UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS QUÍMICAS

IMPLEMENTACIÓN DE LA TEORÍA DE LOS FUNCIONALES DE LA DENSIDAD AUXILIAR ACELERADA MEDIANTE COPROCESAMIENTO EN PARALELO

TESIS

PARA OPTAR POR EL GRADO DE

MAESTRA EN CIENCIAS

PRESENTA

Q. XIAOMIN HUANG

TUTOR: DR. JORGE MARTÍN DEL CAMPO RAMÍREZ

FACULTAD DE QUÍMICA, UNAM

Ciudad de México, Junio 2017

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS QUÍMICAS

IMPLEMENTACIÓN DE LA TEORÍA DE LOS FUNCIONALES DE LA DENSIDAD AUXILIAR ACELERADA MEDIANTE COPROCESAMIENTO EN PARALELO

T E S I S

PARA OPTAR POR EL GRADO DE

MAESTRA EN CIENCIAS

P R E S E N T A

Q. XIAOMIN HUANG

Ciudad de México, Junio 2017

Agradecimientos

- Al Dr. Jorge Martín del Campo Ramírez por darme la oportunidad de participar en este proyecto, por todo el apoyo y los consejos que me ha otorgado en todos los ámbitos académicos, profesionales y personales, y sobre todo, por creer en mí.

- A los miembros del jurado de mi examen de grado, Dr. Emilio Orgaz Baqué, Dr. Carlos Amador Bedolla, Dr. Alberto Vela Amieva, Dr. Rubén Santamaría y Dr. Tomás Rocha Rinza por sus sugerencias para mejorar el presente trabajo y su valioso tiempo para llevar a cabo el examen de grado.

- A la Universidad Nacional Autónoma de México y al Posgrado en Ciencias Químicas por brindarme la oportunidad de continuar con mi formación profesional.

- Al Consejo Nacional de Ciencia y Tecnología (CONACyT), por el apoyo económico brindado a través de la beca con número 577204.

- A la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGTIC) por los recursos computacionales otorgados mediante el proyecto SC16-I-IR-12.

- A la Dirección General de Asuntos del Personal Acaadémio por el apoyo económico brindado para adquirir equipo de cómputo mediante el proyecto IA-104516.

- A mi mamá, papá y toda la familia por estar siempre a mi lado. No habría logrado esto sin ustedes.

- A mis amigos y colegas, Rodrigo Cortés, Felipe Huan, Demetrio Cumplido, Augusto González, Nancy Barrueta, Martha Flores, Gerardo Álvarez, Aimee Torres, Ulises Torres, Iván Flores, Joaquín Flores, María del Mar, Óscar Aguilar, por su apoyo e inigualable compañía.

非常感谢我全家.
我爱你们.

Este trabajo fue desarrollado en el departamento de Física y Química Teórica de la Facultad de Química de la Universidad Nacional Autónoma de México. Los resultados derivados de este proyecto fueron presentados en:

- XIV Reunión Mexicana de Fisicoquímica Teórica. Modalidad cartel. Tonalá, Guadalajara. 19 a 21 de noviembre de 2015.
- Frontiers in Computational Chemistry 2016. Modalidad cartel. Facultad de Química, UNAM. 24 y 25 de agosto de 2016.
- Jornada de la Investigación en la FQ 2016. Modalidad cartel. Facultad de Química, UNAM. 12 a 14 de octubre de 2016.
- XV Reunión Mexicana de Fisicoquímica Teórica. Modalidad cartel. Yucatán, Mérida. 17 a 19 de noviembre de 2016.

Thesis advisor: Prof. Dr. Jorge Martín del Campo Xiaomin Huang

Implementation of auxiliary density functional theory accelerated by parallel coprocessing

Abstract

Computational chemistry has become a useful tool for the prediction of structures and properties of molecules of chemical interest. As the complexity of such molecules or the methodology of the study increases, more efficient mathematical algorithms and programs are demanded. Nowadays, reducing the calculation time, by using approximations, more advanced computers, or both, is one of the main objectives of computational codes.

This work presents the acceleration of a serial code using a graphics processing unit (GPU) for a self-consistent field iteration based on the auxiliary density functional theory (ADFT). The validation, with linear chains of hydrogen atoms and linear alkanes, resulted in a significative decrease of the execution time. The evaluation of electron repulsion integrals (ERI) and the calculation of the Becke weights for the molecular grid reached an acceleration of $\approx 40\times$ and $\approx 16\times$, respectively.

An advantage of the method developed herein is its modularity, hence its implementation on other serial codes is straightforward. In particular, these modules have potential application on Born-Oppenheimer simulations of large intervals of time with the advantage of not saturating the random access memory (RAM) of the GPU when using basis functions of d angular momentum, because ADFT implies two and three center ERIs, that require less RAM.

Director de la tesis: Dr. Jorge Martín del Campo Xiaomin Huang

Implementación de la teoría de los funcionales de la densidad auxiliar acelerada mediante coprocesamiento en paralelo

Resumen

La química computacional se ha convertido en una herramienta útil para la predicción de estructuras y propiedades de moléculas de interés químico. Conforme las moléculas o la metodología del estudio son más complejas, aumenta la demanda de programas y algoritmos matemáticos eficientes. Actualmente, la reducción de los tiempos de cálculo, mediante aproximaciones, uso de equipo de cómputo más avanzado, o ambos, es uno de los objetivos principales de los códigos computacionales.

En este trabajo se presenta la aceleración de un código serial mediante el uso de unidades de procesamiento gráfico (GPU, por sus siglas en inglés) para una iteración del ciclo del potencial autoconsistente basado en la teoría de los funcionales de la densidad auxiliar (ADFT, por sus siglas en inglés). La validación, con cadenas lineales de átomos de hidrógeno y alcanos lineales, resultó en una disminución significativa del tiempo de ejecución. La evaluación de las integrales de repulsión electrónica (ERI, por sus siglas en inglés) y la obtención de los pesos de Becke para el mallado molecular alcanzan aceleraciones de $\approx 40\times$ y $\approx 16\times$, respectivamente.

Una ventaja del método desarrollado en este trabajo es que su programación es modular y por ende su implementación en otros códigos seriales es directa. En particular, estos módulos tienen potencial aplicación a simulaciones de tipo Born-Oppenheimer de intervalos de tiempo largos con la ventaja de no saturar la memoria de acceso aleatorio (RAM, por sus siglas en inglés) de la GPU al emplear funciones base de momento angular d, porque la ADFT implica ERI de sólo tres y dos centros, que requieren de menos RAM.

Contenido

Lista de tablas

Lista de figuras

Introducción

Uno de los problemas fundamentales de la química teórica es encontrar la solución de la ecuación de Schrödinger independiente del tiempo, no relativista, para sistemas de muchos cuerpos interactuantes. La solución nos permite conocer las estructuras y propiedades electrónicas de los sistemas químicos. Analíticamente, sólo se puede resolver la ecuación de Schrödinger para dos partículas interactuantes. En el artículo *Quantum Mechanics of Many-Electron Systems* de 1929, Paul Dirac menciona que [1]:

"The underlying physical laws necessary for the mathematical theory of a large part of physics and the whole of chemistry are thus completely known, and the difficulty is only that the exact application of these equations leads to equations much too complicated to be soluble. It therefore becomes desirable that approximate practical methods of applying quantum mechanics should be developed, which can lead to an explanation of the main features of complex atomic systems without too much computation[1]*."*

Dicha frase inspira ilusión y desesperanza, simultáneamente; ilusión porque afirma que la solución de casi cualquier problema físico o químico existe, y desesperanza porque es demasiado complicado resolver las ecuaciones que implica.

En las siguientes décadas, químicos de lugares distintos formularon métodos matemáticos para simplificar, mediante aproximaciones, la resolución del problema mecánico-cuántico. En esos años se desarrolló el método de potencial autoconsistente (SCF, por sus siglas en inglés), otra herramienta fundamental que utilizan los algoritmos modernos para la resolución de la

[1] Las leyes físicas subyacentes necesarias para la teoría matemática de una gran parte de la física y de toda la química son, por lo tanto, completamente conocidas y la dificultad es sólo que la aplicación exacta de estas ecuaciones da lugar a ecuaciones demasiado complicadas para ser resueltas. Por lo tanto, es deseable que se desarrollen métodos prácticos aproximados de aplicación de la mecánica cuántica, los cuales pueden llegar a explicar las características principales de sistemas atómicos complejos sin realizar demasiados cálculos numéricos.

ecuación de Schrödinger independiente del tiempo y Hartree logró aplicar el método en átomos y iones pequeños. Era inimaginable que se lograra calcular la energía de un átomo pesado e imposible para una molécula. Además de los métodos *ab initio* basados en función de onda, en 1964, Hohenberg y Kohn demuestran que a partir de la densidad del estado basal se puede recuperar la información del sistema [2]. En el libro *Neither Chemistry nor Physics*, Simões y Gavroglu describen este periodo de la siguiente manera [3]: *"... until the extensive use of digital computers in the 1970s, the history of quantum chemistry is a history of the attempts to devise strategies of how to overcome the almost self-negating enterprise of using quantum mechanics for explaining chemical phenomena*[2]*."*

La invención de las computadoras electrónicas adquirió un papel importante en el desarrollo de la mecánica cuántica, ya que fue hasta ese momento en que se pudieron implementar todas las aproximaciones desarrolladas hasta los 1970's. Hoy en día, es claro que la química teórica es multidisciplinaria, ya que emplea fundamentos físicos para explicar fenómenos químicos mediante algoritmos implementados en computadoras. Es por esto que los avances de los últimos 50 años en química teórica dependieron del desarrollo de *hardware*. Actualmente es posible estudiar sistemas de cientos de átomos con cúmulos de computadoras, pero siguen existiendo limitaciones. La demanda por computadoras cada vez más avanzados en la química teórica moderna sigue creciendo. Aunque no se ha logrado calcular todas las propiedades moleculares con exactitud, ni reemplazar experimentos, la química computacional sí ha complementado el trabajo experimental. Por ejemplo, en la industria farmacéutica se emplean computadoras para descartar moléculas que carecen de propiedades farmacológicas, mientras que en investigaciones en materiales se utilizan en el diseño de nanocompuestos para celdas fotovoltáicas.

La mayoría de los programas que realizan cálculos de estructura electrónica están optimizados para unidades de procesamiento central (CPU, por sus siglas en inglés). La capacidad computacional de una CPU depende del número de transistores y sus velocidades de reloj. Se puede aumentar el número de transistores y la velocidad de reloj, pero están limitados por la cantidad de calor que emiten. Para evitar esto, se puede utilizar transistores más pequeños y eficientes par construir procesadores denominados unidades de procesamiento gráfico (GPU, por sus siglas en inglés). La GPU está optimizada para trabajo en paralelo con instrucciones sencillas, por lo que se utiliza ampliamente en la modificación de imágenes.

[2] ...hasta el uso generalizado de computadoras digitales en la década de 1970, la historia de la química teórica es una historia de los intentos por diseñar estrategias de cómo superar la iniciativa casi auto-negada de usar la mecánica cuántica para explicar los fenómenos químicos.

En 2007, la arquitectura de dispositivos de cómputo unificado (CUDA, por sus siglas en inglés) facilitó la programación de GPU dando lugar a cómputo de propósito general en GPU (GPGPU, por sus siglas en inglés). Con CUDA, la comunidad científica adquirió la posibilidad de emplear GPU para cómputo científico. La GPU ha mostrado ser eficiente para acelerar cuellos de botella de los cálculos de estructura electrónica [4--6]. Sin embargo, algunos programas optimizados para sistemas de cómputo heterogéneos no son de acceso libre, por ejemplo, TeraChem [7, 8]. El presente trabajo se enfocará en la optimización de un código serial mediante el empleo de tarjetas gráficas. Este código podrá ser accesible a cualquier científico que tenga interés por utilizar o implementar nuevos algoritmos al programa, tal que, a partir de un código sencillo se obtenga un programa robusto adaptable a las necesidades de cada grupo de trabajo y servir como una librería para aquellos que lo requieran.

El primer capítulo es una breve introducción a la teoría de los funcionales de la densidad auxiliar (ADFT, por sus siglas en inglés). En el segundo capítulo se describe la evaluación de las integrales moleculares de repulsión electrónica y las de intercambio-correlación. Estos son procesos costosos, ya que la cantidad de integrales de repulsión y el número de puntos del mallado molecular empleado para obtener el intercambio-correlación son grandes, y son paralelizables porque el procesamiento de cada integral/punto es independiente de las/los otras/otros. El tercer capítulo explica la ley de Amdahl y conceptos técnicos para el uso correcto de la GPU. Los esquemas de programación y los resultados de la validación están en el cuarto capítulo y en el último capítulo están las conclusiones y perspectivas del trabajo.

1

Teoría de los funcionales de la densidad

Los métodos *ab initio* basados en función de onda que incluyen efectos de correlación son computacionalmente costosos ya que dependen de $3N$ coordenadas espaciales (sin considerar el espín electrónico), donde N es el número de electrones del sistema. Por ejemplo, el escalamiento de Møller-Plesset a segundo orden es de $O(N_{bas}^5)$ [9], donde N_{bas} es el número de funciones base utilizadas, y en cúmulos acoplados el escalamiento es mayor que $O(N_{bas}^5)$ [10] dependiendo de las excitaciones consideradas. De manera alterna, la densidad ρ, que únicamente depende de 3 coordenadas espaciales, se puede emplear para obtener información de los sistemas. Idealmente, el escalamiento de un método basado en la densidad es lineal, es decir, $O(N)$, pero la forma explícita del funcional universal que plantean Hohenberg y Kohn (HK) es desconocida [2]. Por lo tanto, existen diferentes aproximaciones para la aplicación de métodos basados en la densidad. Una vertiente del método se enfoca en alcanzar el escalamiento ideal y la otra se concentra en aumentar la exactitud del modelo. Esto no implica que sus objetivos sean mutuamente excluyentes, sino que las prioridades son diferentes. Ambas se basan en la teoría de los funcionales de la densidad (DFT, por sus siglas en inglés).

1.1 Teoremas de Hohenberg y Kohn

En 1964, HK mostraron los dos teoremas fundamentales de la DFT. El primer teorema de HK es: *El potencial externo $v(\boldsymbol{r})$ es un funcional único (hasta una constante arbitraria) de una densidad electrónica $\rho(\boldsymbol{r})$ del estado basal* [2, 11, 12]. Es decir, a partir de la densidad electrónica del estado basal se puede determinar el potencial externo y el número de electrones de un sistema y finalmente conocer el operador Hamiltoniano correspondiente. La demostración del primer teorema de HK consiste en suponer dos potenciales externos distintos, $v(\mathbf{r})$ y $v'(\mathbf{r})$, que se asocian a una misma densidad ρ. Dos potenciales distintos darán lugar a operadores Hamiltoniano, $\hat{H}$ y $\hat{H}'$, y funciones de onda diferentes, Ψ y Ψ'.

