isomers are the same as those obtained by force fields techniques [32].

We must bear in mind that the CCSD(T)/TZP method is the reference in this article. The relative energy obtained by each method for this cluster is reported in Table 3, where the M06-2X/TZP method gives the smallest deviation as regards to the reference. In particular, for  $\Delta E_2$  all methods, except M06-2X, overestimate this quantity. The CCSD(T)/TZP method defines the ordering presented in this table, which is not followed for several methods. For example, B3LYP breaks this ordering from  $\Delta E_4$  to  $\Delta E_5$ , PBE and PBE0 from  $\Delta E_8$  to  $\Delta E_9$ , M06-2X and MP2 from  $\Delta E_9$  to  $\Delta E_{10}$ . Curiously, the M06/TZP method gives the same ordering as that predicted by CCSD(T)/TZP, although such a method exhibits, in absolute error, large differences for low-lying relative energies.

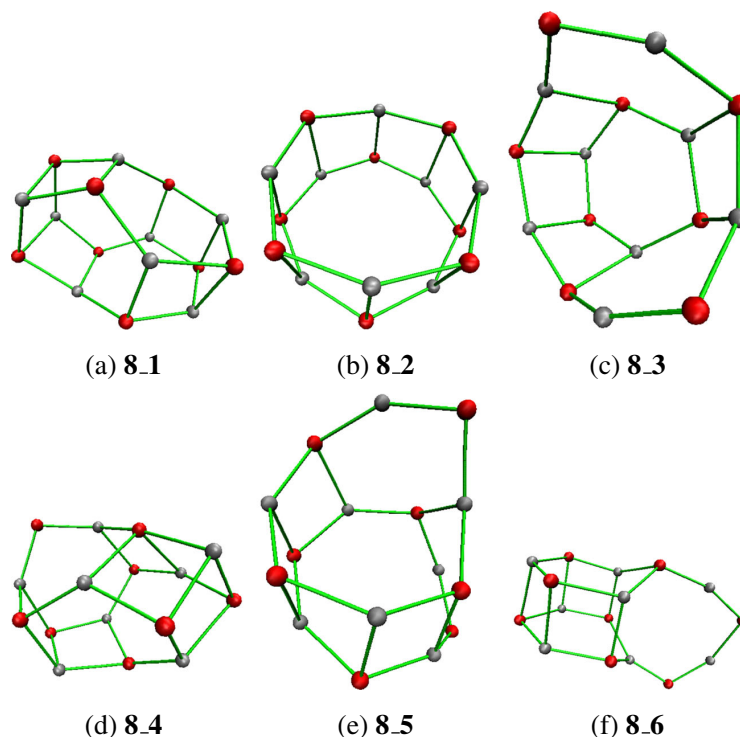
The CCSD(T)/TZP method was also considered for  $Zn_6O_6$  clusters, relative energies for the first 11 isomers are reported in Table 4, and the structures of the six first isomers are presented in Fig. 3. For these clusters, 36 structures were suggested by the stochastic method and KS, all of them are reported in ESI. In Table 4, the first row represents the relative energy obtained by using optimized structures, and the second row is related to relative energies obtained at M06-2X structures. From here, there are important differences among results obtained by several exchange-correlation functionals. First, PBE, PBE0, B3LYP, and M06, over optimized structures, predict relative energies smaller than those predicted by M06-2X, MP2, and CCSD(T) methods. In fact, these exchange-correlation functionals predict isomer **6.1** with the same energy than isomer **6.2**, and the isomer **6.3** presents the lowest energy. Such predictions do not coincide with those obtained by M06-2X, MP2, and CCSD(T) methods. For this



**Fig. 3** Six isomers of the  $Zn_6O_6$  cluster. Red spheres represent oxygen atoms and gray spheres represent zinc atoms

case, M06-2X gives the same ordering than that presented by CCSD(T). This result is relevant since some works use B3LYP geometries as a reference, which is not correct for this case. For such reports, isomer **6.3** is more stable than isomer **6.1**, which is different concerning M06-2X/TZP and CCSD(T) predictions. In this article, one additional isomer goes into the scene, since the isomer **6.2** in Fig. 3 competes in energy with **6.1** and **6.3**. From these three structures, experimental information indicates that 3D structures are most stable. Koyasu et al. deal with the corresponding cations,[26] where the ordering changes as regards to neutral systems. However, structures **6.1** and **6.2** are still the isomers with the lowest energy when they are charged, followed by the isomer **6.14**. The isomer **6.3** considered by Koyasu et al. exhibits high energy when this is charged and has low possibilities to be found in the gas phase, which suggest that 3D structures are more stable than 2D structures for  $n = 6$ .

**Fig. 4** Six isomers of the  $Zn_8O_8$  cluster. Red spheres represent oxygen atoms and gray spheres represent zinc atoms



Our stochastic procedure searched isomers for  $Zn_8O_8$ , and after it, the KS method was applied to optimize the corresponding candidates. In total, we found 56 structures, all of them are in [ESI](#). The six isomers with lowest energies are presented in [Fig. 4](#) and relative energies for the first 11 isomers are reported in [Table 5](#).

There are reports where the monocyclic isomer is mentioned as the most stable structure [22]. However, when the M06-2X/TZP method is used for  $n = 8$ , the one-ring structure is in the 26th place, with 3D structures as most stable.

From [Table 5](#), it is clear that the ordering found for relative energies, in  $Zn_8O_8$  clusters, is not changed by the temperature. Thus, the entropic contribution in these clusters is not relevant. The same observation was found for  $n = 2, 4$  and 6.

### Connectivity in ZnO clusters

The figures presented in this article, to show some isomers, are based on bond paths predicted by the AIM approach. In our procedure, we find first bond critical points and after

**Table 5** Relative Gibbs energies ( $\Delta G_i = G_i - G_1$ ) for the isomers found for  $Zn_8O_8$  clusters and several temperatures by using the M06-2X/TZP method. All quantities are in kJ/mol

	Temperature (K)							
	0	100	140	180	200	240	280	300
$\Delta G_2$	30.0	27.5	26.9	26.2	25.8	25.0	24.2	23.7
$\Delta G_3$	100.1	96.6	95.4	94.1	93.5	92.0	90.6	89.9
$\Delta G_4$	102.7	100.1	99.8	99.4	99.2	98.8	98.3	98.0
$\Delta G_5$	156.2	152.3	151.1	149.8	149.0	147.6	146.1	145.3
$\Delta G_6$	164.4	160.6	159.5	158.2	157.5	156.2	154.8	154.1
$\Delta G_7$	176.5	174.3	173.8	173.4	173.1	172.6	172.1	171.8
$\Delta G_8$	185.9	181.9	180.9	179.8	179.2	177.9	176.7	176.0
$\Delta G_9$	187.4	182.8	181.8	180.7	180.0	178.8	177.5	176.8
$\Delta G_{10}$	188.6	184.3	183.2	181.9	181.2	179.7	178.2	177.4



**Table 6** Electron density and its Laplacian evaluated at critical points found in ZnO clusters

Isomer	O-O		Zn-Zn	
	$\rho$	$\nabla^2\rho$	$\rho$	$\nabla^2\rho$
2.2	0.3148	0.0520	0.0342	0.0345
4.4	0.2197	0.2330	0.0482	0.0217
4.5	0.2152	0.2243	0.0478	0.0216
4.6	0.2400	0.2043	0.0374	0.0258

this step the differential equation to find the corresponding attractors are solved for each isomer considered in this article. If we compare these figures with those presented in other reports, there are clearly important differences. For example, the Zn-Zn contact is presented in some reports for the Zn<sub>2</sub>O<sub>2</sub> cluster, [14, 30], which is different than that presented in Fig. 1. In reference 14, there are many Zn-Zn contacts in several clusters, which is quite strange. For that reason, in this article, the AIM approach was used to obtain the contacts involved in ZnO clusters. In Tables 6 and 7, we present the contacts found for some Zn<sub>n</sub>O<sub>n</sub> clusters. For these cases, the electron density and its Laplacian are evaluated at critical points of the electron density. For  $n \geq 4$ , there are several critical points with the same nature, which appear in the same isomer, and for that reason, average quantities are reported, and the highest and the lowest values are reported above and below, respectively, of the average (bold font). From figures presented at this point, ring and cage critical points are expected for some isomers.