A partir de la ecuación de Schrödinger se tiene que $E_0 = \left\langle \Psi \left| \hat{H} \right| \Psi \right\rangle$ y $E_0' = \left\langle \Psi' \left| \hat{H}' \right| \Psi' \right\rangle$. Si se utiliza Ψ' como función de prueba, por el principio variacional se sabe que,

$$E_0 < \left\langle \Psi' \left| \hat{H} \right| \Psi' \right\rangle = \left\langle \Psi' \left| \hat{H}' \right| \Psi' \right\rangle + \left\langle \Psi' \left| \hat{H} - \hat{H}' \right| \Psi' \right\rangle . \tag{1.1}$$

El primer término del lado derecho de la ecuación (1.1) es E_0', por lo tanto,

$$E_0 < E_0' + \left\langle \Psi' \left| \hat{H} - \hat{H}' \right| \Psi' \right\rangle . \tag{1.2}$$

El Hamiltoniano electrónico de un sistema de M núcleos y N electrones se define de la siguiente forma:

$$\hat{H} = -\frac{1}{2} \sum_{i}^{N} \nabla_i^2 + \sum_{i}^{N} \sum_{i>j}^{N} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i}^{N} v(\mathbf{r}_i), \tag{1.3}$$

donde $\mathbf{r}_i$ y $\mathbf{r}_j$ son las coordenadas de los electrones i y j y la notación $|\mathbf{r}_i - \mathbf{r}_j|$ denota la distancia entre dos electrones. El primer término es el operador de la energía cinética, el segundo es el de la repulsión interelectrónica y el último es el de la atracción núcleo-electrón, donde el potencial externo, $v(\mathbf{r})$, es,

$$v(\mathbf{r}) = -\sum_{A}^{M} \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|}. \tag{1.4}$$

Las coordenadas y cargas nucleares son $\mathbf{R}_A$ y Z_A, respectivamente. Las coordenadas electrónicas están denotadas por $\mathbf{r}$ y la notación $|\mathbf{r} - \mathbf{R}_A|$ representa la distancia entre el electrón y el núcleo A. Considerando las ecuaciones (1.3) y (1.4), se puede expresar el segundo término

del lado derecho de la ecuación (1.1) como:

$$E_0 < E_0' + \int \rho(\mathbf{r}) \left(v(\mathbf{r}) - v'(\mathbf{r}) \right) \mathrm{d}\mathbf{r}. \tag{1.5}$$

Ahora, empleando la función Ψ como función de prueba para $\hat{H}'$ se obtiene:

$$E_0' < \left\langle \Psi \left| \hat{H}' \right| \Psi \right\rangle = \left\langle \Psi \left| \hat{H} \right| \Psi \right\rangle + \left\langle \Psi \left| \hat{H}' - \hat{H} \right| \Psi \right\rangle. \tag{1.6}$$

El primer término del lado derecho de la ecuación (1.6) es E_0. Sustituyendo E_0 se tiene:

$$E_0' < E_0 - \left\langle \Psi \left| \hat{H} - \hat{H}' \right| \Psi \right\rangle. \tag{1.7}$$

De manera similiar a la obtención de la ecuación (1.5), el segundo término de la ecuación (1.6) se reduce a:

$$E_0' < E_0 - \int \rho(\mathbf{r}) \left(v(\mathbf{r}) - v'(\mathbf{r}) \right) \mathrm{d}\mathbf{r}. \tag{1.8}$$

Si se comparan las expresiones (1.5) y (1.8), se puede concluir que:

$$E_0 + E_0' < E_0 + E_0', \tag{1.9}$$

lo cual constituye un absurdo, por lo que no pueden existir dos potenciales externos diferentes que se asocien a una misma densidad del estado basal. Entonces, a partir de la densidad del estado basal, ρ, es posible obtener el potencial externo del sistema. Además, la integral de la densidad electrónica proporciona el número de electrones del sistema,

$$\int \rho(\mathbf{r}) \mathrm{d}\mathbf{r} = N, \tag{1.10}$$

donde la densidad es positiva definida en todo el espacio,

$$\rho(\mathbf{r}) \geq 0 \qquad \forall \mathbf{r} \in \mathbb{R}^3. \tag{1.11}$$

Para los sistemas electrónicos, el potencial externo está determinado por la posición y la carga de los núcleos [13]. La ecuación (1.10) junto con el primer teorema de HK implican que la densidad del estado basal determina el Hamiltoniano $\hat{H}$, que a su vez define la función de onda Ψ y por lo tanto, la energía E del sistema. La energía total se puede descomponer en

funcionales indepedientes,

$$E[\rho] = T[\rho] + E_{ee}[\rho] + E_{ne}[\rho]. \tag{1.12}$$

Las expresiones de los primeros dos funcionales, de energía cinética y de repulsión electrón-electrón, son válidos para cualquier sistema y no dependen explícitamente del potencial externo. Por lo tanto, estos dos se agrupan como $F[\rho]$, el funcional universal de HK,

$$F[\rho] = T[\rho] + E_{ee}[\rho] = \left\langle \Psi \left| \hat{T} + \hat{V}_{ee} \right| \Psi \right\rangle, \tag{1.13}$$

tal que la energía total es:

$$E[\rho] = F[\rho] + \int \rho(\mathbf{r}) v(\mathbf{r})\, \mathrm{d}\mathbf{r}. \tag{1.14}$$

El segundo teorema que plantean HK es: *la energía más baja del sistema se obtiene del funcional $E[\rho]$ si y sólo si la densidad electrónica empleada es la del estado basal* [2, 11, 12]. En otras palabras, el principio variacional también aplica para el funcional $E[\rho]$.

$$E[\rho] = \left\langle \Psi[\rho] \left| \hat{H} \right| \Psi[\rho] \right\rangle. \tag{1.15}$$

Considerando la densidad de prueba $\rho'(\mathbf{r})$ con un potencial $v'(\mathbf{r})$, se tiene que $\Psi'[\rho']$ es la función de prueba tal que:

$$\begin{aligned} E'[\rho] = \left\langle \Psi'[\rho'] \left| \hat{H} \right| \Psi'[\rho'] \right\rangle &> E \quad \text{para} \quad \rho'(\mathbf{r}) \neq \rho(\mathbf{r}) \\ &= E \quad \text{para} \quad \rho'(\mathbf{r}) = \rho(\mathbf{r}). \end{aligned} \tag{1.16}$$

Este teorema garantiza la minimización de la energía total del sistema por un método variacional. La DFT planteada por HK (HKDFT) es exacta, pero desafortunadamente, la forma explícita del funcional universal $F[\rho]$ se desconoce. De la HKDFT se derivan dos categorías, DFT libre de orbitales (OFDFT por sus siglas en inglés) y KSDFT (DFT planteada por Kohn y Sham). En OFDFT se utiliza un funcional de la densidad para la energía cinética basado en modelos sencillos, como el de Thomas-Fermi o el de von Weizsäcker [14, 15]. Emily Carter *et al.* emplean una transformada de Fourier rápida (FFT por sus siglas en inglés) para evaluar la energía cinética y obtienen un escalamiento cuasilineal de $O(N_{FFT} \log N_{FFT})$, donde N_{FFT} es el número de puntos del mallado para la FFT [16]. En KSDFT, se introduce un conjunto base para la evaluación de la energía cinética de un sistema no interactuante [17]. Este conjunto base es utilizado para

obtener la interacción electrónica, que da lugar a un escalamiento de $O(N_{bas}^4)$ (similar al de la teoría de Hartree-Fock). A pesar de que KSDFT es más lento que OFDFT, la gama de aplicación de KSDFT es más amplia y generalmente, es más precisa que OFDFT. Debido a que KSDFT es más popular y está implementada en *Parakata* [18], el código que se emplea en el presente trabajo, se tratará únicamente KSDFT en los siguientes capítulos, en los que se mostrará que el escalamiento de KSDFT puede reducirse hasta $O(N_{bas}^2 N_{aux})$ empleando N_{aux} funciones auxiliares [19--22].

1.2 Método de Kohn y Sham

Como se mencionó anteriormente, KSDFT consiste en introducir orbitales a la HKDFT, aproximando la energía cinética del sistema con interacción como aquella de un sistema de electrones no interactuantes e incluyendo la corrección de la energía cinética en el término de la energía de intercambio-correlación E_{xc} [17].

$$E[\rho] = T_s[\{\psi_i\}] + E_{ne}[\rho] + J[\rho] + E_{xc}[\rho], \tag{1.17}$$

donde $T_s[\{\psi_i\}]$, $E_{ne}[\rho]$, $J[\rho]$ y $E_{xc}[\rho]$ son la energía cinética del sistema no interactuante, la interacción núcleo-electrón, la contribución tipo coulómbica a la interacción electrón-electrón y la energía de intercambio-correlación, respectivamente. Estos se definen de la siguiente manera:

$$T_s[\{\psi_i\}] = -\frac{1}{2}\sum_{i}^{\text{occ}} \left\langle \psi_i \left| \nabla^2 \right| \psi_i \right\rangle, \tag{1.18}$$

$$E_{ne}[\rho] = \sum_{A}^{\text{núcleos}} \int \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|}\rho(\mathbf{r})\, \mathrm{d}\mathbf{r}, \tag{1.19}$$

y

$$J[\rho] = \frac{1}{2}\int\int \frac{\rho(\mathbf{r}_1)\rho(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|}\, \mathrm{d}\mathbf{r}_1\, \mathrm{d}\mathbf{r}_2. \tag{1.20}$$

El determinante de Slater que representa la función de onda exacta del sistema no interactuante, Ψ, está conformado por orbitales de Kohn y Sham (KS) ψ_i. Se selecciona un potencial monoelectrónico del sistema no interactuante de manera tal que la suma de los cuadrados

de los orbitales de KS resultantes sea igual a la densidad electrónica $\rho(\mathbf{r})$ del estado basal del sistema con interacción,

$$\rho(\mathbf{r}) = \sum_{i}^{\text{occ}} |\psi_i|^2 . \tag{1.21}$$

Los orbitales de KS son ortonormales. Bajo esta restricción, las ecuaciones de KS se pueden obtener mediante la minimización de la energía:

$$\left(-\frac{1}{2}\nabla^2 + \sum_{A}^{\text{núcleos}} \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} + \int \frac{\rho(\mathbf{r}_2)}{|\mathbf{r} - \mathbf{r}_2|}\, \mathrm{d}\mathbf{r}_2 + v_{xc}[\rho] \right) \psi_i(\mathbf{r}) = \varepsilon_i \psi_i(\mathbf{r}) \qquad \forall\, i \tag{1.22}$$

donde v_{xc} es la derivada funcional de la energía de intercambio-correlación,

$$v_{xc}[\rho] \equiv \frac{\delta E_{xc}[\rho]}{\delta \rho(\mathbf{r})}. \tag{1.23}$$

Observando las expresiones previas, en especial la ecuación (1.20), es claro que introducir un conjunto base para evaluar los funcionales $T_s[\{\psi_i\}]$, $E_{ne}[\rho]$, $J[\rho]$ y $E_{xc}[\rho]$ para un sistema no interactuante, aleja el método del escalamiento lineal de HKDFT. Aunque el costo computacional no es óptimo, esta aproximación ha mostrado ser muy útil para sistemas químicos variados. La KSDFT es exacta, hasta que se introduce una aproximación al funcional E_{xc}. Un paso fundamental para utilizar este método es encontrar buenas aproximaciones para E_{xc}. Estos funcionales deben cumplir ciertas restricciones como comportamiento asintótico, escalamiento, etc. [23] De acuerdo a la escalera de Jacob [24] planteada por John Perdew, en el primer escalón está la aproximación local de la densidad (LDA por sus siglas en inglés) [25]. En el siguiente peldaño se encuentran los funcionales de gradiente generalizado (GGA) [26, 27], que consideran el gradiente de la densidad local. Más adelante están los funcionales meta-GGA, que incluyen un término de energía cinética. Posteriormente están los funcionales hiper-GGA, que son los más cercanos al "cielo" planteado por Perdew, es decir, el funcional universal exacto. Cabe mencionar que avanzar en la escalera no garantiza una mejor descripción del sistema. Conforme se avanza en la escalera de Jacob, las expresiones matemáticas de estos funcionales se complican, por lo que se usan métodos numéricos para su integración.

1.2.1 Combinación lineal de orbitales de tipo gaussiano

Los orbitales moleculares de KS pueden ser construidos por una combinación lineal de orbitales atómicos de Slater, que tienen un comportamiento asintótico correcto y de cúspide en los núcleos, pero generalmente se emplean funciones gaussianas para representar los orbitales tipo Slater [28]. Las funciones gaussianas facilitan la integración, a pesar de que se requieren varias funciones para representar a un orbital de Slater. Entonces, los orbitales moleculares de KS se obtienen mediante una combinación lineal de orbitales atómicos, o funciones base contraídas, ϕ_a,

$$\psi_i(\mathbf{r}) = \sum_a c_a^i \phi_a(\mathbf{r}), \tag{1.24}$$

donde ϕ_a se puede expandir en funciones gaussianas primitivas centradas en un mismo núcleo, tal que,

$$\phi_a(\mathbf{r}) = S \sum_{k=1}^{K_a} D_k^a \varphi_k^a(\mathbf{r}), \tag{1.25}$$

donde cada función gaussiana primitiva es,

$$\varphi_k^a(\mathbf{r}) = (x - A_x)^{n_x} (y - A_y)^{n_y} (z - A_z)^{n_z} e^{-\alpha_k (\mathbf{r}-\mathbf{A})^2}. \tag{1.26}$$

Aquí, **A** son las coordenadas del núcleo en el que está centrada la función. El vector $\mathbf{n} = (n_x, n_y, n_z)$ representa el número cuántico angular de la función en cada coordenada y el exponente de la gaussiana es α_k. La función contraída se puede expresar como:

$$\phi_a(\mathbf{r}) = S (x - A_x)^{n_x} (y - A_y)^{n_y} (z - A_z)^{n_z} \sum_{k=1}^{K_a} D_k^a e^{-\alpha_k^a (\mathbf{r}-\mathbf{A})^2}, \tag{1.27}$$

donde K_a es el grado de contracción y cada función primitiva tiene un coeficiente y un exponente, D_k^a y α_k^a, respectivamente. Por último, S es el factor de normalización. El número de funciones base se representa como N_{bas}. La densidad para una capa cerrada es:

$$\rho(\mathbf{r}) = \sum_{a,b} P_{ab}\, \phi_a(\mathbf{r}) \phi_b(\mathbf{r}), \tag{1.28}$$

donde **P** es la matriz de densidad cuyos elementos están dados por:

$$P_{ab} = 2\sum_{i}^{\text{occ}} c_a^i c_b^i. \tag{1.29}$$

De esta manera, la energía del sistema se puede escribir como:

$$E_{SCF} = \sum_{a,b} P_{ab}\, H_{ab}^{\text{core}} + \frac{1}{2}\sum_{a,b}\sum_{c,d} P_{ab}\, P_{cd}\, \langle \phi_a \phi_b || \phi_c \phi_d \rangle + E_{xc}[\rho], \tag{1.30}$$

donde H_{ab}^{core} es

$$H_{ab}^{\text{core}} = -\frac{1}{2}\left\langle \phi_a \left| \nabla^2 \right| \phi_b \right\rangle - \sum_A \left\langle \phi_a \left| \frac{Z_A}{|\mathbf{r} - \mathbf{A}|} \right| \phi_b \right\rangle, \tag{1.31}$$

y cada integral de repulsión es

$$\langle \phi_a \phi_b || \phi_c \phi_d \rangle = \int\int \frac{\phi_a(\mathbf{r}_1)\phi_b(\mathbf{r}_1)\phi_c(\mathbf{r}_2)\phi_d(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|}\, \mathrm{d}\mathbf{r}_1\, \mathrm{d}\mathbf{r}_2, \tag{1.32}$$

donde $||$ denota el operador $\frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|}$. Las integrales de repulsión electrónica (ERI por sus siglas en inglés) son de cuatro centros y se requieren evaluar N_{bas}^4 de ellas. La energía de intercambio-correlación se evalúa sobre una malla de integración y su escalamiento es de $O(N_{bas}^2 G)$, donde G es el número de puntos en la malla. La evaluación de las integrales de repulsión electrónica y la energía de intercambio-correlación son computacionalmente costosos y se han desarrollado varias aproximaciones para hacerlos más eficientes.

1.2.2 Teoría de los funcionales de la densidad auxiliar

El desarrollo de algoritmos eficientes para la evaluación de las integrales de repulsión electrónica ha sido de gran interés, ya que es uno de los cuellos de botella de los cálculos de estructura electrónica. En un intento para recuperar el escalamiento lineal de HKDFT utilizando KSDFT, se utiliza el ajuste variacional del potencial de Coulomb mediante una densidad auxiliar [19--21, 29]. El escalamiento de $O(N_{bas}^4)$ se reduce a $O(N_{bas}^2 N_{aux})$, donde N_{aux} es el número de funciones de base gaussianas que generan la densidad auxiliar. En general, N_{aux} es mayor que N_{bas}, pero la evaluación de las integrales es más sencilla [30].

El segundo cuello de botella importante de un cálculo DFT, después de la evaluación de las ERI, es la integración del potencial de intercambio-correlación. El escalamiento para la obtención de la densidad auxiliar es lineal porque la densidad auxiliar es una combinación lineal de funciones gaussianas, a diferencia de los métodos convencionales que requieren de los productos de las funciones base y tienen un escalamiento cuadrático. Además se pueden utilizar funciones gaussianas hermitianas para poder aprovechar las relaciones de recurrencia de los polinomios de Hermite y reducir el número de funciones exponenciales a evaluar [30--32].