From these tables, it is evident the nature of each bond, since the electron density evaluated at bond critical points of O-O, Zn-Zn and Zn-O present different values with the ordering  $\rho_{\text{O-O}}(r_{cp}) > \rho_{\text{Zn-O}}(r_{cp}) > \rho_{\text{Zn-Zn}}(r_{cp})$ . These values have important effects over bond lengths; for example, the isomer **2.2** (Fig. 1b) presents all these bonds with  $d_{\text{O-O}} = 1.40 \text{ \AA}$ ,  $d_{\text{Zn-O}} = 1.94 \text{ \AA}$  and  $d_{\text{Zn-Zn}} = 2.50 \text{ \AA}$ . Thus, the smaller the bond length, the bigger  $\rho$  at the corresponding bond critical point. From Tables 6 and 7, it is evident that the Zn-O interaction is present with higher frequency than O-O and Zn-Zn contacts. In fact, for  $n = 6$  and 8, we have observed that O-O and Zn-Zn contacts are present only for isomers with high energy.

One-ring structures are clusters that present only Zn-O contacts, where one atom, oxygen or zinc, is involved in two bonds. When the size of the ring is increased, such a contact presents high  $\rho(r_{cp})$  values, as it can be appreciated from Table 7 for  $n = 2, 4$  and 6. Also, one-ring structures for  $n = 3, 5, 7, 8$  and 9 were analyzed by the AIM approach. The behavior of the electron density evaluated at the bond

critical point, of the Zn-O contact, is presented in Fig. 5, where it is clear that this quantity goes to a limit value for large rings.

For no one-ring structures, oxygen and zinc atoms present more than two contacts, as it can be appreciated for  $n > 2$ . For these contacts, the electron density evaluated at bond critical points are less than those values presented by one-ring structures. For this reason, in Table 7 we present an average of this quantity for the Zn-O contact, where several isomers present  $\rho(r_{cp}) \approx 0.13$  a. u., indicating that the corresponding contact is related to one atom with only two contacts. For  $n = 6$  and 8, in some isomers, there are atoms with 4, 5, 6, or 7 contacts, although all of them have associated high energies.

The AIM approach was applied over all isomers found in this article, where we can find one important conclusion, the rings involved in some isomers are built with the Zn-O pair, and for that reason, we did not find pentagons. In fact, the ZnO in crystal phase (zincite) exhibits only hexagons. [52]

For all clusters studied here, the Laplacian is positive indicating possible ionic interactions, [53] however, analyzing local kinetic energy, virial field, and total energy density, we determine that Zn-O contacts represent covalent bonds for all Zn<sub>n</sub>O<sub>n</sub> clusters. In this article, we search the critical points for the electron density for many ZnO clusters. It is well known that this kind of task takes a long time for large systems. For our case, this task was relatively fast since for the Zn<sub>10</sub>O<sub>10</sub> cluster, which has 190 orbitals and 1760 primitive functions, our code developed over GPUs [50, 51] spent 114 s by using a Quadro GP100 card and 82 s with a Titan V to find all critical points involved in this system. It is worth noting that these GPUs are in common desktops.

In the last section, we concluded that M06-2X geometries give reliable results by using the CCSD(T) method as the reference. In this article, we used critical points and bond paths obtained for each isomer to determine bond lengths of all contacts predicted by the AIM approach.

For each isomer, we found several Zn-O contacts, for example, the Zn<sub>2</sub>O<sub>2</sub> isomer presents four (Fig. 1a) and two (Fig. 1b) contacts. Naturally, each contact has a different bond length, and this quantity is different for each method used in the optimization process. For this reason, we report the Zn-O bond length, on average, predicted by each method and for each isomer in Fig. 6. For the Zn<sub>2</sub>O<sub>2</sub> cluster, there are only two isomers, which are in this figure. For Zn<sub>4</sub>O<sub>4</sub> and Zn<sub>6</sub>O<sub>6</sub> clusters, this figure contains the 11 low-lying energy isomers of each one.

For the smallest cluster considered in this article, Zn<sub>2</sub>O<sub>2</sub>, we can see from Fig. 6 that the XC functionals used here give similar structures among themselves, which are different than that predicted by MP2 for the isomer **2.2**. If we use the M06-2X/TZP method as the reference, in

**Table 7** Electron density and its Laplacian evaluated at critical points found in ZnO clusters

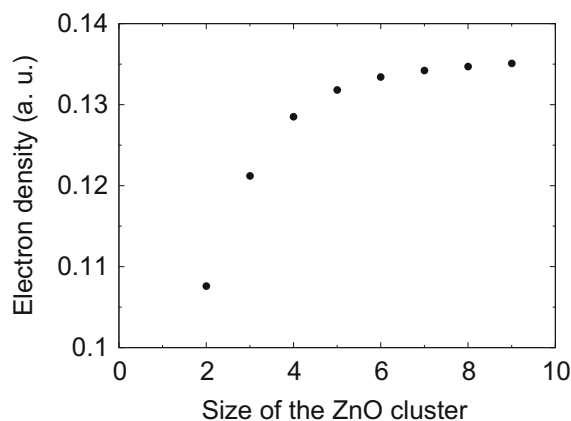
Isomer	Zn–O		Ring		Cage	
	$\rho$	$\nabla^2\rho$	$\rho$	$\nabla^2\rho$	$\rho$	$\nabla^2\rho$
Zn <sub>2</sub> O <sub>2</sub>						
<b>2.1</b>	0.1076	0.0395	0.0395	0.1564	–	–
<b>2.2</b>	0.0924	0.4352	0.0229	0.0806	–	–
Zn <sub>4</sub> O <sub>4</sub>						
	0.1284	0.6476	–	–	–	–
<b>4.1</b>	<b>0.1285</b>	<b>0.6480</b>	0.0037	0.0090	–	–
	0.1285	0.6481	–	–	–	–
	0.0844	0.4173	0.0290	0.0914	–	–
<b>4.2</b>	<b>0.0845</b>	<b>0.4179</b>	<b>0.0290</b>	<b>0.0915</b>	0.0174	0.0471
	0.0846	0.4185	0.0291	0.0915	–	–
	0.0854	0.4311	0.0127	0.0289	–	–
<b>4.3</b>	<b>0.1052</b>	<b>0.5273</b>	<b>0.0127</b>	<b>0.0289</b>	0.0118	0.0313
	0.1385	0.6901	0.0127	0.0289	–	–
	0.0761	0.4219	0.0051	0.0113	–	–
<b>4.4</b>	<b>0.1022</b>	<b>0.5400</b>	<b>0.0468</b>	<b>0.2711</b>	–	–
	0.1278	0.6467	0.0701	0.4111	–	–
	0.0810	0.4524	0.0047	0.0113	–	–
<b>4.5</b>	<b>0.1020</b>	<b>0.5449</b>	<b>0.0477</b>	<b>0.2734</b>	–	–
	0.1261	0.6381	0.0692	0.4045	–	–
	0.0631	0.3207	0.0105	0.0281	–	–
<b>4.6</b>	<b>0.0978</b>	<b>0.5040</b>	<b>0.0341</b>	<b>0.1760</b>	–	–
	0.1282	0.6439	0.0661	0.4077	–	–
Zn <sub>6</sub> O <sub>6</sub>						
	0.0706	0.3494	0.0076	0.0194	–	–
<b>6.1</b>	<b>0.0884</b>	<b>0.4495</b>	<b>0.0226</b>	<b>0.0720</b>	0.0051	0.0167
	0.0974	0.4996	0.0276	0.0896	–	–
	0.0445	0.2023	0.0148	0.0414	–	–
<b>6.2</b>	<b>0.0802</b>	<b>0.4037</b>	<b>0.0245</b>	<b>0.0759</b>	0.0126	0.0358
	0.0938	0.5131	0.0290	0.0928	–	–
	0.1334	0.6899	–	–	–	–
<b>6.3</b>	<b>0.1334</b>	<b>0.6899</b>	0.0001	0.0003	–	–
	0.1334	0.6899	–	–	–	–
	0.0624	0.3176	0.0102	0.0258	–	–
<b>6.4</b>	<b>0.1112</b>	<b>0.5733</b>	<b>0.0150</b>	<b>0.0507</b>	–	–
	0.1303	0.6618	0.0246	0.1004	–	–
	0.0687	0.3494	0.0046	0.0121	–	–
<b>6.5</b>	<b>0.0971</b>	<b>0.4890</b>	<b>0.0201</b>	<b>0.0642</b>	0.0045	0.0134
	0.1287	0.6467	0.0301	0.0979	–	–
	0.0776	0.4026	0.0046	0.0115	–	–
<b>6.6</b>	<b>0.0974</b>	<b>0.4917</b>	<b>0.0244</b>	<b>0.0800</b>	–	–
	0.1357	0.6860	0.0303	0.0989	–	–
Zn <sub>8</sub> O <sub>8</sub>						
	0.0744	0.3769	0.0076	0.0200	–	–
<b>8.1</b>	<b>0.0899</b>	<b>0.4617</b>	<b>0.0200</b>	<b>0.0646</b>	0.0036	0.0108
	0.1022	0.5378	0.0289	0.0954	–	–
	0.0638	0.3154	0.0017	0.0045	–	–
<b>8.2</b>	<b>0.0908</b>	<b>0.4681</b>	<b>0.0216</b>	<b>0.0713</b>	0.0013	0.0040

**Table 7** (continued)

Isomer	Zn-O		Ring		Cage	
	$\rho$	$\nabla^2\rho$	$\rho$	$\nabla^2\rho$	$\rho$	$\nabla^2\rho$
<b>8.3</b>	0.1044	0.5445	0.0266	0.0880	–	–
	0.0604	0.3004	0.0077	0.0198	–	–
	<b>0.0971</b>	<b>0.4985</b>	<b>0.0196</b>	<b>0.0637</b>	–	–
	0.1347	0.6729	0.0286	0.0938	–	–
<b>8.4</b>	0.0472	0.2190	0.0084	0.0197	–	–
	<b>0.0887</b>	<b>0.4514</b>	<b>0.0201</b>	<b>0.0632</b>	0.0042	0.0133
	0.1196	0.6088	0.0290	0.0947	–	–
<b>8.5</b>	0.0542	0.2650	0.0032	0.0092	–	–
	<b>0.0966</b>	<b>0.4936</b>	<b>0.0192</b>	<b>0.0634</b>	–	–
	0.1366	0.6719	0.0282	0.0923	–	–
<b>8.6</b>	0.0706	0.3478	0.0045	0.0117	–	–
	<b>0.0966</b>	<b>0.4935</b>	<b>0.0179</b>	<b>0.0570</b>	0.0056	0.0171
	0.1368	0.6859	0.0291	0.0982	–	–

this figure there are important differences between this exchange-correlation functional and the results obtained by the other functionals for  $Zn_4O_4$  and  $Zn_6O_6$ . In particular, we see the importance of the exact exchange, since there are pronounced differences between M06 and M06-2X results. Although there are few isomers where there are appreciable differences between MP2 and M06-2X results, the comparison between them is reasonably good.

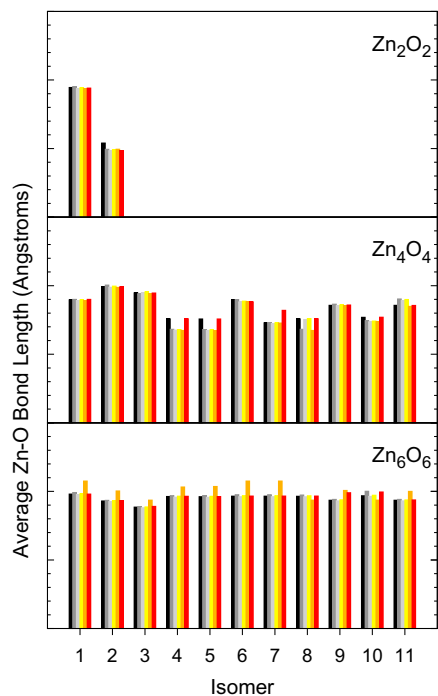
Finally, from this figure, we appreciate important deviations in the Zn-O bond length among isomers for  $Zn_2O_2$  and  $Zn_4O_4$  clusters. However, for  $Zn_6O_6$  this property exhibits small changes among all isomers. The Zn-O bond length, on average, for the zincite is 1.92 Å. [52] Thus, the  $Zn_6O_6$  starts to reflex properties contained in bulk.



**Fig. 5** Electron density evaluated at the bond critical point of Zn-O contacts for one-ring structures

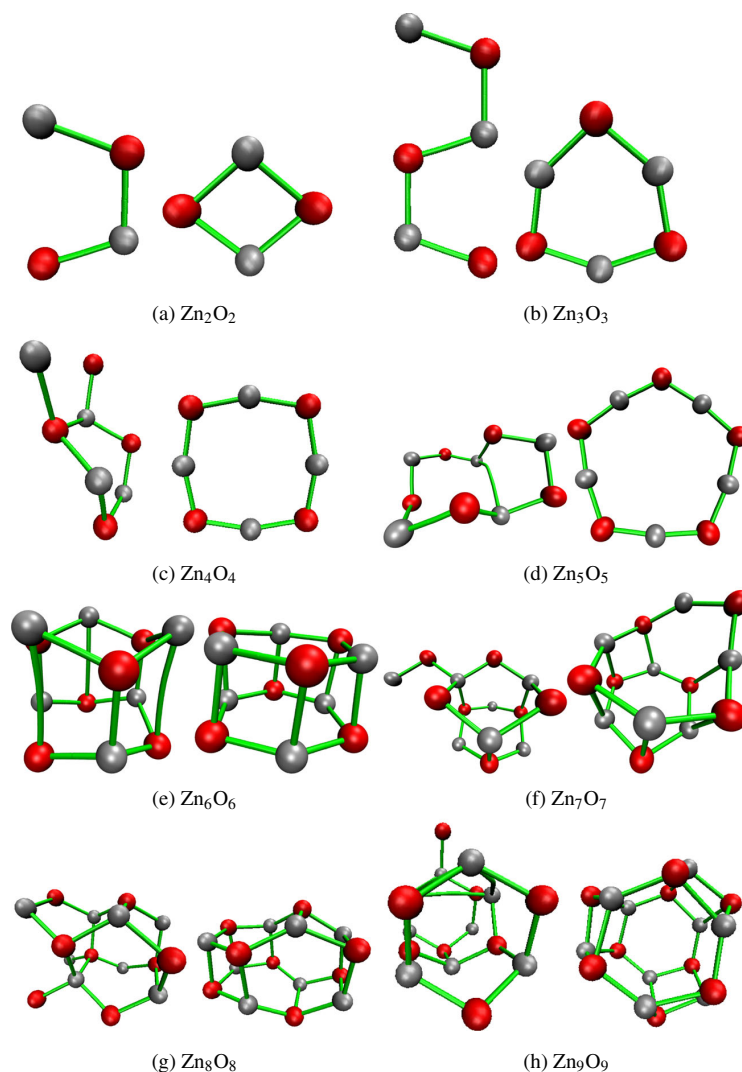
### Clusters from zincite

By analyzing the zincite, [52] the isomer **4.3** for  $Zn_4O_4$  (Fig. 2c) presents one structure quite similar to that involved



**Fig. 6** Bond length, on average, of the Zn-O contact for clusters reported in Tables 1, 3 and 4. For each isomer from left to right: MP2 (black), PBE (gray), PBE0 (silver), B3LYP (yellow), M06 (orange) and M06-2X (red)