1.2.3 Ajuste variacional del potencial de Coulomb

El ajuste variacional del potencial de Coulomb consiste en introducir una densidad auxiliar normalizada y positiva definida $\tilde{\rho}(\mathbf{r})$:

$$\tilde{\rho}(\mathbf{r}) = \sum_{m=1}^{N_{aux}} x_m \bar{k}_m(\mathbf{r}). \tag{1.33}$$

Los coeficientes de expansión son x_m, mientras que $\bar{k}_m$ son las funciones gaussianas auxiliares. En la práctica, no se tiene garantía de que la densidad auxiliar siempre sea positiva definida, pero el error que genera es generalmente despreciable [30]. El objetivo de esta aproximación es minimizar el error de la siguiente expresión:

$$\varepsilon_{coul} = \frac{1}{2} \int\int \frac{[\rho(\mathbf{r}_1) - \tilde{\rho}(\mathbf{r}_1)][\rho(\mathbf{r}_2) - \tilde{\rho}(\mathbf{r}_2)]}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2 \geq 0, \tag{1.34}$$

donde ρ es la densidad real del sistema. En notación de Dirac, se tiene,

$$\varepsilon_{coul} = \frac{1}{2} \langle \rho - \tilde{\rho} || \rho - \tilde{\rho} \rangle = \frac{1}{2} \langle \rho || \rho \rangle - \langle \rho || \tilde{\rho} \rangle + \frac{1}{2} \langle \tilde{\rho} || \tilde{\rho} \rangle . \tag{1.35}$$

Sustituyendo la expresión de la densidad, ecuación (1.28), y la densidad auxiliar, ecuación (1.33), en el segundo y tercer término de la ecuación (1.35),

$$\varepsilon_{coul} = \frac{1}{2} \langle \rho || \rho \rangle - \sum_m x_m \sum_{a,b} P_{ab} \left\langle \phi_a \phi_b || \bar{k}_m \right\rangle + \frac{1}{2} \sum_{m,l} x_m \, x_l \left\langle \bar{k}_m || \bar{k}_l \right\rangle \geq 0. \tag{1.36}$$

El error es positivo definido y podemos despejar $\langle \rho || \rho \rangle$,

$$\frac{1}{2} \langle \rho || \rho \rangle \geq \sum_m x_m \sum_{a,b} P_{ab} \left\langle \phi_a \phi_b || \bar{k}_m \right\rangle - \frac{1}{2} \sum_{m,l} x_m \, x_l \left\langle \bar{k}_m || \bar{k}_l \right\rangle . \tag{1.37}$$

Si sustituimos en la expresión (1.30) de la energía, se tiene que,

$$E_{SCF} \approx \sum_{a,b} P_{ab} H_{ab}^{\text{core}} + \sum_{m} x_m \sum_{a,b} P_{ab} \left\langle \phi_a \phi_b || \bar{k}_m \right\rangle - \frac{1}{2} \sum_{m,l} x_m x_l \left\langle \bar{k}_m || \bar{k}_l \right\rangle + E_{xc}[\rho]. \quad (1.38)$$

Los coeficientes x_m se obtienen minimizando ε_{coul} a **P** constante,

$$\left(\frac{\partial \varepsilon_{coul}}{\partial x_n} \right)_{\mathbf{P}} = -\sum_{a,b} P_{ab} \left\langle \phi_a \phi_b || \bar{k}_n \right\rangle + \sum_{l} x_l \left\langle \bar{k}_l || \bar{k}_n \right\rangle = 0 \qquad \forall \quad n. \quad (1.39)$$

Si se denomina a **G** como la matriz de Coulomb,

$$\mathbf{G} = \begin{pmatrix} \left\langle \bar{k}_1 || \bar{k}_1 \right\rangle & \cdots & \left\langle \bar{k}_1 || \bar{k}_m \right\rangle \\ \vdots & \ddots & \vdots \\ \left\langle \bar{k}_m || \bar{k}_1 \right\rangle & \cdots & \left\langle \bar{k}_m || \bar{k}_m \right\rangle \end{pmatrix}, \quad (1.40)$$

y a **J** como el vector de Coulomb,

$$\mathbf{J} = \begin{pmatrix} \sum_{a,b} P_{ab} \left\langle \phi_a \phi_b || \bar{k}_1 \right\rangle \\ \vdots \\ \sum_{a,b} P_{ab} \left\langle \phi_a \phi_b || \bar{k}_m \right\rangle \end{pmatrix} = \begin{pmatrix} \left\langle \rho || \bar{k}_1 \right\rangle \\ \vdots \\ \left\langle \rho || \bar{k}_m \right\rangle \end{pmatrix}, \quad (1.41)$$

el conjunto de ecuaciones (1.39) puede expresarse de forma matricial como,

$$\mathbf{Gx} = \mathbf{J}, \quad (1.42)$$

donde **x** es el vector de coeficientes ajustados y se puede resolver como,

$$\mathbf{x} = \mathbf{G}^{-1} \mathbf{J}. \quad (1.43)$$

Originalmente, los coeficientes de las funciones auxiliares para la evaluación del potencial de intercambio-correlación se ajustaban por mínimos cuadrados en un mallado [20]. Debido a que el método no es variacional, los gradientes aproximados resultan en geometrías incorrectas [30]. Esto se puede evitar si se reutilizan los coeficientes obtenidos previamente en el ajuste del potencial de Coulomb para el potencial de intercambio-correlación [33].

Entonces, la energía se puede expresar como:

$$E_{SCF} \approx \sum_{a,b} P_{ab} H_{ab}^{\text{core}} + \sum_{a,b} \sum_{m} P_{ab} \langle \phi_a \phi_b || \bar{k}_m \rangle x_m - \frac{1}{2} \sum_{m,l} x_m x_l \langle \bar{k} || \bar{l} \rangle + E_{xc}[\tilde{\rho}]. \quad (1.44)$$

A esta aproximación se le denomina teoría de los funcionales de la densidad auxiliar (ADFT por sus siglas en inglés). Cada elemento de la matriz de potencial de KS se obtiene de la derivada parcial de E_{SCF} con respecto a un elemento de la matriz de densidad, esto es,

$$K_{ab} = H_{ab}^{\text{core}} + \sum_{m} \langle \phi_a \phi_b || \bar{k}_m \rangle x_m + \frac{\partial E_{xc}[\tilde{\rho}]}{\partial P_{ab}}, \quad (1.45)$$

donde la derivada parcial de la energía de intercambio-correlación con respecto a un elemento de la matriz de densidad es:

$$\frac{\partial E_{xc}[\tilde{\rho}]}{\partial P_{ab}} = \int \frac{\delta E_{xc}[\tilde{\rho}]}{\delta \tilde{\rho}(\mathbf{r})} \frac{\partial \tilde{\rho}(\mathbf{r})}{\partial P_{ab}} d\mathbf{r} = \sum_{m} \frac{\partial x_m}{\partial P_{ab}} \int v_{xc}[\tilde{\rho}] \bar{k}_m(\mathbf{r}) d\mathbf{r}, \quad (1.46)$$

tal que el potencial de intercambio-correlación es la derivada funcional de la energía de intercambio-correlación,

$$v_{xc}[\tilde{\rho}] \equiv \frac{\delta E_{xc}[\tilde{\rho}]}{\delta \tilde{\rho}(\mathbf{r})}. \quad (1.47)$$

De la ecuación (1.39), se puede obtener la siguiente expresión:

$$\frac{\partial x_m}{\partial P_{ab}} = \sum_{l} G_{lm}^{-1} \langle \bar{k}_l || \phi_a \phi_b \rangle. \quad (1.48)$$

Sustituyendo en la ecuación (1.46), se tiene que

$$\frac{\partial E_{xc}[\tilde{\rho}]}{\partial P_{ab}} = \sum_{m,l} G_{lm}^{-1} \langle \bar{k}_l || \phi_a \phi_b \rangle \langle \bar{k}_m || v_{xc} \rangle. \quad (1.49)$$

Se definen como coeficientes del ajuste del potencial de intercambio-correlación como z_l,

$$z_l = \sum_{m} G_{lm}^{-1} \langle \bar{k}_m || v_{xc} \rangle. \quad (1.50)$$

Así, los elementos de la matriz de KS pueden escribirse como:

$$K_{ab} = H_{ab}^{\text{core}} + \sum_{m} \langle \phi_a \phi_b || \bar{k}_m \rangle (x_m + z_m). \tag{1.51}$$

La energía se puede expresar de la siguiente manera:

$$E_{SCF} \approx \sum_{a,b} P_{ab} K_{ab} - \frac{1}{2} \sum_{m,l} x_m x_l \langle \bar{k} || \bar{l} \rangle. \tag{1.52}$$

En un SCF, los parámetros variacionales son los coeficientes de la densidad auxiliar, que dependen de los coeficientes de expansión de los orbitales moleculares. De manera más detallada, el SCF consiste de los siguientes pasos [32, 34, 35]:

1. Obtener datos de la molécula de interés (los conjuntos de coordenadas nucleares $\{\mathbf{R_A}\}$, números atómicos $\{Z_A\}$ y número de electrones N) y especificar el conjunto base $\{\phi_a\}$.
2. Calcular las integrales moleculares S_{ab}, H_{ab}^{core} (ecuación (1.31)), $(\phi_a \phi_b || \bar{k}_m)$ (ecuación (1.32)).
3. Diagonalizar la matriz de traslape **S** y obtener la matriz de transformación **X** que ortogonaliza la base.
4. Estimar los elementos de la matriz de densidad **P** (ecuación (1.29)).
5. Obtener el vector de Coulomb **J** (ecuación (1.41)).
6. Calcular la matriz de Coulomb **G** (ecuación (1.40)) y obtener su inversa.
7. Obtener los coeficientes de expansión **x** (ecuación (1.43)).
8. Obtener los coeficientes **z** (ecuación (1.50)).
9. Calcular la matriz de potencial **K** (ecuación (1.51)).
10. Calcular la energía del sistema E_{SCF} (ecuación (1.52)).
11. Si el error entre la energía de la iteración con respecto a la interación anterior es menor al error permitido, entonces, el procedimiento convirgió y si no es menor, entonces, se repite el procedimiento desde el paso 5 con la nueva matriz de densidad **P**, que se obtiene de la matriz **K**.

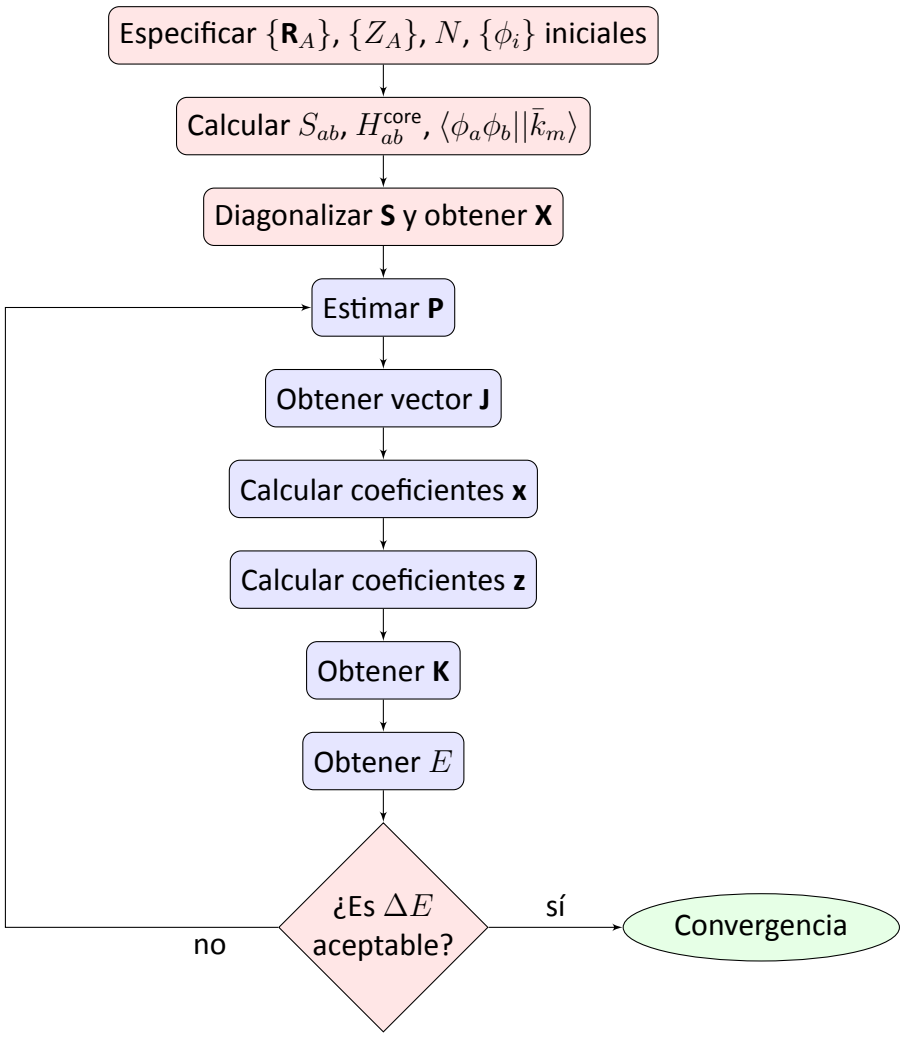


Figura 1.1: Esquema de trabajo de un SCF basado en ADFT.

En la Figura 1.1 se muestran los pasos del procedimiento SCF. Se debe notar que las integrales de tres centros se evalúan antes que se inicie la iteración. Siguiendo el método convencional, las ERI se guardan en la memoria de acceso aleatorio (RAM, por sus siglas en inglés). El tamaño del sistema de estudio está limitado por la cantidad de RAM disponible. Dichas integrales se utilizan en cada iteración dos veces. La primera sirve para evaluar la matriz de KS y la segunda para la obtención del vector de Coulomb **J** [36].

2

Integrales moleculares

2.1 Integrales de repulsión electrónica

Las integrales de repulsión electrónica se pueden evaluar numéricamente con un mallado molecular o de manera analítica. En este capítulo se describe detalladamente la evaluación analítica de las integrales mediante relaciones de recurrencia.

Cada una de las N_{bas}^4 ERI en un sistema de N electrones, es una integral sobre seis dimensiones. En esta sección, es importante indicar el momento angular de cada centro, por lo que se utilizarán **a**, **b**, **c** y **d** para representar cada centro y su momento angular correspondiente.

$$\langle \mathbf{ab} \,||\, \mathbf{cd} \rangle = \int\int \frac{\phi_a(\mathbf{r}_1)\phi_b(\mathbf{r}_1)\phi_c(\mathbf{r}_2)\phi_d(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2, \tag{2.1}$$

Además, cada orbital atómico ϕ_a es una combinación lineal de funciones gaussianas (primitivas), por lo tanto, esta integral es una suma sobre cuatro índices de las ERI en primitivas,

$$\langle \mathbf{ab} \,||\, \mathbf{cd} \rangle = \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{m=1}^{M} \sum_{n=1}^{N} D_k^a D_l^b D_m^c D_n^d [\mathbf{a}_k \mathbf{b}_l \,||\, \mathbf{c}_m \mathbf{d}_n], \tag{2.2}$$

donde cada ERI en primitivas es una integral sobre $\mathbf{r}_1$ y $\mathbf{r}_2$,

$$[\mathbf{a}_k \mathbf{b}_l \,||\, \mathbf{c}_m \mathbf{d}_n] = \int \int \frac{\varphi_k^a(\mathbf{r}_1) \varphi_l^b(\mathbf{r}_1) \varphi_m^c(\mathbf{r}_2) \varphi_n^d(\mathbf{r}_2)}{|\mathbf{r}_1 - \mathbf{r}_2|} d\mathbf{r}_1 d\mathbf{r}_2. \tag{2.3}$$

Los caracteres $<>$ indican que la integración es sobre funciones contraídas y los corchetes $[\,]$ denotan que la ERI es sobre primitivas. Por simplicidad, los subíndices k, l, m y n se omitirán en el resto del escrito. En la Figura 2.1 se observa que para un sistema con momento angular hasta p genera seis tipos de integrales diferentes. Las integrales [**ab** || **cd**], [**ba** || **cd**], [**ba** || **dc**] y [**ab** || **dc**] son iguales porque las funciones son reales. Esto implica que sólo hay $\left(\frac{N_{bas}[N_{bas}+1]}{2}\right) \times \left(\frac{N_{bas}[N_{bas}+1]}{2} + 1\right)/2$ integrales únicas.

2.1.1 Relaciones de recurrencia para ERI de cuatro centros

En 1988, Obara y Saika [37] mostraron que una integral de cuatro centros (**A**, **B**, **C** y **D**) con momentos angulares (**a**, **b**, **c** y **d**) y exponentes arbitrarios (α, β, γ y δ), se puede reducir a integrales de tipo $[ss \,||\, ss]^{(m)}$ mediante la siguiente relación de recurrencia,

$$\begin{aligned}
[(\mathbf{a} + \mathbf{1}_i)\, \mathbf{b} \,||\, \mathbf{cd}]^{(m)} = {} & (P_i - A_i)\, [\mathbf{ab} \,||\, \mathbf{cd}]^{(m)} + (W_i - P_i)\, [\mathbf{ab} \,||\, \mathbf{cd}]^{(m+1)} \\
& + \frac{a_i}{2\zeta} \left([(\mathbf{a} - \mathbf{1}_i)\, \mathbf{b} \,||\, \mathbf{cd}]^{(m)} - \frac{\eta}{\xi} [(\mathbf{a} - \mathbf{1}_i)\, \mathbf{b} \,||\, \mathbf{cd}]^{(m+1)} \right) \\
& + \frac{b_i}{2\zeta} \left([\mathbf{a}\, (\mathbf{b} - \mathbf{1}_i) \,||\, \mathbf{cd}]^{(m)} - \frac{\eta}{\xi} [\mathbf{a}\, (\mathbf{b} - \mathbf{1}_i) \,||\, \mathbf{cd}]^{(m+1)} \right) \\
& + \frac{c_i}{2\xi} [\mathbf{ab} \,||\, (\mathbf{c} - \mathbf{1}_i)\, \mathbf{d}]^{(m+1)} - \frac{d_i}{2\xi} [\mathbf{ab} \,||\, \mathbf{c}\, (\mathbf{d} - \mathbf{1}_i)]^{(m+1)}\,.
\end{aligned} \tag{2.4}$$

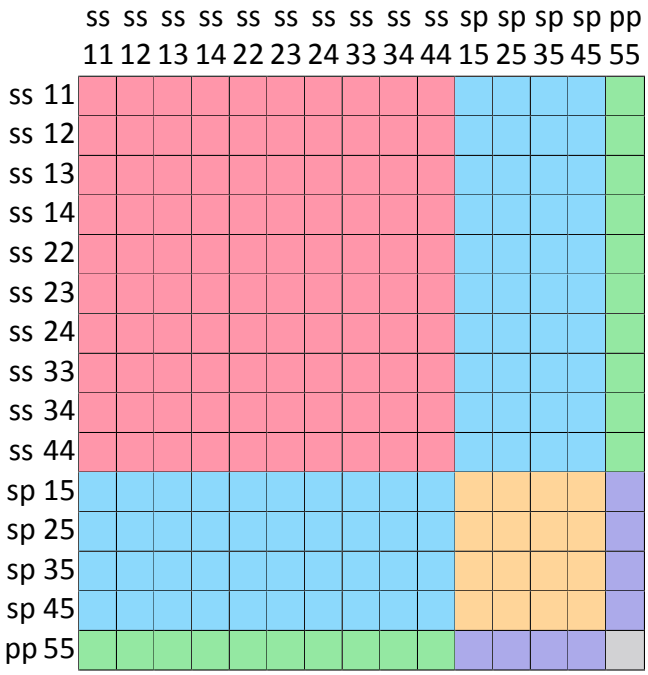


Figura 2.1: Representación gráfica de las integrales de cuatro centros requeridas para un cálculo DFT de un sistema con cuatro funciones s y una función p, donde cada color representa un tipo de integral de cuatro centros. Cada cuadrado representa un conjunto de integrales primitivas que corresponden al *shell pair* del bra (fila) y del ket (columna). Las letras y los números al inicio de cada fila o columna indica el momento angular de cada centro y el número de la función respectivamente.

Los enteros a_i, b_i, c_i y d_i indican el momento angular en la orientación i del centro **a**, **b**, **c** y **d**, respectivamente. **P**, **Q** y **W** son los nuevos centros y ζ, η y ξ son los coeficientes correspondientes que resultan de aplicar el teorema de producto gaussiano (GPT, por sus siglas en inglés), que se presenta en el Apéndice A, y se expresan de la siguiente forma:

$$\begin{aligned} \zeta &= \alpha + \beta \\ \eta &= \gamma + \delta \\ \xi &= \zeta + \eta \end{aligned} \qquad \begin{aligned} P_i &= \frac{\alpha A_i + \beta B_i}{\zeta} \\ Q_i &= \frac{\gamma C_i + \delta D_i}{\eta} \\ W_i &= \frac{\zeta P_i + \eta Q_i}{\xi} \end{aligned} \tag{2.5}$$

Únicamente nos interesan las ERI reales de orden cero ($m = 0$) y las ERI de orden mayor son ERI auxiliares para la evaluación de ERI con mayor momento angular. La regla de recurrencia

se aplica sucesivamente hasta que todas las integrales tengan centros con momento angular s. Por ejemplo, si se desea calcular $[\mathbf{sp}_i \,||\, \mathbf{p}_j\mathbf{s}]$, se realiza la recurrencia sobre uno de los centros que tiene momento angular p. La ERI $[\mathbf{sp}_i \,||\, \mathbf{p}_j\mathbf{s}]$ se puede expresar como $[\mathbf{s}\,(\mathbf{s}+\mathbf{1}_i) \,||\, \mathbf{p}_j\mathbf{s}]$. Al aplicar la relación de recurrencia sobre el segundo centro, se obtiene que

$$[\mathbf{s}\,(\mathbf{s}+\mathbf{1}_i) \,||\, \mathbf{p}_j\mathbf{s}]^{(0)} = (P_i - B_i)\,[\mathbf{ss} \,||\, \mathbf{p}_j\mathbf{s}]^{(0)} + (W_i - P_i)\,[\mathbf{ss} \,||\, \mathbf{p}_j\mathbf{s}]^{(1)} + \frac{\delta_{ij}}{2\xi}\,[\mathbf{ss} \,||\, \mathbf{ss}]^{(1)}\,. \tag{2.6}$$

Únicamente quedan tres términos porque no existe momento angular menor a s. Finalmente, a cada uno de los dos primeros términos se les aplica otra vez la relación para obtener

$$[\mathbf{ss} \,||\, (\mathbf{s}+\mathbf{1}_j)\,\mathbf{s}]^{(m)} = (Q_i - C_i)\,[\mathbf{ss} \,||\, \mathbf{ss}]^{(m)} + (W_i - Q_i)\,[\mathbf{ss} \,||\, \mathbf{ss}]^{(m+1)}\,. \tag{2.7}$$

Sustituyendo la expresión (2.7) en la ecuación (2.6),

$$\begin{aligned}[\mathbf{s}\,(\mathbf{s}+\mathbf{1}_i) \,||\, \mathbf{p}_j\mathbf{s}]^{(0)} = {} & (P_i - B_i)\,\{(Q_i - C_i)\,[\mathbf{ss} \,||\, \mathbf{ss}]^{(0)} + (W_i - Q_i)\,[\mathbf{ss} \,||\, \mathbf{ss}]^{(1)}\} \\ & + (W_i - P_i)\,\{(Q_i - C_i)\,[\mathbf{ss} \,||\, \mathbf{ss}]^{(1)} + (W_i - Q_i)\,[\mathbf{ss} \,||\, \mathbf{ss}]^{(2)}\} \\ & + \frac{\delta_{ij}}{2\xi}\,[\mathbf{ss} \,||\, \mathbf{ss}]^{(1)}\,.\end{aligned} \tag{2.8}$$

Las ERI de tipo $[ss \,||\, ss]$ se evalúan de la siguiente manera,

$$[ss \,||\, ss]^{(m)} = \frac{2\pi^{5/2} K_{AB} K_{CD} F_m(T)}{\zeta\eta\sqrt{\xi}}, \tag{2.9}$$

donde $F_m(T) = \int_0^1 t^{2m} \exp(-Tt^2)\,dt$ es la función gamma incompleta y T, K_{AB} y K_{CD} son constantes,

$$T = \frac{\zeta\eta}{\xi}|\mathbf{P} - \mathbf{Q}|^2, \tag{2.10}$$

$$K_{AB} = \exp\left[-\frac{\alpha\beta}{\zeta}|\mathbf{A} - \mathbf{B}|^2\right], \tag{2.11}$$

$$K_{CD} = \exp\left[-\frac{\gamma\delta}{\eta}|\mathbf{C} - \mathbf{D}|^2\right]. \tag{2.12}$$

La aproximación que fue empleada en este trabajo para la evaluación de la función gamma incompleta está descrita en el artículo de McMurchie-Davidson [38]. Un caso particular sucede cuando $m = 0$, ya que $F_0(T)$ es la función de error

$$F_0(T) = \int_0^1 \exp(-Tt^2)\, dt. \tag{2.13}$$

Las relaciones de recurrencia de Obara y Saika se denominan también reglas de recurrencia vertical (VRR), ya que el momento angular total de los centros se reduce gradualmente. Otra regla de recurrencia importante es la de Head-Gordon y Pople [39]. Estas últimas se denominan relaciones de recurrencia horizontal (HRR) porque la suma de los momentos angulares de los cuatro centros no cambia, sino que se acumula en dos centros. La demostración de la HRR es sencilla, ya que sólo es la resta entre la VRR para $[\mathbf{a}\,(\mathbf{b}+\mathbf{1}_i)\,||\,\mathbf{cd}]$ y $[(\mathbf{a}+\mathbf{1}_i)\,\mathbf{b}\,||\,\mathbf{cd}i]$, resultando en

$$[\mathbf{a}\,(\mathbf{b}+\mathbf{1}_i)\,||\,\mathbf{cd}] = [(\mathbf{a}+\mathbf{1}_i)\,\mathbf{b}\,||\,\mathbf{cd}] + (A_i - B_i)\,[\mathbf{ab}\,||\,\mathbf{cd}]\,. \tag{2.14}$$

La ventaja de emplear las HRR es que acumulan el momento angular en el primer y tercer centro, reduciendo el número de ERI de tipo $[ss\,||\,ss]^{(m)}$ necesario para evaluar integrales con momento angular alto. Además se pueden aprovechar para obtener ERI con el mismo momento angular total.

El potencial de Coulomb, V_{ab}^C, se obtiene de derivar el segundo término de la ecuación (1.30) con respecto a P_{ab}. Este se expresa, en función de ERI primitivas, como:

$$V_{ab}^C = \sum_{cd}^{K}\sum_{k=1}^{L}\sum_{l=1}^{M}\sum_{m=1}^{N}\sum_{n=1} P_{cd} D_k^a D_l^b D_m^c D_n^d\,[\mathbf{a}_k\mathbf{b}_l\,||\,\mathbf{c}_m\mathbf{d}_n]\,. \tag{2.15}$$

2.1.2 Relaciones de recurrencia para ERI de tres centros

Si se realiza el ajuste variacional (subsección 1.2.3) del potencial de Coulomb, se calculan integrales de tres y dos centros. La Figura 2.2 muestra seis tipos de integrales diferentes. La simetría entre $\left[\mathbf{ab}\,||\,\bar{k}_m\right]$ y $\left[\mathbf{ba}\,||\,\bar{k}_m\right]$ se mantiene, pero ahora es necesario evaluar $(N_{bas}[N_{bas}+1]/2)N_{aux}$ integrales. Generalmente N_{aux} es de tres a cinco veces N_{bas} [30]. Esto significa que si N_{bas} es pequeño, se requieren más integrales con la densidad auxiliar y si N_{bas} es grande, se reduce el número de integrales. Otra ventaja de emplear una densidad auxiliar es la simplificación de su evaluación.

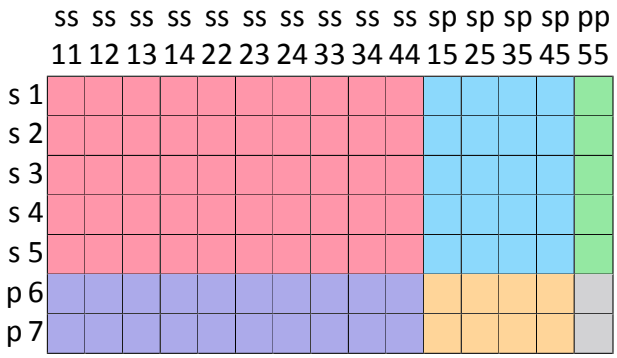


Figura 2.2: Representación gráfica de las integrales requeridas para un cálculo DFT donde cada color representa un tipo de integral de tres centros (las columnas corresponden al bra y las filas al ket). Cada cuadrado representa un conjunto de integrales primitivas que corresponden al *shell pair* del bra y el *shell* auxiliar del ket.

La evaluación de las integrales de tres centros requiere más tiempo que las de dos, por lo que se demuestra únicamente las relaciones de recurrencia de las primeras. La VRR para tres centros se puede derivar fácilmente de la ecuación (2.4) considerando el último centro con un exponente de 0, dando lugar a la siguiente expresión,

$$\begin{aligned}
\left[(\mathbf{a}+\mathbf{1}_i)\,\mathbf{b}\,||\,\mathbf{c}\right]^{(m)} &= (P_i - A_i)\left[\mathbf{ab}\,||\,\mathbf{c}\right]^{(m)} + (U_i - P_i)\left[\mathbf{ab}\,||\,\mathbf{c}\right]^{(m+1)} \\
&+ \frac{a_i}{2\zeta}\left(\left[(\mathbf{a}-\mathbf{1}_i)\,\mathbf{b}\,||\,\mathbf{c}\right]^{(m)} - \frac{\gamma}{\zeta+\gamma}\left[(\mathbf{a}-\mathbf{1}_i)\,\mathbf{b}\,||\,\mathbf{c}\right]^{(m+1)}\right) \\
&+ \frac{b_i}{2\zeta}\left(\left[\mathbf{a}\,(\mathbf{b}-\mathbf{1}_i)\,||\,\mathbf{c}\right]^{(m)} - \frac{\gamma}{\zeta+\gamma}\left[\mathbf{a}(\mathbf{b}-\mathbf{1}_i)\,||\,\mathbf{c}\right]^{(m+1)}\right) \\
&+ \frac{c_i}{2(\zeta+\gamma)}\left[\mathbf{ab}\,||\,(\mathbf{c}-\mathbf{1}_i)\right]^{(m+1)}. \qquad (2.16)
\end{aligned}$$

De manera semejante a **W**, el centro **U** es

$$U_i = \frac{\zeta P_i + \gamma C_i}{\zeta+\gamma}. \qquad (2.17)$$

Ahora, las ERI se reducen a integrales de tipo $\left[ss\,||\,s\right]^{(m)}$,

$$\left[ss\,||\,s\right]^{(m)} = \frac{2\pi^{5/2} K_{AB} F_m(T)}{\zeta\delta\sqrt{\zeta+\gamma}}. \qquad (2.18)$$

Tabla 2.1: Relaciones de recurrencia vertical para la evaluación de ERI de tres centros. El símbolo δ_{ij} denota la función delta de Kronecker.

$$
\begin{aligned}
[ss\,||\,s]^{(0)} &= 2\pi^{5/2}\zeta^{-1}\delta^{-1}(\zeta+\gamma)^{-1/2}K_{AB}F_m(T)\\
[ss\,||\,p_i]^{(0)} &= (U_i - C_i)\,[ss\,||\,s]^{(1)}\\
[p_i s\,||\,s]^{(0)} &= (P_i - A_i)\,[ss\,||\,s]^{(0)} + (U_i - P_i)\,[ss\,||\,s]^{(1)}\\
[p_i s\,||\,p_j]^{(0)} &= (P_i - A_i)\,[ss\,||\,p_j]^{(0)} + (U_i - P_i)\,[ss\,||\,p_j]^{(1)} + \delta_{ij}2^{-1}(\zeta+\gamma)^{-1}\,[ss\,||\,s]^{(1)}\\
[p_i p_j\,||\,s]^{(0)} &= (P_j - A_j)\,[p_i s\,||\,s]^{(0)} + (U_j - P_j)\,[p_i s\,||\,s]^{(1)}\\
&\quad +\delta_{ij}2^{-1}\zeta^{-1}\left([ss\,||\,s]^{(0)} - \gamma(\zeta+\gamma)^{-1}\,[ss\,||\,s]^{(1)}\right)\\
[p_i p_j\,||\,p_k]^{(0)} &= (P_j - A_j)\,[p_i s\,||\,s]^{(0)} + (U_j - P_j)\,[p_i s\,||\,s]^{(1)} + \delta_{ij}2^{-1}\zeta^{-1}\left([ss\,||\,p_k]^{(0)}\right.\\
&\quad \left.-\gamma(\zeta+\gamma)^{-1}\,[ss\,||\,p_k]^{(1)}\right) + \delta_{ik}2^{-1}(\zeta+\gamma)^{-1}\,[p_i s\,||\,s]^{(1)}
\end{aligned}
$$

Las HRR, ecuación (2.14), también se pueden aplicar a las integrales de tres centros, pero únicamente se aplica sobre el bra,

$$[\mathbf{a}\,(\mathbf{b}+\mathbf{1}_i)\,||\,\mathbf{c}] = [(\mathbf{a}+\mathbf{1}_i)\,\mathbf{b}\,||\,\mathbf{c}] + (A_i - B_i)\,[\mathbf{ab}\,||\,\mathbf{c}]\,. \tag{2.19}$$

Además de la simplificar el cálculo de las ERI, se reducen las sumatorias para evaluar el potencial de Coulomb sobre tres índices, $\tilde{V}^C_{ab}$, por la naturaleza no-contraída de la densidad auxiliar. Derivando el segundo término de la ecuación (1.44) con respecto a un elemento de la matriz de densidad P_{ab}, se tiene que:

$$\tilde{V}^C_{ab} = \sum_{k=1}^{K}\sum_{l=1}^{L}\sum_{m=1}^{M_{aux}} x_m D^a_k D^b_l \left[\mathbf{a}_k\mathbf{b}_l\,||\,\bar{k}_m\right]. \tag{2.20}$$

2.2 Mallados moleculares de Becke

El mallado de Becke [40] consiste en utilizar un conjunto de pesos para descomponer una función molecular, que depende de todos los átomos, en componentes independientes para cada núcleo. Esto se lleva a cabo partiendo el espacio molecular en celdas centradas en cada átomo parecidas a los poliedros de Voronoi, pero con fronteras difuminadas. La derivación del mallado de Becke está descrita en el Apéndice B.