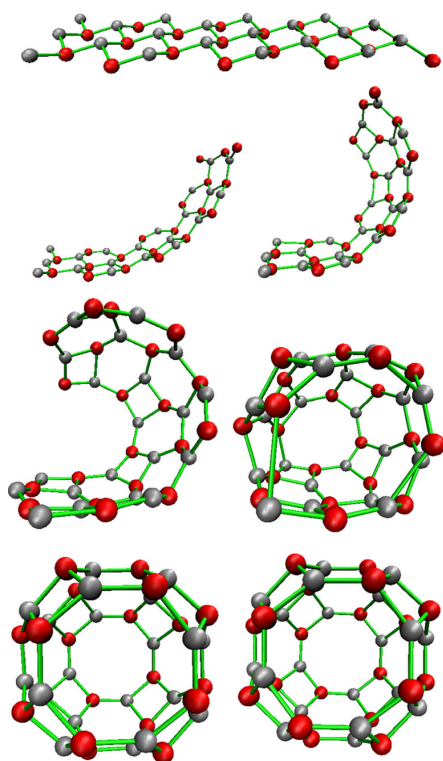
**Fig. 7**  $Zn_nO_n$  structures obtained from crystal information (*left side*) and the corresponding optimized structure (*right side*) for  $n = 1 - 9$



in this crystal. From this observation, we tried to find some structures obtained by stochastic methods from crystal information. In Fig. 7, we present structures where the starting point of the optimization process was obtained from the crystal structure (no modifications were performed over these structures before the optimization process), such structures are presented on the left side, and the corresponding optimized structures are depicted on the right side. In this figure, we present clusters with  $n = 2 - 9$ . For all these cases, we recovered the most stable structures obtained here and in other reports by stochastic methods. In Fig. 7, we want to stress the initial point to optimize  $Zn_3O_3$ , where the one-ring structure presents the lowest energy. For this case, the most obvious starting point is one hexagon obtained from one layer defined on the (001)

plane of zincite; however, for small clusters, the one-ring structure presents the lowest energy, and consequently, many proposals go to that final point on the PES.

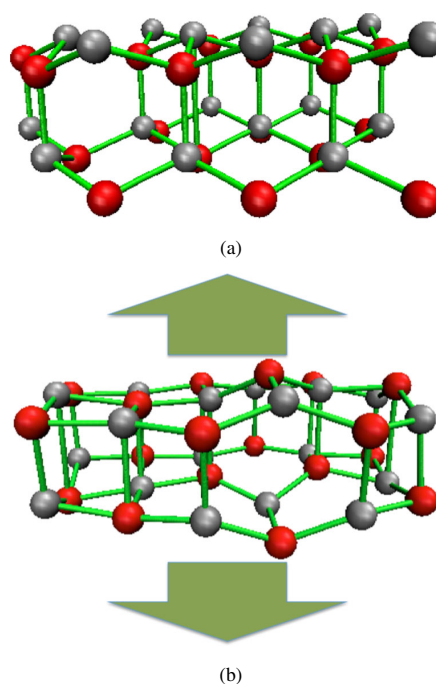
The impressive behavior presented by the  $Zn_3O_3$  cluster, to form one-ring structure from a completely different structure, is presented in other systems. In Fig. 8, the  $Zn_{20}O_{20}$  structure reported as the most stable by Chen et al. [30] is obtained from one layer on the (001) plane in zincite. For this case, we present several steps of the optimization process where it is clear that for the final step the structure acquires a compact form, where hexagons and squares predominate. In fact, the formation of the squares induces bends on the initial structure. It is worth noting that we did not induce any of these structures in the optimization process.



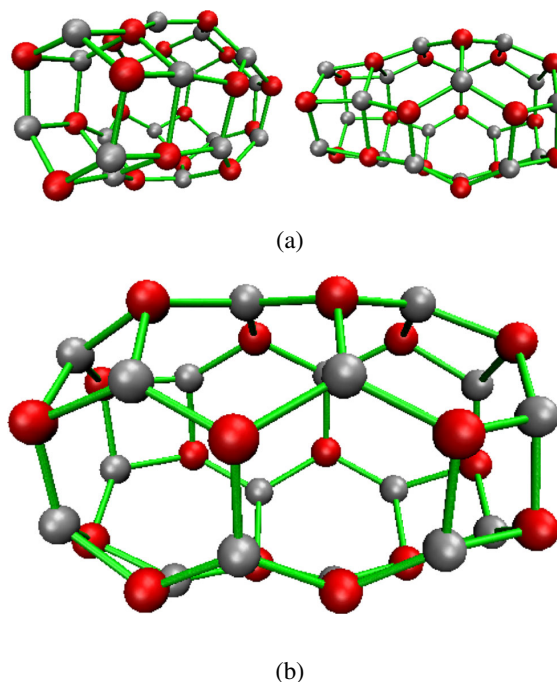
**Fig. 8** Optimization process for the  $Zn_{20}O_{20}$  cluster. The initial stage corresponds to one layer of the zincite, followed by 5, 10, 15, 20, and 40 steps of the optimization process. The final structure is presented at the end of this figure

For  $Zn_{18}O_{18}$ , we followed the same procedure used on  $Zn_{20}O_{20}$ . For this case, the cluster is folded in such a way that the edges are not touched by themselves. For this reason, in this case we adopted another strategy, where the starting point is built with two sheets of zincite, as this is presented in Fig. 9a. The corresponding optimization process yields one structure where some hexagons disappear to form squares and consequently the structure shows two planes almost parallel, although there are hexagons where some atoms are out of the flat zones (Fig. 9b). This structure suggests pulling out internal atoms to break the flat zones, as this is depicted through the arrows in Fig. 9b.

The distorted structure from Fig. 9b gives the structure in Fig. 10a. The hexagonal prism formed by  $Zn_6O_6$  can be appreciated in this structure (left side of Fig. 10a). This structure is quite similar to that reported as the global minimum (Fig. 10b) by Chen et al. [30]. It is important to note that the energy difference between structures depicted in Fig. 10a and b is 13.8 kJ/mol. Thus, these structures could coexist at high temperatures. For all these clusters, we can conclude that hexagons are preferred over squares. Such a



**Fig. 9**  $Zn_{18}O_{18}$  cluster. **a** Structure from zincite. **b** Optimized structure from zincite



**Fig. 10**  $Zn_{18}O_{18}$  cluster. **a** Isomer 13.8 kJ/mol above of the global minimum. **b** Global minimum

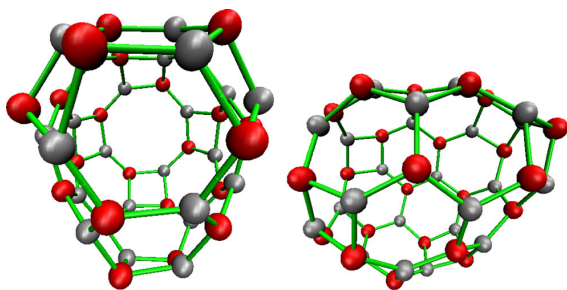


Fig. 11  $Zn_{23}O_{23}$  cluster

conclusion is valid also for the crystal where only hexagons are observed.

By the two strategies mentioned before, we found one structure for the  $Zn_{23}O_{23}$  cluster, which is proposed as one of the most stable structures. The structure of this cluster is presented in Fig. 11, where hexagons, squares, and one octagon are presented in its structure. We have analyzed  $Zn_{18}O_{18}$  and  $Zn_{23}O_{23}$  clusters because they are involved in one study where there is available experimental information, [26], which can be completed by using theoretical findings; these structures are provided in ESI and they could be useful to confirm experimental data.

## Conclusions

In this article, the performance of the M06-2X/TZVP method is contrasted with PBE, PBE0, B3LYP, M06, and MP2 methods, giving good results concerning CCSD(T) method for the analysis of  $Zn_nO_n$  clusters. This conclusion is reached for 24 isomers found for  $n = 2, 4,$  and  $6$  within a set of 49 generated by a simulated annealing implementation coupled with the Kohn–Sham method. In several reports, geometries obtained by B3LYP exchange–correlation functional are used as the reference, however, in this article, this exchange–correlation functional fails, and therefore we do not recommend this functional to study ZnO clusters. The analysis of possible contacts involved in all clusters by using the atoms in molecules approach indicates that hexagons and squares are the geometrical structures that dominate in all clusters. Also, few octagons are observed, however, pentagons were not observed as part of the structures found in this study.