La integración numérica de una función tridimensional, $F(r, \theta, \phi)$, se puede llevar a cabo de varias maneras. La función se puede integrar por regla de los trapecios, regla de Simpson o cuadratura gaussiana cada dimensión de manera independiente. Otra forma es mediante una separación de la función F en una parte radial, que se integra con una cuadratura gaussiana, y una parte angular, que se integra mediante el mallado de Lebedev. El mallado de Becke se basa en esta segunda opción. El peso de un punto del mallado de Becke, además de las contribuciones radial y angular, incluye un factor que indica qué tan cerca está el punto del átomo. Dicho factor depende de la función de corte que define las fronteras difuminadas del un átomo de la molécula.

Este mallado es utilizado para la evaluación del potencial de intercambio-correlación, cuya expresión es difícil de integrar analíticamente. Para esto, primero se genera el mallado, después se obtiene el valor de la densidad y sus gradientes en cada punto, posteriormente, se evalúa el funcional y sus derivadas, y finalmente, se obtiene la matriz del potencial de intercambio correlación. Los cuatro pasos son lentos debido a la cantidad de puntos que evalúan, en especial, el segundo y cuarto paso. En este trabajo se paraleliza la obtención de los pesos de Becke, ya que es sencillo y el algoritmo está descrito por Luehr *et al.* [5].

Lo costoso de la generación del mallado es el paso en el que se obtienen los pesos de Becke para cada átomo [5]. El Algoritmo 1 describe el cálculo de dichos pesos. Cada átomo crea un arreglo lineal que contiene las coordenadas de cada punto de la malla y las contribuciones a los pesos de Becke por la cuadratura de Gauss-Chebyshev y Lebedev. Posteriormente, realiza un llamado a la GPU para que genere los pesos en todos los puntos que le corresponden. Los puntos de la malla son distribuidos entre los hilos de trabajo.

La evaluación del peso en cada punto está descrito por el Algoritmo 2. Primero, se obtiene el arreglo con las distancias de los núcleos al punto, luego, se genera la celda con límites difuminados a partir de la multiplicación de las funciones de corte, y finalmente se escala la celda para obtener el peso de Becke.

para *cada átomo* A *de la molécula* **hacer**
    crear vectores de datos;
    obtener pesos por cuadratura;
    escalar pesos por Lebedev;
    llamar a rutina bweights en GPU;
    evaluar densidad y gradientes;
    evaluar intercambio y correlación;
fin

Algoritmo 1: Llamado a la GPU para obtener los pesos de Becke

para *cada punto* i *de la malla* **hacer**
    para *cada átomo* A **hacer**
        calcular r_{iA};
        $p_{atom} = 1$;
    fin
    para *cada átomo* A **hacer**
        para *cada átomo* $B \neq A$ **hacer**
            obtener r_{AB};
            obtener $\mu_{AB} = (r_{iA} - r_{iB})/R_{AB}$;
            obtener función de corte de Becke $s(\mu_{AB})$;
            obtener celda $P_A* = s(\mu_{AB})$;
        fin
    fin
    para *cada átomo* A **hacer**
        $P_{suma} += P_A$
    fin
    escalar pesos $g_i* = P_{atom}/P_{suma}$;
fin

Algoritmo 2: Generación de los pesos de Becke (rutina bweights)

If you were plowing a field, which would you rather choose? Two strong oxen or 1024 chickens?

Seymour Cray

3

Unidades de procesamiento gráfico

La ley de Moore dice que el número de transistores que caben dentro de un chip se duplica cada dos años [41]. Esta ley se ha cumplido por más de 50 años, ya que el diseño de transistores los ha vuelto más pequeños, rápidos y han disminuido su comsumo energético. Hoy en día, los transistores tienen tamaños de $45 - 22$ nm. Esta ley está por llegar a su límite porque tener billones de transistores trabajando al mismo tiempo genera mucho calor y se requiere enfriamiento. Por esta misma razón, en la última década, se ha dejado de aumentar la frecuencia de reloj (velocidad para realizar una operación) ya que implica un mayor consumo de energía.

Para reducir el consumo energético, se han construido procesadores con transistores más pequeños, más lentos, pero con mayor aprovechamiento energético. Esta idea dio lugar a las GPU. Dichas unidades se enfocan más en procesamiento de datos y generan menos calor, pero la programación está restringida por un *hardware* de control más sencillo que una CPU [42]. A diferencia de una CPU que busca realizar un trabajo lo más rápido posible (minimizar latencia), la GPU optimiza rendimiento (maximiza el trabajo realizado en un tiempo definido). Una CPU puede tener decenas de procesadores, mientras que una GPU puede llegar a tener

miles de núcleos de trabajo. Se denomina un sistema de cómputo de alto rendimiento (HPC) heterogéneo o híbrido cuando está constituido tanto por CPU como GPU. Generalmente, las GPU se emplean como coprocesadores de las CPU ya que la mayor parte del hardware está apartado para procesamiento de datos y son muy eficientes para realizar operaciones sencillas. En dichos sistemas, la CPU, la cual posee un *hardware* de control más complejo, realiza las operaciones que requieren más instrucciones. La distribución de trabajo entre la CPU y la GPU es muy importante, ya que cada unidad está optimizada para tareas con ciertas características.

3.1 Ley de Amdahl

Un factor importante para determinar la eficiencia de un programa, además de que funcione correctamente, es que realice el trabajo rápidamente. Para medir la eficiencia de un código, se emplea la ley de Amdahl [43]. La ley de Amdahl es una serie de observaciones que se reducen a la siguiente expresión:

$$S(n) = \frac{T(1)}{T(n)}. \tag{3.1}$$

En esta ecuación, $S(n)$ es la aceleración que el programa puede presentar con n núcleos de trabajo, $T(1)$ y $T(n)$, son los tiempos de ejecución con 1 núcleo y n de ellos, respectivamente. Si la paralelización es óptima, $T(n)$ se puede expresar como:

$$T(B, n) = T(1)\left(B + \frac{1}{n}(1 - B)\right), \tag{3.2}$$

donde B es la fracción del programa que es serial, y, por lo tanto $1 - B$ es la fracción paralelizada. Simplificando y considerando que $P = 1 - B$, $S(n)$ es:

$$S(P, n) = \frac{1}{(1 - P) + \frac{P}{n}}. \tag{3.3}$$

El límite de esta función cuando $n \to \infty$ es:

$$\lim_{n\to\infty} S(P, n) = \frac{1}{(1 - P)}. \tag{3.4}$$

Esto implica que $S(n)$ presenta un límite asintótico, el cual cambia en función del porcentaje paralelizado P. En la figura 3.1 se muestra el comportamiento de $S(n)$ a diferentes valores de

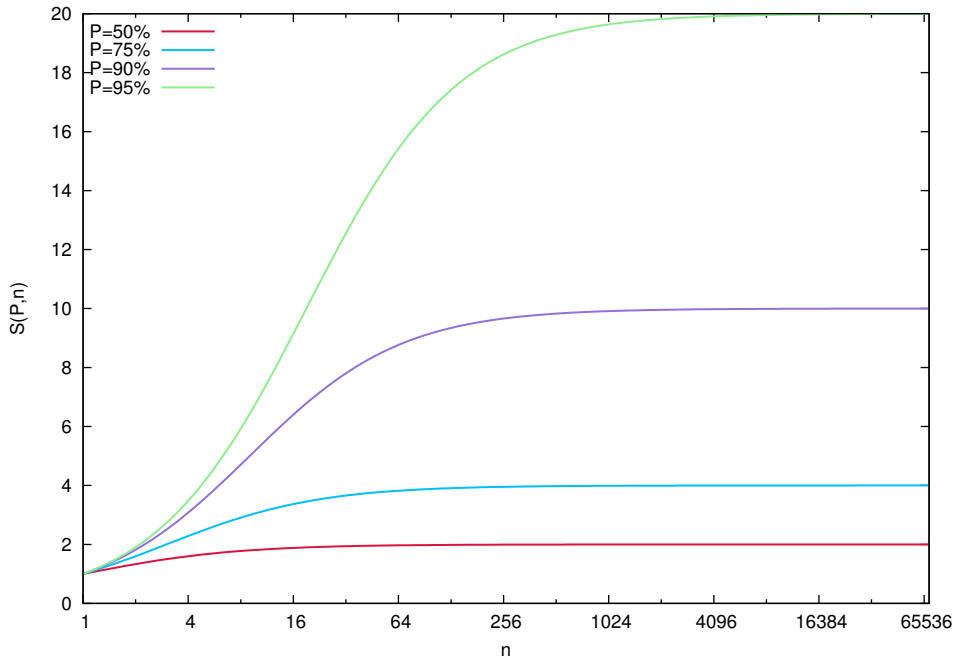


Figura 3.1: Ley de Amdahl para diferentes porcentajes de código paralelizado.

P y se observa que este límite aumenta si más parte del código se paraleliza. Es importante notar que S máxima, cuando $n \to \infty$, con una $P = 0.95$ es sólo de 20 veces y mucho menor si P disminuye. Esto significa que es necesario incrementar el porcentaje paralelizado lo máximo posible.

Ahora, el límite de $S(n)$ cuando $P \to 1$ es:

$$\lim_{P \to 1} S(P, n) = n. \tag{3.5}$$

Esto demuestra que también es importante distribuir el trabajo en más núcleos. En sistemas heterogéneos, que consisten de unidades de procesamiento central (CPU) y unidades de procesamiento gráfico (GPU), la ecuación (3.3) no se puede aplicar de manera directa. Esto se debe a que no tienen las mismas velocidades de reloj (la CPU es más rápida que la GPU), pero si consideran que son iguales, aumentar P y n sería el mismo objetivo. En programas que aprovechan arquitecturas de este tipo, se busca que la parte serial se ejecute en la CPU, por su velocidad de reloj, y la parte paralelizada en la GPU, por su cantidad de núcleos de trabajo.

3.2 Arquitectura de una GPU

Una GPU tiene más transistores dedicados a ser unidades aritméticas lógicas (ALU por sus siglas en inglés) que para control y caché. En contraste, una CPU tiene menos ALU y más transistores para el control y caché de datos.

Para poder aprovechar al máximo una tarjeta gráfica, es necesario considerar su arquitectura para conocer sus limitaciones [42]. Los datos deben estar en bloques denominados *streams* y su transformación se lleva a cabo mediante un *kernel*. Los *streams* se forman y son modificados en paralelo por procesadores que ejecutan el *kernel*. En esta sección se describirán las especificaciones de una GPU Tesla K40, que es la tarjeta empleada en este estudio.

La Tesla K40 tiene 15 multiprocesadores de *stream* (SM, por sus siglas en inglés) con 192 núcleos de trabajo en cada uno de ellos y trabajan a una velocidad de reloj de 0.75 GHz. Estos núcleos son más lentos que los procesadores de una CPU convencional, pero son 2880 unidades de procesamiento totales. Si se considera una CPU con 16 procesadores, la Tesla K40 tiene 180 veces más núcleos que la CPU.

Cada SM tiene 192 núcleos que reciben una misma instrucción. Esta implementación se denomina *Single Instruction Multiple Thread* (SIMT). La unidad de instrucción propaga la misma indicación a cada procesador de la SM, tal que cada uno de ellos modifique los datos que le corresponden en cada ciclo de reloj.

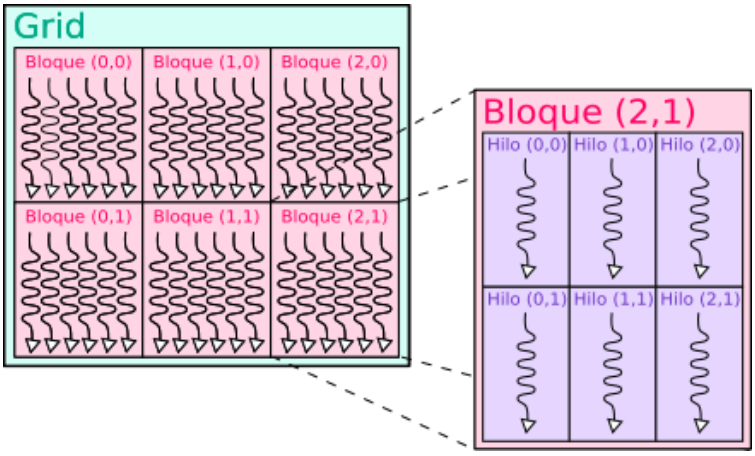


Figura 3.2: Jerarquía de los hilos en CUDA. El *grid* está constituido por bloques y éstos a su vez son un conjunto de hilos (flechas).

En la Figura 3.2 se muestra la jerarquía de los hilos en CUDA. Los hilos se agrupan en bloques y éstos a su vez forman un *grid*. Los bloques y *grids* pueden ser de una, dos o hasta tres dimensiones, dependiendo del problema de interés. El índice *threadIdx* y *blockIdx*, junto con las dimensiones *blockDim* y *gridDim*, permiten recuperar el índice global de un hilo específico, es decir, la posición del elemento de vector, matriz o volumen a procesar [43].

Otra característica importante de las GPU es la jerarquía de memoria que tienen. Existe una memoria local (hilo), memoria compartida (bloque) y memoria global (GPU). El acceso a memoria local es más rápido que a la compartida, y esta a su vez es mucho más rápido que a la global. Por lo tanto, es muy importante que cada *stream* sea independiente de los demás, ya que así se evita el uso de memoria global y compartida [43]. Si es necesario que se transfiera información de un hilo a otro hilo del bloque, se recomienda emplear la memoria compartida y buscar la mejor manera para regular el acceso a ella. La memoria de la CPU y la GPU son independientes. La GPU no puede accesar a la memoria de la CPU y viceversa, por lo tanto, es muy importante realizar un copiado de datos antes y después de emplear la GPU.

3.3 CUDA

El uso de las tarjetas gráficas se enfocaba principalmente al procesamiento de imágenes en videojuegos debido a la complejidad de la programación usando DirectX. El desarrollo de interfaces para programación de GPU como lenguaje de cómputo abierto (OpenCL: Open Computing Language) y CUDA facilitó el control de las GPU, las cuales ganaron potencial como alternativas para la implementación de códigos científicos. Este tipo de programación es de propósito general en GPU, es decir, se puede utilizar para programas no relacionadas a gráficos. CUDA funciona para diferentes lenguajes: Python, Fortran, C/C++. También está la posibilidad de utilizar OpenACC (equivalente a OpenMP) para paralelizar con una granularidad menos fina. En este trabajo se utilizó CUDA C, ya que esta distribución es gratuita y hay mayor flexibilidad para la paralelización. Bajo el modelo de maestro-esclavo, la CPU es el nodo maestro (anfitrión ó *host* en inglés) y las GPU son los nodos esclavos (dispositivo ó *devices* en inglés).

El esquema general de un programa de CUDA [42] consiste de lo siguiente (en itálicas está escrita la directiva generalizada para realizar la instrucción correspondiente):

1. La CPU asigna espacio de memoria en la GPU
2. La CPU copia los datos desde CPU a GPU
3. La CPU ejecuta el kernel en la GPU
4. La CPU copia los resultados desde GPU a CPU

La GPU no puede accesar la memoria de la CPU. Primero, se asigna el espacio necesario para que posteriormente se copien los datos desde el *host* hasta el *device*. Es importante notar que todas las instrucciones son realizadas por el *host*, ya que la GPU no puede dar instrucciones propias. Se realiza la transformación de los datos mediante la ejecución del *kernel* por cada hilo de trabajo. Finalmente, debido a que la CPU tampoco puede leer información de la memoria de la GPU, se copian los resultados del *kernel* a la CPU. Una herramienta importante para códigos que ejecutan varios *kernels* es la sincronización de hilos para evitar lectura y escritura de datos incorrectos y tener una cooperación eficiente.