This article found one important observation for  $Zn_nO_n$  clusters; many of them can be obtained from the crystal structure. This observation does not mean that the searching of most stable clusters of this system can be obtained easily since the main task is how to cut the crystal [54]. However, the possibilities involved in this task can be used as a starting point of genetic algorithms [55]. The observation is not

new and is related to the searching of clusters from crystal information or with the construction of a crystal from its basic structures [56]. Naturally, this observation is valid only for ZnO. Currently, in our laboratory, we are studying several metal oxides to see if the observation found in this article is also valid.

**Acknowledgements** This article is dedicated to Professor Pratim Kumar Chattaraj for his contributions around density-functional-theory and as part of the celebration of his 60th anniversary. We thank the Laboratorio de Supercómputo y Visualización en Paralelo at the Universidad Autónoma Metropolitana-Iztapalapa for access to their computer facilities. L.-A. S.-A. and R. H.-E. thank CONACYT, México, for the scholarships 265471 and 283251, respectively.

## References

1. Cai F, Wang J, Yuan Z, Duan Y (2012) Magnetic-field effect on dye-sensitized ZnO nanorods-based solar cells. *J Power Sources* 216:269–272
2. Wang Z (2004) Zinc oxide nanostructures: growth, properties and applications. *J Phys-Condens Matter* 16:R829–R858
3. Anandan S, Vinu A, Lovely KLPS, Gokulakrishnan N, Srinivasu P, Mori T, Murugesan V, Sivamurugan V, Ariga KA (2007) Photocatalytic activity of La-doped ZnO for the degradation of monocrotophos in aqueous suspension. *J Mol Catal A: Chem* 266:149–157
4. Dagdeviren C, Hwang S, Su Y, Kim S, Cheng H, Gur O, Haney R, Omenetto F, Huang Y, Rogers J (2013) Transient, biocompatible electronics and energy harvesters based on ZnO. *Small* 9:3398–3404
5. Patra A, Dutta A, Bhaumik A (2014) Self-assembled ultra small ZnO nanocrystals for dye-sensitized solar cell application. *J Solid State Chem* 215:135–142
6. Senay V, Pat S, Korkmaz S, Aydogmus T, Elmas S, Ozen S, Ekm N, Balbag M (2014) ZnO thin film synthesis by reactive radio frequency magnetron sputtering. *Appl Surf Sci* 318:2–5
7. Perdew J, Burke K, Ernzerhof M (1996) Generalized gradient approximation made simple. *Phys Rev Lett* 77:3865–3868
8. Becke A (1988) Density-functional exchange-energy approximation with correct asymptotic-behavior. *Phys Rev A* 38:3098–3100
9. Lee C, Yang W, Parr R (1988) Development of the Colle–Salvetti correlation-energy formula into a functional of the electron-density. *Phys Rev B* 37:785–789
10. Becke A (1993) Density-functional thermochemistry. III. The role of exact exchange. *J Chem Phys* 98:5648–5652
11. Stephens P, Devlin F, Chabalowski C, Frisch M (1994) Ab initio calculation of vibrational absorption and circular dichroism spectra using density functional force fields. *J Phys Chem* 98:11623–11627
12. Matxain J, Fowler J, Ugalde J (2000) Small clusters of II–VI materials:  $Zn_iO_i$ ,  $i = 1 - 9$ . *Phys Rev A* 62:053201
13. Matxain J, Mercero J, Fowler J, Ugalde J (2003) Electronic excitation energies of  $Zn_iO_i$  clusters. *J Am Chem Soc* 123:9494–9499
14. Wang B, Nagase S, Zhao J, Wang G (2007) Structural growth sequences and electronic properties of Zinc oxide clusters  $(ZnO)_n$  ( $n = 2 - 18$ ). *J Phys Chem C* 111:4956–4963
15. Zhao M, Xia Y, Tan Z, Liu X, Mei L (2007) Design and energetic characterization of ZnO clusters from first-principles calculations. *Phys Lett A* 372:39–43