4

Validación del programa

La implementación se llevo a cabo en el código para cálculos de estructura electrónica de distribución libre, *Parakata*. Los resultados de la evaluación de las ERI y los pesos de Becke empleando GPU se muestran en esta sección. La implementación actualmente abarca hasta funciones base de momento angular p, por lo que los sistemas de estudio y los conjuntos de funciones base están limitados. La validación se realizó con cadenas de átomos de hidrógeno de longitudes distintas y con alcanos lineales.

Para las cadenas de átomos de hidrógeno se utilizó la base 6-311G(d,p). Debido a que únicamente se tiene implementada la evaluación de ERI hasta momento angular p, se utiliza un conjunto base auxiliar práctico para la valoración del código, pero no tiene una derivación rigurosa. El conjunto base auxiliar está detallado en el Apéndice C. Se observa que las funciones auxiliares tienen coeficientes unitarios porque no tienen contracción. En los alcanos lineales, se emplea la base 6-311G y la base auxiliar que está en el Apéndice C. La base auxiliar no es lo suficientemente grande para obtener la convergencia del procedimiento SCF por lo que se presentarán los resultados para un ciclo del SCF únicamente.

La CPU empleada es una Intel(R) Xeon(R) CPU $E5-2630$ v3 @ 2.40 GHz y la GPU es una NVIDIA Tesla K40 con 2880 núcleos @ 0.75 GHz. El código original es serial y el nuevo programa utiliza un sistema heterogéneo de 1CPU-1GPU.

4.1 Esquema de programación de las integrales de tres centros

En el código original, se evalúan únicamente $(N_{bas}[N_{bas}+1]/2)N_{aux}$ integrales y descarta aquellas que son muy pequeñas. En el programa modificado se evalúan $N_{bas}^2 N_{aux}$ integrales, aunque se repitan las integrales por la simetría de las ERI. Para una comparación más justa, también se modificó el código original para que evalúe todas las integrales como el código de GPU.

El código original es serial. Se realiza un barrido sobre cada capa (*shell*) de cada centro y se evalúa el conjunto o *batch* de integrales contraídas, cuyo tamaño depende del momento angular de cada uno de los centros.

Para emplear correctamente la GPU, es necesario cambiar el algoritmo de trabajo utilizado en el código serial. En el código que utiliza una GPU, se crean los vectores de datos cuando se genera la matriz de traslape.

Entonces, se tiene un vector de coordenadas, uno de momento angular, uno de coeficientes, uno de exponentes y uno de constantes de normalización para cada *shell pair* de funciones primitivas, bra, posible. Esto mismo se realiza para la parte del ket (densidad auxiliar, ecuación (1.33)) y se realiza un solo llamado a la rutina *primeris*, que contiene el *kernel* como se describe en el algoritmo 3. Cada hilo identifica el momento angular de las integrales que se le asigna, llama a la subrutina correspondiente al tipo de integral y las evalúa. De acuerdo al algoritmo 4, un hilo realiza el barrido sobre todas las funciones auxiliares para un *shell pair* fijo, es decir, un hilo evalúa una columna de la figura 2.2. El índice globlal idx del algoritmo 4 está dado por el índice del *shell pair* correspondiente. El arreglo de todas las ERI primitivas es copiado de GPU a CPU. En la CPU se contraen las ERI y se guardan en un archivo posteriormente.

crear vectores de datos;
llamar a rutina primeris en GPU;
contracción de ERI;
para *cada función base auxiliar* **hacer**
    recuperar matriz de ERI;
    escribir en archivo;
fin

Algoritmo 3: Evaluación de ERI en GPU

inicialización (cudaMalloc, cudaMemcpy);
para *cada hilo* **hacer**
    $idx = threadIdx.x + blockIdx.x * blockDim.x$;
    obtener datos de *shell pair* asociado a idx;
    para *cada shell base auxiliar* **hacer**
        si *cierto momento angular en cada centro* **entonces**
            llamar a rutina correspondiente;
            normalizar;
        fin
        $idx+ = blockDim.x * gridDim.x$;
    fin
fin

Algoritmo 4: Rutina primeris en GPU

4.2 Resultados de la evaluación de las integrales de tres centros

4.2.1 Cadenas lineales de hidrógeno

En la Tabla 4.1 se presentan los tiempos de evaluación de las ERI para cadenas de átomos de hidrógeno empleando precisión doble. La aceleración de los tiempos es significativo cuando se tienen muchos átomos. Cuando se tienen pocos átomos, se observa que la evaluación de las integrales con el algoritmo propuesto en este trabajo puede ser incluso más lenta, lo cuál también ocurre cuando se utiliza OpenMP (*Open Multi-Processing*) o MPI (*Message Passing Interface*). Si el trabajo es muy sencillo, distribuir el trabajo a cada nodo y luego juntarlo requiere más tiempo que usar un solo nodo de trabajo.

La cadena de 16 átomos tiene 9216 *shell pairs* de funciones primitivas. Idealmente, la razón entre los tiempos de evaluación crecería linealmente conforme aumenta el número de hilos y a partir de los 9216 hilos de trabajo, la aceleración se mantendría constante. En la Figura 4.1, se observa que una tendencia similar, pero la razón entre los tiempos deja de aumentar, incluso disminuye, antes de que se tenga una relación $1:1$ entre el número de *shell pairs* y el número de hilos de trabajo.

De acuerdo a la Tabla 4.1, la aceleración máxima observada es de más de $16\times$ para la cadena de 32 átomos de hidrógeno. Se observa que para las cadenas de hidrógeno de 32 y 64 átomos, el número de *shell pairs*, 65536 y 147456, respectivamente, excede el número de hilos de trabajo totales, 25600, lo cual significa que hay hilos que están evaluando las ERI para más de un *shell pair*. Es decir, la cadena de 32 átomos necesita que cada hilo evalúe 2 o 3 *shell pairs* ($\frac{65536}{25600} \approx 2.6$). De manera similar a la cadena de 16 átomos, el crecimiento de la razón entre los tiempos de evaluación en CPU y en GPU de las cadenas de 32 y 64 átomos aparenta llegar a un límite superior antes de que se utilicen suficientes hilos de trabajo para que cada hilo calcule una sola integral. Otro aspecto interesante de la Figura 4.1 es que la cadena de 64 átomos presenta una aceleración menor a la de la cadena de 32 después de los 2048 hilos de trabajo. Esto se discutirá más adelante. El promedio de errores absolutos (PEA) es despreciable en la evaluación de las ERI.

Tabla 4.1: Tiempos de evaluación de las ERI de cadenas de átomos de hidrógeno de longitudes distintas con CPU y con GPU empleando precisión doble. Se utilizan 200 bloques de 128 hilos en la GPU, es decir, 25600 hilos totales. PEA se refiere al promedio de los errores absolutos con respecto al código original.

# átomos	# *shell pair*	Tiempo CPU (s)	Tiempo GPU (s)	t_{CPU}/t_{GPU}	PEA
2	144	0.001	0.037	0.027	1.52×10^{-17}
4	576	0.012	0.045	0.267	1.69×10^{-17}
8	2304	0.085	0.046	1.848	1.21×10^{-17}
16	9216	0.640	0.084	7.618	6.03×10^{-18}
32	65536	5.447	0.336	16.217	3.29×10^{-18}
64	147456	43.872	3.527	12.441	1.68×10^{-18}

4.2.2 Alcanos lineales

En la Tabla 4.2 se observan los tiempos de evaluación de las ERI para alcanos de diferente longitud. Los errores son similares a los de las cadenas de átomo de hidrógeno y son despreciables. Además, en las Figuras 4.1 y 4.2, la razón entre los tiempos de las cadenas de átomos

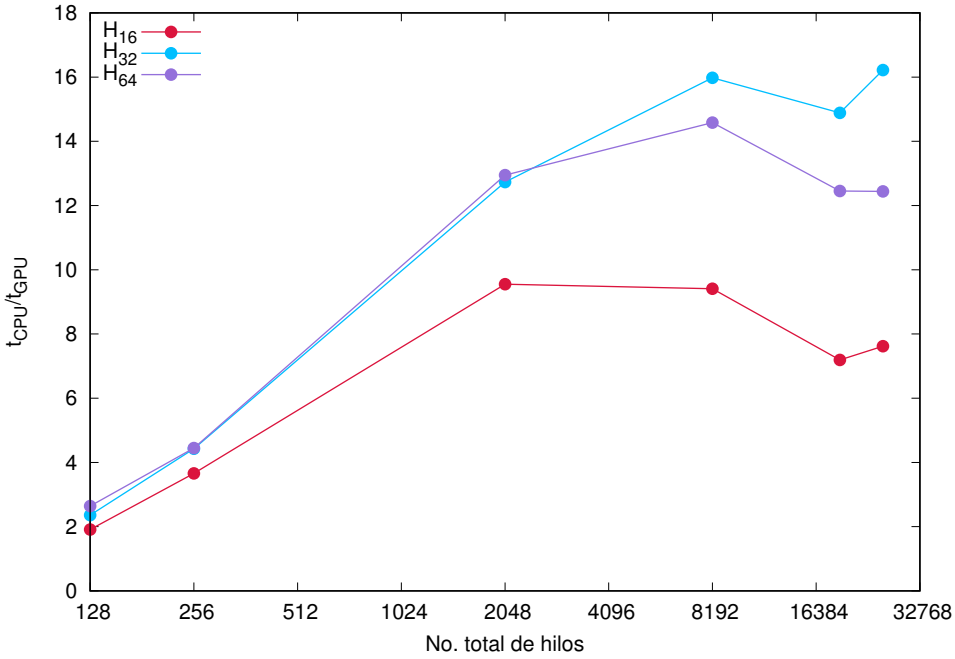


Figura 4.1: Razón entre los tiempos de evaluación de las ERI con CPU y con GPU como función del número de hilos totales empleados. Cada color representa una cadena lineal de átomos de hidrógeno de longitud diferente. El eje horizontal está en escala $\log_2$.

de hidrógeno y de los alcanos crece rápidamente hasta 2048 hilos de trabajo y disminuye su aumento, incluso puede disminuir en ciertos casos. Utilizar 2048 hilos de trabajo da una mejor relación costo-tiempo.

Se tiene que el icosano, CH_3-$(CH_2)_{18}$-CH_3, igual que la cadena de 64 átomos de hidrógeno, no presenta una aceleración tan favorable como las otras moléculas. Se realizó la evaluación de las ERI en dos *batches*, de manera tal que cada *batch* llame a la GPU independientemente, uno después de que haya terminado el otro. La suma de los tiempos de evaluación en la GPU de las ERI es de 8.472 s, menor que el tiempo reportado en la Tabla 4.2 para el icosano. Con tres *batches*, el tiempo disminuye a 7.653 s. Esto implica que la GPU no es tan eficiente si procesa arreglos demasiado grandes de manera concurrente. Utilizar *batches* puede reducir el tiempo de cálculo porque se evita el agotamiento de la RAM.

Tabla 4.2: Tiempos de evaluación de las ERI de alcanos lineales, CH_3-$(CH_2)_n$-CH_3, con CPU y con GPU empleando precisión doble. Se utilizan 200 bloques de 128 hilos en la GPU, es decir, 25600 hilos totales. PEA se refiere al promedio de los errores absolutos con respecto al código original.

n	# *shell pair*	Tiempo CPU (s)	Tiempo GPU (s)	t_{CPU}/t_{GPU}	PEA
3	51076	1.754	0.219	8.008	4.89×10^{-18}
8	190096	13.058	0.9879	13.218	4.88×10^{-18}
13	417316	41.638	3.160	13.179	3.66×10^{-18}
18	732736	96.273	12.143	7.928	2.96×10^{-18}

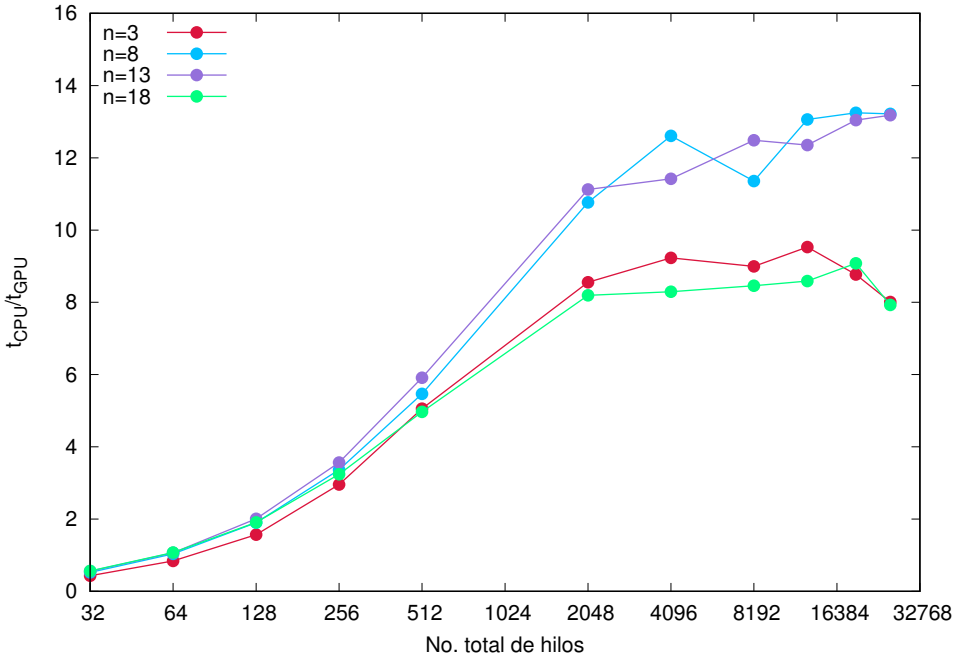


Figura 4.2: Razón entre los tiempos de evaluación de las ERI con CPU y con GPU como función del número de hilos totales empleados de los alcanos lineales CH_3-$(CH_2)_n$-CH_3 con $n = 3$, $n = 8$, $n = 13$ y $n = 18$. El eje horizontal está en escala $\log_2$.

4.3 Esquema de programación de la obtención de los pesos de Becke

El algoritmo utilizado está descrito en la sección de integrales molecularespor el algoritmo 1. El código serial utiliza el mismo esquema. Primero se genera un arreglo de datos con las coordenadas de cada punto del mallado y las contribuciones al peso de la cuadratura de Gauss y el mallado de Lebedev. Después se realiza un llamado a la CPU (si es el código serial) o a la GPU (el código optimizado) por cada átomo de la molécula. La CPU o la GPU se encarga de generar

los pesos de Becke para cada punto del mallado del átomo correspondiente mediante el algoritmo 2. Finalmente, las coordenadas y los pesos de cada punto del mallado son utilizados para evaluar el potencial de intercambio correlación.

4.4 Resultados de la obtención de los pesos de Becke

4.4.1 Cadenas lineales de hidrógeno

En la Tabla 4.3 se muestran las aceleraciones obtenidas para la evaluación de los pesos de Becke para diferentes números de átomos. La malla tiene un número de puntos constante, 22650 por átomo, y sólo depende del número de átomos en la molécula. La generación de los pesos de Becke es costosa incluso para sistemas pequeños. A diferencia de la evaluación de las ERI, que es ineficiente para sistemas pequeños, la Tabla 4.3 muestra que incluso la cadena de dos átomos de hidrógeno tiene una aceleración de $2\times$. Esto es porque el número de puntos de la malla es grande. A excepción de la cadena de cuatro átomos de hidrógeno, se observa que la tendencia es que conforme aumenta el número de átomos en la molécula, mejora la aceleración.

Tabla 4.3: Tiempos de evaluación de los pesos de Becke para cadenas de átomos de hidrógeno de longitudes distintas con CPU y con GPU empleando precisión doble. Se utilizan 200 bloques de 128 hilos en la GPU, es decir, 25600 hilos totales. PEA se refiere al promedio de los errores absolutos con respecto al código original.

# átomos	# puntos	Tiempo CPU (s)	Tiempo GPU (s)	t_{CPU}/t_{GPU}	PEA
2	45300	0.002	0.001	2.000	6.98×10^{-17}
4	181200	0.005	0.004	1.667	6.12×10^{-17}
8	362400	0.062	0.008	7.750	7.26×10^{-17}
16	724800	0.488	0.017	28.706	6.03×10^{-18}
32	1449600	3.973	0.126	31.535	7.99×10^{-17}
64	28992000	32.835	0.849	38.680	9.97×10^{-17}

La aceleración de los pesos de Becke es más apreciable que la correspondiente a la evaluación de las ERI. La máxima aceleración obtenida para la cadena de H_{64} mostrada en la Tabla 4.3 es de $40\times$ aproximadamente. Una posible explicación es que, en la evaluación de las ERI, cada hilo necesita identificar el tipo de integral, es decir el momento angular de cada centro, por lo tanto, se requieren varias operaciones de condicionales. El resultado es un *kernel* con

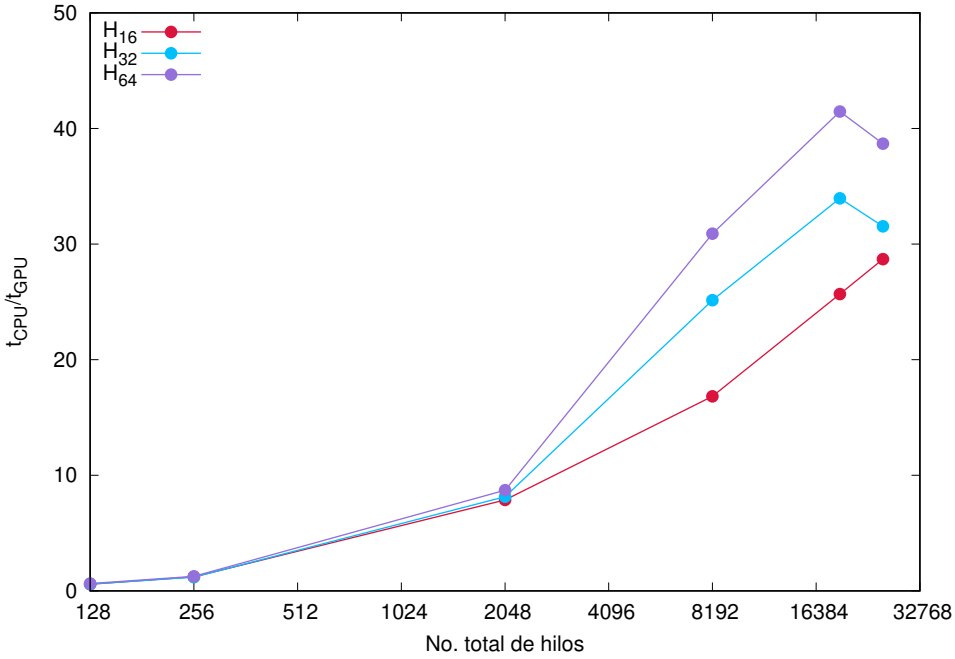


Figura 4.3: Razón entre los tiempos de evaluación de los pesos de Becke con CPU y con GPU como función del número de hilos totales empleados. Cada color representa una cadena lineal de átomos de hidrógeno de longitud diferente.

muchas instrucciones, que es ineficiente en una GPU. Una posible solución para esto es agrupar las integrales en bloques por el momento angular de cada uno de los centros y llamar al *kernel* correspondiente al tipo de integral. Así se evita la necesidad de identificar la integral con múltiples condicionales. Además, la separación de los datos en bloques independientes puede acelerarse mediante llamados asíncronos a la GPU, o utilizando OpenMP acoplado a varias GPU, de manera que cada una de ellas procesa un bloque de ERI distinto.

El algoritmo para la evaluación de los pesos de Becke consiste en realizar un llamado a la GPU por cada átomo, uno por uno. Por lo tanto, se esperaría que el aumento en la razón entre los tiempos de evaluación en la CPU y la GPU sea constante hasta que el número de hilos sea de 22650. En la Figura 4.3 se observa que casi siempre hay un aumento en la razón. Esto reafirma que la programación de la evaluación de las ERI no es tan eficiente.

4.4.2 Cadenas de alcanos

Tanto los tiempos de ejecución como los PEA mostrados en la Tabla 4.4 son consistentes con la validación de las cadenas de átomos de hidrógeno de longitudes distintas. La aceleración es más significativa cuando se tienen sistemas de muchos átomos.

Tabla 4.4: Tiempos de evaluación de los pesos de Becke para alcanos lineales, CH_3-$(CH_2)_n$-CH_3, con CPU y con GPU empleando precisión doble. Se utilizan 200 bloques de 128 hilos en la GPU, es decir, 25600 hilos totales. PEA se refiere al promedio de los errores absolutos con respecto al código original.

n	# puntos	Tiempo CPU (s)	Tiempo GPU (s)	t_{CPU}/t_{GPU}	PEA
3	770100	0.602	0.025	24.076	4.85×10^{-17}
8	1449600	3.976	0.116	34.279	3.93×10^{-17}
13	2129100	12.873	0.362	35.571	4.08×10^{-17}
18	2808600	29.821	0.775	38.483	3.37×10^{-17}

En la Figura 4.4 se observa que la misma tendencia que la Tabla 4.4. La razón entre los tiempos de evaluación de los pesos de Becke para los alcanos lineales crece hasta casi llegar a 25600 hilos de trabajo.

4.5 Resultados del ciclo de campo autoconsistente

El esquema, que está mostrado en la Figura 4.5, describe la distribución de trabajo en el sistema heterogéneo que se empleó. La CPU se encarga de un mayor número de pasos del SCF, pero los procesos más costosos se realizan en la GPU. Únicamente se utiliza la GPU para evaluar las ERI al inicio del ciclo iterativo y la obtención de los pesos de Becke en cada iteración.

Tabla 4.5: Perfil de tiempos de ejecución para la evaluación de las ERI, los pesos de Becke y la evaluación del potencial de intercambio-correlación en una iteración de un ciclo de campo autoconsistente de los alcanos lineales CH_3-$(CH_2)_n$-CH_3 con $n = 3$, $n = 8$, $n = 13$ y $n = 18$. La evaluación del potencial de intercambio-correlación incluye la obtención de los pesos de Becke.

n	ERI	Becke	V_{xc}	SCF (s)
3	25.56%	8.77%	69.62%	6.862
8	36.41%	11.09%	58.02%	35.863
13	43.04%	13.31%	50.86%	96.739
18	47.40%	14.68%	45.82%	203.095

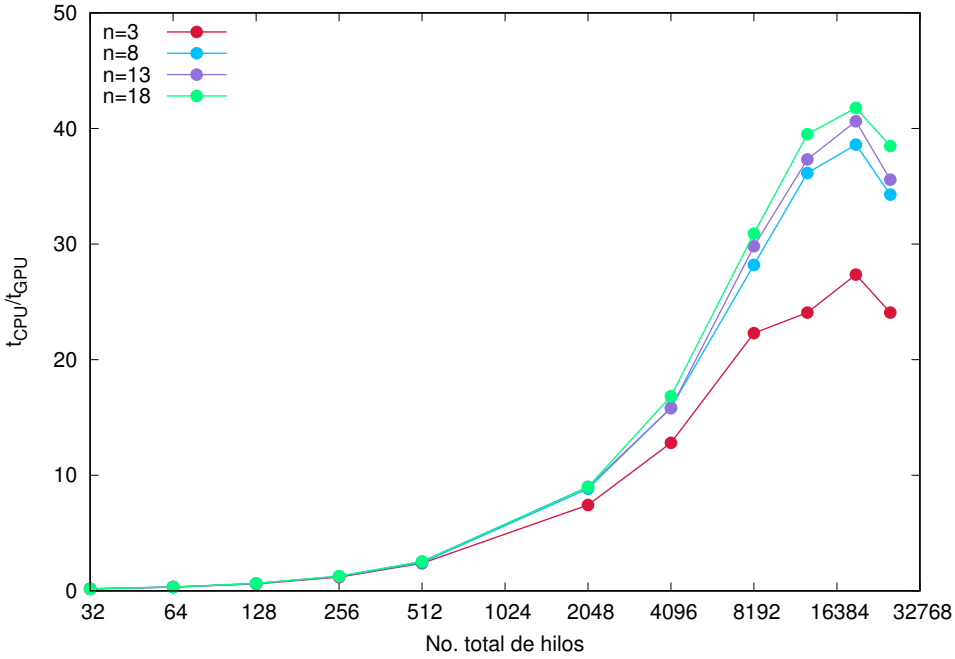


Figura 4.4: Razón entre los tiempos de evaluación de los pesos de Becke con CPU y con GPU como función del número de hilos totales empleados de los alcanos lineales CH_3-$(CH_2)_n$-CH_3 con $n = 3$, $n = 8$, $n = 13$ y $n = 18$. El eje horizontal está en escala $\log_2$.

En la Tabla 4.5 se muestra el porcentaje del tiempo de ejecución de una iteración del ciclo de campo autoconsistente que se atribuye a la evaluación de las ERI, los pesos de Becke y el potencial de intercambio-correlación. La evaluación del potencial incluye la obtención de los pesos de Becke, por lo que la resta de estos porcentaje corresponde al cálculo de la densidad, sus derivadas y la evaluación del funcional de intercambio-correlación (Dirac y VWN). Conforme crece el sistema, la evaluación de las ERI aumenta su contribución al tiempo total de ejecución. Aunque en un ciclo de campo autoconsistente completo, podría no contribuir mucho porque las ERI se calculan una vez y se guardan. El cálculo de los pesos de Becke también aumenta su contribución conforme crece el sistema, pero una parte importante del tiempo de ejecución se atribuye a los otros pasos que implican la evaluación del potencial de intercambio-correlación.

Aunque cada paso individual haya sido acelerado apreciablemente, esto no necesariamente se reflejará en el código completo porque, de acuerdo a la ecuación (3.3), es necesario paralelizar al menos el 50% para obtener una aceleración de más de 2 para la ejecución del programa completo.

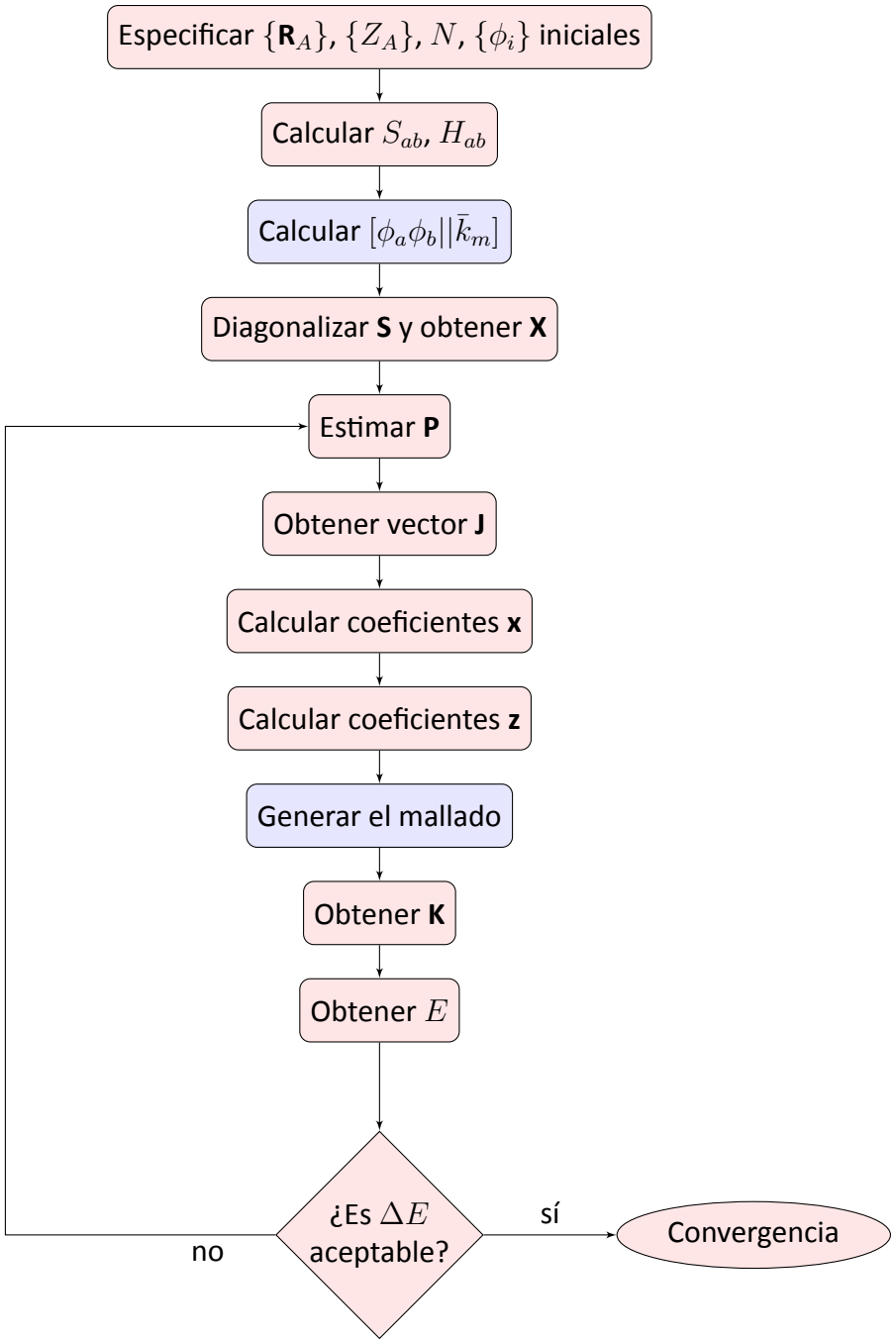
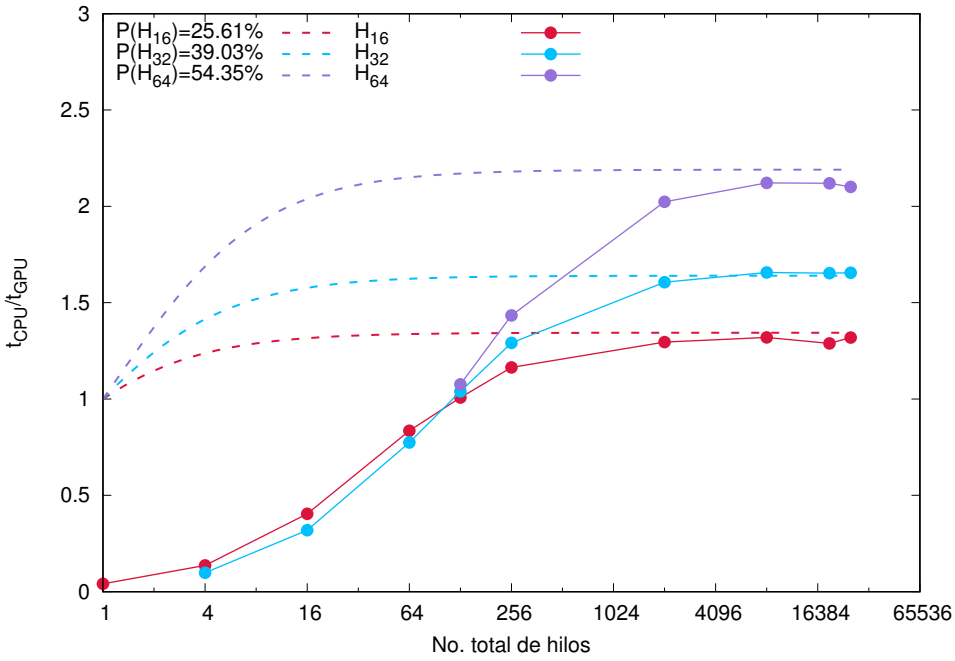
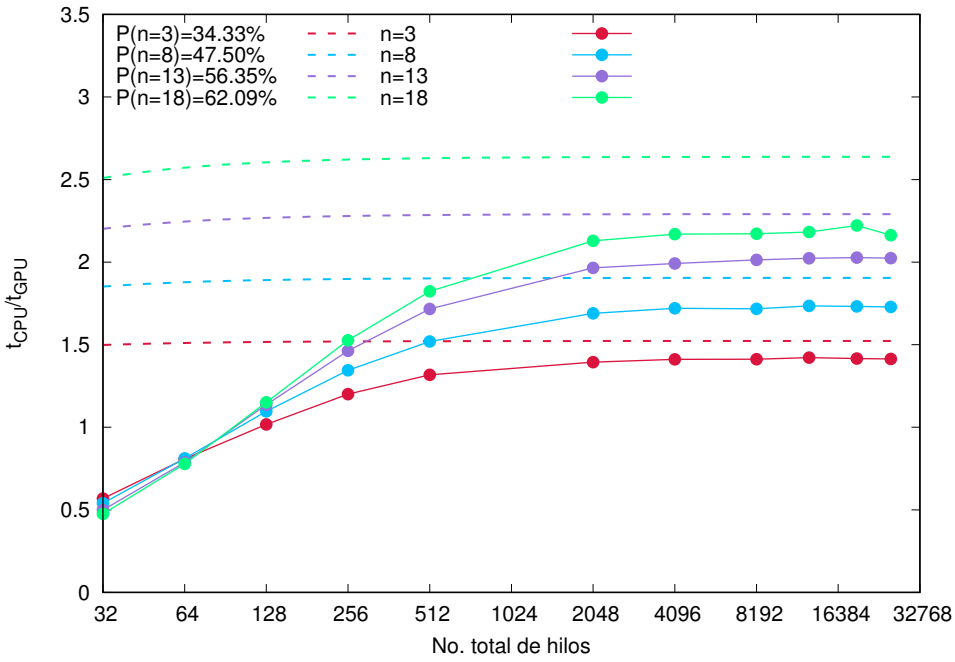


Figura 4.5: Distribución del trabajo en un SCF basado en la ADFT en un sistema heterogéneo. El color rojo indica que el proceso se realiza en la CPU y el color azul que se procesa en GPU.