16. Wang B, Wang X, Chen G, Nagase S, Zhao J (2008) Cage and tube structures of medium-sized zinc oxide clusters  $(\text{ZnO})_n$  ( $n = 24, 28, 36,$  and  $48$ ). *J Chem Phys* 128:144710
17. Cheng X, Li F, Zhao Y (2009) A DFT investigation on ZnO clusters and nanostructures. *J Mol Structure (Theochem)* 894:121–127
18. Wang X, Wang B, Tang L, Sai L, Zhao J (2010) What is atomic structures of  $(\text{ZnO})_{34}$  magic cluster? *Phys Lett A* 374:850–853
19. Wang B, Wang X, Zhao J (2010) Atomic structure of the magic  $(\text{ZnO})_{60}$  cluster: first-principles prediction of a sodalite motif for ZnO nanoclusters. *J Phys Chem C* 114:5741–5744
20. Wang W, Xu S, Ye L, Lei W, Cui Y (2011) Theoretical investigation of ZnO and its doping clusters. *J Mol Model* 2011:1075–1080
21. Trushin EV, Zilberberg IL, Bulgakov AV (2012) Structure and stability of small zinc oxide clusters. *Phys Solid State* 54:859–865
22. Abdolhosseini-Sarsari I, Javad-Hashemifar S, Salamati H (2012) First-principles study of ring to cage structural crossover in small ZnO clusters. *J Phys: Condens Matter* 24:505502
23. Gunaratne K, Berkdemir C, Harmon C, Castleman A Jr (2012) Investigating the relative stabilities and electronic properties of small zinc oxide clusters. *J Phys Chem A* 116:12429–12437
24. Yong Y, Wang Z, Liu K, Song B, He P (2012) Structures, stabilities, and magnetic properties of Cu-doped  $\text{Zn}_n\text{O}_n$  ( $n = 3, 9, 12$ ) clusters: a theoretical study. *Comput Theor Chem* 989:90–96
25. Flores-Hidalgo MA, Glossman-Mitnik D, Galvan DH, Barraza-Jimenez D (2013) Computational study of cage like  $(\text{ZnO})_{12}$  cluster using hybrid and hybrid meta functionals. *J Chin Chem Soc* 60:1082–1091
26. Koyasu K, Komatsu K, Misaizu F (2013) Structural transition of zinc oxide cluster cations: smallest tube like structure at  $(\text{ZnO})_6^+$ . *J Chem Phys* 139:164308
27. Woodley S, Sokol A, Catlow C, Al-Sunaidi A, Woodley S (2013) Structural and optical properties of Mg and Cd-doped ZnO nanoclusters. *J Phys Chem C* 117:27127–27145
28. Heinzelmann J, Koop A, Proch S, Ganteför G, Lazarski R, Sierka M (2014) Cage-like nanoclusters of ZnO probed by time-resolved photoelectron spectroscopy and theory. *J Phys Chem Lett* 5:2642–2648
29. Lazarski R, Sierka M, Heinzelmann J, Koop A, Sedlak R, Proch S, Ganteför G (2015) CdO and ZnO clusters as potential building blocks for cluster-assembled materials: a combined experimental and theoretical study. *J Phys Chem C* 119:6886–6895
30. Chen M, Straatsma T, Fang Z, Dixon D (2016) Structural and electronic property study of  $(\text{ZnO})_n$ ,  $n <= 168$ : transition from zinc oxide molecular clusters to ultrasmall nanoparticles. *J Phys Chem C* 120:20400–20418
31. Jain A, Kumar V, Kawazoe Y (2006) Ring structures of small ZnO clusters. *Comput Mater Sci* 36:258–262
32. Al-Sunaidi A, Sokol A, Catlow C, Woodley S (2008) Structures of zinc oxide nanoclusters: as found by revolutionary algorithm techniques. *J Phys Chem C* 112:18860–18875
33. Pérez J, Florez E, Hadad C, Fuentealba P, Restrepo A (2008) Stochastic search of the quantum conformational space of small lithium and bimetallic lithium-sodium clusters. *J Phys Chem A* 112(25):5749–5755
34. Pérez J, Hadad C, Restrepo A (2008) Structural studies of the water tetramer. *Int J Quantum Chem* 108:1653–1659
35. (2012) Stewart Computational Chemistry, Colorado Springs, CO, USA Mopac. <https://OpenMOPAC.net>
36. García-Hernández E, Flores-Moreno R, Vázquez-Mayagoitia A, Vargas R, Garza J (2016) Initial stage of the degradation of three common neonicotinoids: theoretical prediction of charge transfer sites. *New J Chem* 41:965–974
37. Adamo C, Barone V (1999) Toward reliable density functional methods without adjustable parameters: the PBE0 model. *J Chem Phys* 110:6158–6170
38. Zhao Y, Truhlar D (2008) The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, non-covalent interactions, excited states, and transition elements. *Theor Chem Acc* 120:215–241
39. Zhao Y, Truhlar D (2006) A new local density functional for main-group thermochemistry, transition metal bonding, thermochemical kinetics, and noncovalent interactions. *J Chem Phys* 125:194101
40. Moller C, Plesset M (1934) Note on an approximation treatment for many-electron systems. *Phys Rev* 46:0618–0622
41. Barbieri P, Fantin P, Jorge F (2006) Gaussian basis sets of triple and quadruple zeta valence quality for correlated wave functions. *Mol Phys* 104:2945–2954
42. Machado S, Camiletti G, Canal-Neto A, Jorge F, Jorge R (2009) Gaussian basis set of triple zeta valence quality for the atoms from K to Kr: application in DFT and CCSD(T) calculations of molecular properties. *Mol Phys* 107:1713–1727
43. Bartlett R, Shavitt I, Purvis G (1979) Quartic force-field of  $\text{H}_2\text{O}$  determined by many-body methods that include quadruple excitation effects. *J Chem Phys* 71:281–291
44. Bartlett R (1981) Many-body perturbation-theory and cluster theory for electron correlation in molecules. *Ann Rev Phys Chem* 32:359–401
45. Hirata S (2003) Tensor contraction engine: abstraction and automated parallel implementation of configuration-interaction, coupled-cluster, and many-body perturbation theories. *J Phys Chem A* 107:9887–9897
46. Balabanov N, Peterson K (2005) Systematically convergent basis sets for transition metals. I. All-electron correlation consistent basis sets for the 3D elements Sc–Zn. *J Chem Phys* 123:064107
47. Hay P, Wadt W (1985) Abinitio effective core potentials for molecular calculations—potentials for the transition-metal atoms Sc to Hg. *J Chem Phys* 82:270–283
48. Valiev M, Bylaska E, Govind N, Kowalski K, Straatsma T, Dam H, Wang D, Nieplocha J, Apra E, Windus T, de Jong W (2010) NWChem: a comprehensive and scalable open-source solution for large-scale molecular simulations. *Comput Phys Comm* 181:1477–1489
49. Bader R (1994) Principle of stationary action and the definition of a proper open system. *Phys Rev B* 49:13348–13356
50. Hernández-Esparza R, Mejía-Chica S, Zapata-Escobar A, Guevara-García A, Martínez-Melchor A, Hernández-Pérez J, Vargas R, Garza J (2014) Grid-based algorithm to search critical points, in the electron density, accelerated by graphics processing units. *J Comput Chem* 35:2272–2278
51. Hernández-Esparza R, Vázquez-Mayagoitia A, Soriano-Agueda L-A, Vargas R, Garza J (2018) GPUs as boosters to analyze scalar and vector fields in quantum chemistry. *Int J Quantum Chem*. <https://doi.org/10.1002/qua.25671>
52. Kihara K, Donnay G (1985) Anharmonic thermal vibrations on ZnO. *Can Mineral* 23:647–654
53. Espinosa E, Alkorta I, Elguero J, Molins E (2002) From weak to strong interactions: a comprehensive analysis of the topological and energetic properties of the electron density distribution involving x-H...F-y systems. *J Chem Phys* 117:5529–5542
54. Lü X, Xu X, Wang N, Zhang Q, Ehara M, Nakatsuji H (1998) Cluster modeling of metal oxides: how to cut out a cluster? *Chem Phys Lett* 291:445–452
55. Goldberg D (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading
56. Ouvrard C, Price S (2004) Toward crystal structure prediction for conformationally flexible molecules: the headaches illustrated by aspirin. *Cryst Growth Des* 4:1119–1127





---

## REFERENCIAS

- [1] Nutaro, J. J., 2011, *Building Software for Simulation: Theory and Algorithms, with Applications in C++*, John Wiley & Sons, Inc., New Jersey, 1a. edición.
- [2] Goldstine, H. H. y Goldstine, A., The Electronic Numerical Integrator and Computer (ENIAC), *J. Math. Tab. Comput.*, 2(15), (1946), páginas 97 – 110.
- [3] Thomas, L. H., The calculation of atomic fields, *Proc. Camb. Phil. Soc.*, 23(5), (1927), páginas 542 – 548.
- [4] Fermi, E., Eine statistische Methode zur Bestimmung einiger Eigenschaften des Atoms und ihre Anwendung auf die Theorie des periodischen Systems der Elemente, *Z. Phys.*, 48(1), (1928), páginas 73 – 79.
- [5] Hohenberg, P. y Kohn, W., Inhomogeneous electron gas, *Phys. Rev.*, 136, (1964), páginas 864 – 871.
- [6] Top 500. The List, <https://www.top500.org/>, consultada: Febrero 2018.
- [7] Nemoto, T., Fedorov, D. G., Uebayasi, M., Kanazawa, K., Kitaura, K., y Komeiji, Y., Ab initio fragment molecular orbital (FMO) method applied to analysis of the ligand–protein interaction in a pheromone-binding protein, *Comput. Biol. Chem.*, 29(6), (2005), páginas 434 – 439.
- [8] Dewar, M. J. S. y Thiel, W., Ground states of molecules. 38. The MNDO method. Approximations and parameters, *J. Am. Chem. Soc.*, 99(15), (1977), páginas 4899 – 4907.
- [9] Dewar, M. J. S., Zoebisch, E. G., Healy, E. F., y Stewart, J. J. P., Development and use of quantum mechanical molecular models. 76. AM1: A new general purpose quantum mechanical molecular model, *J. Am. Chem. Soc.*, 107(13), (1985), páginas 3902 – 3909.
- [10] Stewart, J. J. P., Optimization of parameters for semiempirical methods I. Method, *J. Comp. Chem.*, 10(2), (1989), páginas 209 – 220.
- [11] Stewart, J. J. P., Optimization of parameters for semiempirical methods VI: More modifications to the NDDO approximations and re-optimization of parameters, *J. Mol. Model.*, 19(1), (2013), páginas 1 – 32.
- [12] Levine, I. N., 2001, *Química Cuántica*, Pearson Educacion, Madrid, 5a. edición.

- [13] Szabo, A. y Ostlund, N., 1996, *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*, Dover Publications, Inc., New York, 1a. edición.
- [14] Jensen, E., 2007, *Introduction to Computational Chemistry*, John Wiley & Sons, Inc., Inglaterra, 2a. edición.
- [15] Parr, R. G. y Yang, W., 1989, *Density-Functional Theory of Atoms and Molecules*, Oxford Science Publications, New York, 1a. edición.
- [16] Bader, R. F. W., 1990, *Atoms in Molecules. A Quantum Theory*, Oxford Science Publications, New York, 1a. edición.
- [17] Bader, R. F. W., Principle of stationary action and the definition of a proper open system, *Phys. Rev. B.*, 49, (1994), páginas 13348 – 13356.
- [18] Popelier, P., 2000, *Atoms in Molecules. An Introduction*, Prentice Hall, United Kingdom, 1a. edición.
- [19] Matta, C. y Boyd, R., 2007, *The Quantum Theory of Atoms in Molecules: From Solid State to DNA and Drug Design*, John Wiley & Sons, Inc., New Jersey, 1a. edición.
- [20] Espinosa, E., Molins, E., y Lecomte, C., Hydrogen bond strengths revealed by topological analyses of experimentally observed electron densities, *Chem. Phys. Lett.*, 285, (1998), páginas 170 – 173.
- [21] Parthasarathi, R., Raman, S. S., Subramanian, V., y Ramasami, T., Bader's electron density analysis of hydrogen bonding in secondary structural elements of protein, *J. Phys. Chem. A*, 111(30), (2007), páginas 7141 – 7148.
- [22] Johnson, E. R., Keinan, S., Mori-Sánchez, P., Contreras-García, J., Cohen, A. J., y Yang, W., Revealing Noncovalent Interactions, *J. Am. Chem. Soc.*, 132(18), (2010), páginas 6498 – 6506.
- [23] Contreras-García, J., Johnson, E. R., Keinan, S., Chaudret, R., Piquemal, J.-P., Beratan, D. N., y Yang, W., NCIPLOT: A Program for Plotting Noncovalent Interaction Regions, *J. Chem. Theory Comput.*, 7(3), (2011), páginas 625 – 632.
- [24] Perdew, J. P., Burke, K., y Ernzerhof, M., Generalized gradient approximation made simple, *Phys. Rev. Lett.*, 77, (1996), páginas 3865 – 3868.
- [25] Becker, E. B., Carey, G. F., y Oden, J. T., 1981, *Finite Elements an Introduction vol. 1*, Prentice Hall, New Jersey, 1a. edición.
- [26] Ram-Mohan, L. R., 2002, *Finite Element and Boundary Element Applications in Quantum Mechanics*, Oxford University Press, New York, 1a. edición.
- [27] George, P. y Borouchaki, H., 1998, *Delaunay Triangulation and Meshing: Application to Finite Elements*, Editions Hermes, Paris, 1a. edición.
- [28] Rebay, S., Efficient Unstructured Mesh Generation by Means of Delaunay Triangulation and Bowyer-Watson Algorithm, *J. Comp. Phys.*, 106, (1993), páginas 125 – 138.
- [29] Humphrey, W., Dalke, A., y Schulten, K., VMD - Visual Molecular Dynamics, *J. Mol. Graphics*, 14, (1996), páginas 33 – 38.
- [30] Jmol: an open-source java viewer for chemical structures in 3d.
- [31] Schrödinger, LLC, 2015, The PyMOL molecular graphics system, version 1.8.
- [32] Schaftenaar, G. y Noordik, J., Molden: a pre- and post-processing program for molecular and electronic structures, *J. Comput. Aid. Mol. Des.*, 14(2), (2000), páginas 123 – 134.

- 
- [33] Schroeder, W. J., Martin, K. M., Avila, L. S., y Law, C. C., 2001, *The Visualization Toolkit User's Guide, Version 4.0*, Kitware, Inc.
- [34] Childs, H., Brugger, E., Whitlock, B., Meredith, J., Ahern, S., Pugmire, D., Biagas, K., Miller, M., Harrison, C., Weber, G. H., Krishnan, H., Fogal, T., Sanderson, A., Garth, C., Bethel, E. W., Camp, D., Rübel, O., Durant, M., Favre, J. M., y Navrátil, P., 2012, VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data, en *High Performance Visualization - Enabling Extreme-Scale Scientific Insight*, páginas 357 – 372.
- [35] Cuda Zone, <https://developer.nvidia.com/cuda-zone>, consultada: Febrero 2018.
- [36] Sanders, J. y Kandrot, E., 2010, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Addison-Wesley, New Jersey, 1a. edición.
- [37] Cook, S., 2013, *CUDA Programming: A Developer's Guide to Parallel Computing with GPUs*, Morgan Kaufmann Publishers Inc., California, USA, 1a. edición.
- [38] Farber, R., 2011, *CUDA Application Design and Development*, Morgan Kaufmann Pub Inc., California, USA, 1a. edición.
- [39] Stone, J. E., Hardy, D. J., Ufimtsev, I. S., y Schulten, K., GPU-accelerated molecular modeling coming of age, *J. Mol. Graph. Model.*, 29(2), (2010), páginas 116–125.
- [40] Yasuda, K., Accelerating density functional calculations with graphics processing unit, *J. Chem. Theory Comput.*, 4(8), (2008), páginas 1230–1236.
- [41] Nitsche, M. A., Ferreria, M., Mocskos, E. E., y Gonzalez Lebrero, M. C., GPU Accelerated Implementation of Density Functional Theory for Hybrid QM/MM Simulations, *J. Chem. Theory Comput.*, 10(3), (2014), páginas 959–967.
- [42] Wu, X., Koslowski, A., y Thiel, W., Semiempirical Quantum Chemical Calculations Accelerated on a Hybrid Multicore CPU-GPU Computing Platform, *J. Chem. Theory Comput.*, 8(7), (2012), páginas 2272–2281.
- [43] Miao, Y. y Merz, K. M., Jr., Acceleration of Electron Repulsion Integral Evaluation on Graphics Processing Units via Use of Recurrence Relations, *J. Chem. Theory Comput.*, 9(2), (2013), páginas 965–976.
- [44] Hernández-Esparza, R., Mejía-Chica, S. M., Zapata-Escobar, A. D., Guevara-García, A., Martínez-Melchor, A., Hernández-Pérez, J. M., Vargas, R., y Garza, J., Grid-based algorithm to search critical points, in the electron density, accelerated by graphics processing units, *J. Comp. Chem.*, 35, (2014), páginas 2272 – 2278.
- [45] B. Kirk, D. y mei W. Hwu, W., 2010, *Programming Massively Parallel Processors: A Hands-on Approach*, Applications of GPU Computing Series, New York, 1a. edición.
- [46] Scarpino, M., 2012, *OpenCL in Action: How to Accelerate Graphics and Computation*, Manning Publications Co., New York, 1a. edición.
- [47] Banger, R. y Bhattacharyya, K., 2013, *OpenCL Programming by Example*, Packt Publishing, Reino Unido, 1a. edición.
- [48] Tay, R., 2013, *OpenCL Parallel Programming Development Cookbook*, Packt Publishing, Reino Unido, 1a. edición.
- [49] Boys, S. F., Electronic Wave Functions I. A General Method of Calculation for the Stationary States of any Molecular System, *Proc. R. Soc. Lon. Ser-A*, 200(1063), (1950), páginas 542–554.

- [50] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, B. Mennucci, G. A. Petersson, H. Nakatsuji, M. Caricato, X. Li, H. P. Hratchian, A. F. Izmaylov, J. Bloino, G. Zheng, J. L. Sonnenberg, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. Bearpark, J. J. Heyd, E. Brothers, K. N. Kudin, V. N. Staroverov, T. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, N. Rega, J. M. Millam, M. Klene, J. E. Knox, J. B. Cross, V. Bakken, C. Adamo, J. Jaramillo, R. Gomperts, R. E. Stratmann, O. Yazyev, A. J. Austin, R. Cammi, C. Pomelli, J. W. Ochterski, R. L. Martin, K. Morokuma, V. G. Zakrzewski, G. A. Voth, P. Salvador, J. J. Dannenberg, S. Dapprich, A. D. Daniels, O. Farkas, J. B. Foresman, J. V. Ortiz, J. Cioslowski, and D. J. Fox, Gaussian 09 Revision B.01, Gaussian Inc. Wallingford CT 2016.
- [51] Schmidt, M. W., Baldrige, K. K., Boatz, J. A., Elbert, S. T., Gordon, M. S., Jensen, J. H., Koseki, S., Matsunaga, N., Nguyen, K. A., Su, S., Windus, T. L., Dupuis, M., y Montgomery, J. A., General Atomic and Molecular Electronic Structure System, *J. Comp. Chem.*, 14(11), (1993), páginas 1347 – 1363.
- [52] Valiev, M., Bylaska, E., Govind, N., Kowalski, K., Straatsma, T., Dam, H. V., Wang, D., Nieplocha, J., Apra, E., Windus, T., y de Jong, W., NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations, *Comput. Phys. Commun.*, 181(9), (2010), páginas 1477 – 1489.
- [53] Stewart, J. J. P., MOPAC: A semiempirical molecular orbital program, *J. Comput. Aid. Mol. Des.*, 4(1), (1990), páginas 1 – 103.
- [54] Stewart, J. J. P., MOPAC2016, Stewart Computational Chemistry, Colorado Springs.
- [55] Pettersen, E., Goddard, T., Huang, C., Couch, G., Greenblatt, D., Meng, E., y Ferrin, T., UCSF Chimera - A visualization system for exploratory research and analysis, *J. Comp. Chem.*, 25(13), (2004), páginas 1605 – 1612.
- [56] Henderson, A. *et al.*, 2004, *The ParaView Guide*.
- [57] Bohorquez, H. J. y Boyd, R. J., A localized electrons detector for atomic and molecular systems, *Theor. Chem. Acc.*, 127(4), (2010), páginas 393–400.
- [58] Becke, A. D. y Edgecombe, K. E., A simple measure of electron localization in atomic and molecular systems, *J. Chem. Phys.*, 92, (1990), página 5397.
- [59] Schmider, H. y Becke, A., Chemical content of the kinetic energy density, *J. Mol. Struct. (THEOCHEM)*, 527(1), (2000), páginas 51 – 61.
- [60] Politzer, P. y Truhlar, D. G., 1981, *Chemical applications of atomic and molecular electrostatic potential*, Plenum Press, New York, 1a. edición.
- [61] Gadre, S. R., Kulkarni, S. A., y Shrivastava, I. H., Molecular electrostatic potentials: A topographical study, *J. Chem. Phys.*, 96(7), (1992), páginas 5253 – 5260.
- [62] Obara, S. y Saika, A., Efficient recursive computation of molecular integrals over Cartesian Gaussian functions, *J. Chem. Phys.*, 84(7), (1986), páginas 3963–3974 .
- [63] Lu, T. y Chen, F., Multiwfn: A multifunctional wavefunction analyzer, *J. Comput. Chem.*, 33, (2012), páginas 580–592.
- [64] Riley, K., Hobson, M., y Bence, S., 2002, *Mathematical Methods for Physics and Engineering: A Comprehensive Guide*, Cambridge University Press, New York, 2a. edición.

- 
- [65] Arfken, G., 1970, *Mathematical methods for physicists*, Academic Press International Edition, New York, 2a. edición.
- [66] Dovesi, R., Orlando, R., Erba, A., Zicovich-Wilson, C. M., Civalleri, B., Casassa, S., Maschio, L., Ferrabone, M., Pierre, M. D. L., D'Arco, P., Noël, Y., Causa, M., Rerat, M., y Kirtman, B., CRYSTAL14: A program for the ab initio investigation of crystalline solids, *Int. J. Quantum Chem.*, 114(19), (2014), páginas 1287 – 1317.
- [67] Inc., W. R., Mathematica, Version 11.0.1 for linux x86 (64-bit), champaign, IL, 2017.
- [68] Fischer, C., 1977, *The Hartree-Fock Method for Atoms: A Numerical Approach*, A Wiley-Interscience publication, John Wiley & Sons, Inc., 1a. edición.
- [69] Becke, A. D., A multicenter numerical integration scheme for polyatomic molecules, *J. Chem. Phys.*, 88(4), (1988), páginas 2547 – 2553.
- [70] Becke, A. D. y Dickson, R. M., Numerical solution of Poisson's equation in polyatomic molecules, *J. Chem. Phys.*, 89(5), (1988), páginas 2993 – 2997.
- [71] Slater, J. C., Atomic Radii in Crystals, *J. Chem. Phys.*, 41(10), (1964), páginas 3199 – 3204.
- [72] Cortés-Santiago, A., Vargas, R., y Garza, J., Noble Gases Encaged by the C<sub>60</sub> Increase Their Chemical Reactivity, *J. Mex. Chem. Soc.*, 56(3), (2012), páginas 270 – 274.
- [73] Wigner, E. y Huntington, H., On the possibility of a metallic modification of hydrogen, *J. Chem. Phys.*, 3, (1935), página 764.
- [74] Rodríguez-Bautista, M., Díaz-García, C., Navarrete-López, A. M., Vargas, R., y Garza, J., Roothaan's approach to solve the hartree-fock equations for atoms confined by soft walls: Basis set with correct asymptotic behavior, *J. Chem. Phys.*, 143(3), (2015), página 034103.
- [75] Ley-Koo, E. y Rubinstein, S., The hydrogen atom within spherical boxes with penetrable walls, *J. Chem. Phys.*, 71(1), (1979), páginas 351 – 357.
- [76] Martínez-Sánchez, M.-A., Aquino, N., Vargas, R., y Garza, J., Exact solution for the hydrogen atom confined by a dielectric continuum and the correct basis set to study many-electron atoms under similar confinements, *Chem. Phys. Lett.*, 690, (2017), páginas 14 – 19.
- [77] Ludeña, E. V., SCF calculations for hydrogen in spherical box, *J. Chem. Phys.*, 66 (2), (1977), página 468.
- [78] Ludeña, E. V., SCF Hartree-Fock calculations of ground state wavefunctions of compressed atoms, *J. Chem. Phys.*, 69, (1978), página 1770.
- [79] Garza, J., Hernández-Pérez, J.-M., Ramírez, J.-Z., y Vargas, R., Basis set effects on the Hartree-Fock description of confined many-electron atoms, *J. Phys. B: At. Mol. Opt. Phys.*, 45, (2012), páginas 1 – 6.
- [80] Connerade, J., Dolmatov, V., Lakshmi, P., y Manson, S., Electron structure on endohedrally confined atoms: atomic hydrogen in an attractive shell, *J. Phys. B: At. Mol. Opt. Phys.*, 32, (1999), página 239.
- [81] Ley-Koo, E. y Cruz, S. A., The hydrogen atom and the H<sub>2</sub><sup>+</sup> and HeH<sup>++</sup> molecular ions inside prolate spheroidal boxes, *J. Chem. Phys.*, 74, (1981), página 4603.
- [82] Cruz, S. A. y Colín-Rodríguez, R., Spheroidal confinement of a single electron and of the hydrogen atom, the H<sub>2</sub><sup>+</sup> and HeH<sup>++</sup> molecular ions with arbitrary nuclear positions along the major axis, *Int. J. Quantum Chem.*, 109, (2009), páginas 3041 – 3054.

- [83] Colín-Rodríguez, R. y Cruz, S. A., The hydrogen molecule inside prolate spheroidal boxes: full nuclear position optimization, *J. Phys. B: At. Mol. Opt. Phys.*, 43, (2010), páginas 1 – 8.
- [84] López-Albarrán, P., Pizzi, A., Navarro-Santos, P., Hernández-Esparza, R., y Garza, J., Oligolignols within lignin-adhesive formulations drive their Young's modulus: A ReaxFF-MD study, *Int. J. Adhes. Adhes.*, 78, (2017), páginas 227 – 233.
- [85] Zhao, Y. y Truhlar, D. G., The M06 suite of density functionals for main group thermochemistry, thermochemical kinetics, noncovalent interactions, excited states, and transition elements: two new functionals and systematic testing of four M06-class functionals and 12 other functionals, *Theor. Chem. Acc.*, 120(1-3), (2008), páginas 215–241.
- [86] Sivaraman, G., Amorim, R. G., Scheicher, R. H., y Fyta, M., Benchmark investigation of diamondoid-functionalized electrodes for nanopore DNA sequencing, *Nanotechnology*, 27(41).
- [87] Hay, P. J. y Wadt, W. R., Ab initio effective core potentials for molecular calculations. Potentials for K to Au including the outermost core orbitals, *J. Chem. Phys.*, 82(1), (1985), páginas 299 – 310.