(a) Cadenas de átomos de hidrógeno



(b) Alcanos lineales.

Figura 4.6: Razón entre los tiempos de ejecución de una iteración del ciclo de campo autoconsistente con CPU y con GPU como función del número de hilos totales empleados para H_{16}, H_{32}, H_{64} y los alcanos lineales CH_3-$(CH_2)_n$-CH_3 con $n = 3$, $n = 8$, $n = 13$ y $n = 18$. El eje horizontal está en escala $\log_2$. Las líneas punteadas representan la ley de Amdahl con porcentajes de paralelización diferentes. La P está definida por el porcentaje del tiempo total de ejecución del código serial modificado que fue empleado para la evaluación de las ERI y los pesos de Becke.

A pesar de que la ley de Amdahl no es aplicable para sistemas de cómputo heterogéneos, se empleará como referencia para observar tendencias. En este caso, la velocidad de reloj de la GPU es menor que la de CPU, por lo tanto, se necesitarán más núcleos de trabajo en una GPU que en una CPU para alcanzar el límite definido por ley de Amdahl.

En las Figuras 4.6a y 4.6b se observa que se necesitan más núcleos de trabajo en una GPU para alcanzar la asíntota de aceleración de una CPU. El icosano alcanza una aceleración de $\approx 2.1\times$, que es significativamente menor que la calculada por ley de Amdahl, de $\approx 2.6\times$. Esto sucede porque la evaluación de las ERI no es tan eficiente. En general, las otras moléculas están cerca de la asíntota predicha por la ley de Amdahl.

Para mejorar la aceleración de una iteración del ciclo de campo autoconsistente, es necesario paralelizar un porcentaje mayor del código completo, como el resto de los cálculos necesarios para la obtención del potencial de intercambio-correlación.

5

Conclusiones y perspectivas

En este trabajo se logró la paralelización parcial del programa *Parakata*, el cual se encarga de resolver las ecuaciones de KS, que resultan de la combinación lineal de orbitales de tipo gaussiano empleando densidades auxiliares, mediante comprocesamiento en tarjetas gráficas. En particular, se implementó la evaluación de los pesos de Becke, que se utilizan para la integración numérica del potencial de intercambio-correlación, y el cálculo de las ERI de tres centros, para obtener el potencial de Coulomb. En un SCF basado en ADFT, dichos procesos son costosos computacionalmente y su implementación en GPU mostró una aceleración significativa. La obtención de los pesos de Becke alcanzó una aceleración de más de $40\times$. Esto demuestra que es una paralelización muy eficiente. La paralelización de la evaluación de las integrales de tres centros utilizando relaciones de recurrencia de Obara y Saika, se acelera hasta $\approx 16\times$. La eficiencia de la implementación de los pesos de Becke es mejor que la de la evaluación de las ERI porque requieren menos operaciones de condicionales. Se han reordenado, por momento angular, las funciones del conjunto base para separar en bloques de integrales de diferente tipo. Si se emplean bloques de integrales clasificados por el momento angular en cada uno de sus centros, se evitan los condicionales para la identificación del momento

angular de los centros, simplificando de esta manera el *kernel* y mejorando la eficiencia. Otra ventaja de separar las ERI en bloques es que se forman arreglos más pequeños que pueden llamar de manera asíncrona a la GPU, que en caso de tener varias tarjetas gráficas, permitiría asignar un bloque de ERI diferente a cada GPU. Utilizar arreglos de menor tamaño evita saturar la RAM y mejora la aceleración, como se mostró en los resultados de la evaluación de las ERI del icosano. La evaluación de los pesos de Becke no implica saturación de la RAM porque se procesan los pesos por átomo. La implementación del cálculo de los pesos de Becke y las ERI mostró una aceleración máxima de $\approx 2\times$ en una iteración del ciclo autoconsistente para un sistema relativamente grande. Esto se debe a que sólo se realizó una paralelización parcial del programa completo. De acuerdo a la ley de Amdahl, paralelizar un porcentaje mayor del código, aumenta el límite asintótico de aceleración. Esta aceleración será aún mayor si se implementan los otros modulos del programa en GPU.

Este trabajo se puede extender al empleo de orbitales de tipo gaussiano hermitiano, ya que es posible aprovechar las relaciones de recurrencia de los polinomios de Hermite. Actualmente, la evaluación de las ERI de cuatro centros con momento angular mayor a d representa un problema de memoria RAM para implementaciones que requieran el cálculo de gradientes debido al incremento en el momento angular hasta funciones f. La implementación hasta momento angular d empleando ADFT requiere menos RAM, aún para evaluar gradientes nucleares. Es por esto que una implementacón completamente paralelizada de ADFT en GPU hasta momento angular alto podría ser acoplada en códigos de simulaciones de tipo Born-Oppenheimer, y con ello, tener simulaciones a primeros principios de tiempos relativamente largos.

Queda entonces por identificar si ADFT con momento angular alto requiere de trabajar en *kernels* analíticos separados por momento angular o si la mezcla de *kernels* analíticos, para momento angular bajo, y el empleo de la cuadratura de Rys [44], para momento angular alto [45], es más recomendable.

A

El teorema del producto de gaussianas

El producto de dos funciones gaussianas con momento angular arbitrario y centradas en centros **A** y **B** es una función gaussiana centrada en un nuevo centro **P**. Para demostrar esto, primero se consideran dos funciones gaussianas,

$$\begin{aligned}\varphi_k(\mathbf{r}) &= (x-A_x)^{a_x}(y-A_y)^{a_y}(z-A_z)^{a_z}\, e^{-\alpha_k(\mathbf{r}-\mathbf{A})^2},\\ \varphi_l(\mathbf{r}) &= (x-B_x)^{b_x}(y-B_y)^{b_y}(z-B_z)^{b_z}\, e^{-\alpha_l(\mathbf{r}-\mathbf{B})^2}.\end{aligned} \tag{A.1}$$

Si se multiplican las exponenciales de φ_k y φ_l, se tiene que,

$$\begin{aligned}e^{-\alpha_k(\mathbf{r}-\mathbf{A})^2}e^{-\alpha_l(\mathbf{r}-\mathbf{B})^2} &= e^{-\alpha_k r_A^2-\alpha_l r_B^2}\\ &= \exp[-(\alpha_k+\alpha_l)\mathbf{r}\cdot\mathbf{r}+2(\alpha_k\mathbf{A}+\alpha_l\mathbf{B})\cdot\mathbf{r}-\alpha_k\mathbf{A}\cdot\mathbf{A}-\alpha_l\mathbf{B}\cdot\mathbf{B}].\end{aligned} \tag{A.2}$$

El exponente de la nueva gaussiana es

$$\gamma = \alpha_k + \alpha_l, \tag{A.3}$$

y se encuentra centrada en el punto **P**

$$\mathbf{P} = \frac{\alpha_k \mathbf{A} + \alpha_l \mathbf{B}}{\gamma}, \tag{A.4}$$

de manera tal que se cumpla la siguiente igualdad:

$$e^{-\alpha_k r_A^2 - \alpha_l r_B^2} = K \exp[-\gamma(\mathbf{r}\cdot\mathbf{r} - \mathbf{r}\cdot\mathbf{P} + \mathbf{P}\cdot\mathbf{P})]. \tag{A.5}$$

De la ecuación A.2 se observa que los dos últimos términos del argumento de la exponencial son constantes, al igual que el último término del argumento de la exponencial de la ecuación A.5, por lo que

$$K e^{-\gamma \mathbf{P}\cdot\mathbf{P}} = e^{-\alpha_k \mathbf{A}\cdot\mathbf{A} - \alpha_l \mathbf{B}\cdot\mathbf{B}}. \tag{A.6}$$

Si se despeja K,

$$K = e^{-\alpha_k \mathbf{A}\cdot\mathbf{A} - \alpha_l \mathbf{B}\cdot\mathbf{B} + \gamma \mathbf{P}\cdot\mathbf{P}}. \tag{A.7}$$

Es posible expandir **P** · **P**,

$$\begin{aligned} K &= \exp[-\alpha_k \mathbf{A}\cdot\mathbf{A} - \alpha_l \mathbf{B}\cdot\mathbf{B} + \gamma^{-1}(\alpha_k^2 \mathbf{A}\cdot\mathbf{A} + 2\alpha_k\alpha_l \mathbf{A}\cdot\mathbf{B} + \alpha_l^2 \mathbf{B}\cdot\mathbf{B})] \\ &= \exp[\gamma^{-1}(-\alpha_k^2 \mathbf{A}\cdot\mathbf{A} - \alpha_k\alpha_l \mathbf{A}\cdot\mathbf{A} - \alpha_k\alpha_l \mathbf{B}\cdot\mathbf{B} + \alpha_k^2 \mathbf{A}\cdot\mathbf{A} + 2\alpha_k\alpha_l \mathbf{A}\cdot\mathbf{B} + \alpha_l^2 \mathbf{B}\cdot\mathbf{B})] \\ &= \exp[-\gamma^{-1}\alpha_k\alpha_l(\bar{\mathbf{AB}}^2)], \end{aligned} \tag{A.8}$$

donde

$$\bar{\mathbf{AB}}^2 = \mathbf{A} - \mathbf{B}. \tag{A.9}$$

Entonces, cualquier multiplicación de funciones gaussianas resulta en,

$$\begin{aligned} \varphi_k(\mathbf{r})\,\varphi_l(\mathbf{r}) = &(x - A_x)^{a_x}(y - A_y)^{a_y}(z - A_z)^{a_z} \\ &\times (x - B_x)^{b_x}(y - B_y)^{b_y}(z - B_z)^{b_z} \\ &\times e^{-\gamma^{-1}\alpha_k\alpha_l(\bar{\mathbf{AB}}^2)} e^{\gamma(\mathbf{r}-\mathbf{P})^2}. \end{aligned} \tag{A.10}$$

B

Integración numérica

LA INTEGRACIÓN NUMÉRICA es una herramienta matemática muy útil cuando se desea integrar funciones muy complicadas [46]. Uno de los métodos sencillos para integración numérica es por regla de trapecios. Este consiste en dividir el área debajo de la función $f(x)$ en N trapecios con un ancho $\Delta x = (b-a)/N$. La suma de las áreas de los trapecios es la integración de la función

$$A = \int_a^b f(x)\mathrm{d}x \approx \frac{\Delta x}{2}\sum_{i=1}^{N} f(x_i) + f(x_{i+1}). \tag{B.1}$$

En la regla de trapecios, la interpolación es lineal. A diferencia de esta, la regla de Simpson utiliza N parabolas para aproximar $2N$ partes de la función (cada parabola representa dos partes), tal que A es

$$A \approx \frac{\Delta x}{3}\left(f(x_0) + f(x_{2N}) + 4\sum_{i\ non}^{2N-1} f(x_i) + 2\sum_{i\ par}^{2N-2} f(x_i)\right). \tag{B.2}$$

Las reglas de los trapecios y de Simpson son exactas para una función lineal y para una cuadrática, respectivamente. Aumentar el orden de los polinomios y el número de divisiones mejora la aproximación de funciones más complejas. Estos dos procesos son costosos, ya sea por la dificultad de evaluación del polinomio o la cantidad de puntos a evaluar. Si se utiliza un conjunto de puntos x_i que no están a una separación constante, se puede obtener la misma precisión con menos puntos y funciones de aproximación más simples mediante una cuadratura gaussiana.

B.1 Cuadratura gaussiana

Las cuadraturas gaussianas aproximan funciones por medio de una suma de polinomios ortogonales ponderados. De esta manera, una función $f(x)$ se puede aproximar por una suma de abscisas x_j por su peso correspondiente w_j

$$\int_a^b w(x)f(x)\mathrm{d}x \approx \sum_{j=1}^{N} \omega_j(x_j)f(x_j). \tag{B.3}$$

La función de peso w y los límites de integración a y b dependen del tipo de polinomios que se usa. En el caso de los polinomios de Legendre, el intervalo de aplicación es de $[-1, 1]$ y su función de peso es de 1, tal que

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \sum_{j=1}^{N} \omega_j(x_j)f(x_j). \tag{B.4}$$

Pero el intervalo de integración de interés no siempre coincide con el de los polinomios de Legendre, por lo que es necesario un cambio de variable para generalizar esta aproximación,

$$x = \frac{b + a + t(b - a)}{2} \quad \text{donde} \quad -1 \leq t \leq 1. \tag{B.5}$$

Entonces, la integración de una función en un intervalo arbitrario empleando polinomios de Legendre es

$$\int_a^b f(x)\mathrm{d}x = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b + a + t(b - a)}{2}\right)\mathrm{d}t \approx \frac{b-a}{2}\sum_{j=1}^{N} \omega_j f(t_j). \tag{B.6}$$

La cuadratura de Gauss-Legendre es buena para calcular integración de polinomios con orden de hasta $2N-1$. Para funciones más complicadas, se emplean polinomios diferentes, como los de Hermite o Chebyshev.

B.2 Mallado de Lebedev

El mallado de Lebedev está formado por puntos sobre una esfera unitaria que tienen una geometría octaédrica [47, 48]. Esta malla de puntos permite integrar sobre un ángulo sólido la parte radial de una función tridimensional. La integración de una función f sobre una esfera unitaria es:

$$I = \int f(\Omega)\mathrm{d}\Omega = \int_0^{2\pi} \mathrm{d}\psi \int_0^{\pi} f(\theta,\psi)\sin(\theta)\mathrm{d}\theta, \tag{B.7}$$

donde ψ y θ son los ángulos de las coordenadas esféricas. La variable r se puede integrar independientemente aunque la función dependa de r, porque la una integral de superficie se realiza bajo la condición $r = 1 = constante$. Esta integral se puede aproximar mediante una serie de nodos con sus pesos correspondientes,

$$I \approx 4\pi \sum_{i=1}^{N} w_i f(\theta_i,\psi_i). \tag{B.8}$$

La malla más sencilla de Lebedev incluye 6 puntos que son $(\pm1,0,0)$, $(0,\pm1,0)$ y $(0,0,\pm1)$ y se denotan como a_i^1. El siguiente mallado está conformado por los 6 puntos anteriormente descritos y 12 puntos más que están definido por las posibles permutaciones de coordenadas y cambios de signo de $(\sqrt{2}/2,\sqrt{2}/2,0)$, denotados por a_i^2, dando lugar a un mallado de 18 nodos. Para mallados más finos se requieren más puntos (b_i^k, c_i^k y d_i^k) cuya derivación está detallada en el trabajo de Levedev [47]. De manera general, este esquema de integración se puede expresar como:

$$\begin{aligned} I \approx I_N =& A_1 \sum_{i=1}^{6} f(a_i^1) + A_2 \sum_{i=1}^{12} f(a_i^2) + A_3 \sum_{i=1}^{8} f(a_i^3) \\ &+ \sum_{k=1}^{N_1} B_k \sum_{i=1}^{24} f(b_i^k) + \sum_{k=1}^{N_2} C_k \sum_{i=1}^{24} f(c_i^k) + \sum_{k=1}^{N_3 i} D_k \sum_{i=1}^{48} f(d_i^k). \end{aligned} \tag{B.9}$$

Las letras mayúsculas A_1, A_2, A_3, B_k, C_k, y D_k, representan los pesos y el número total de puntos es,

$$N = 6 + 12 + 8 + 24(N_1 + N_2) + 48N_3. \tag{B.10}$$

El número de nodos a utilizar depende del orden L, de la función a intergrar,

$$N = \frac{(L+1)^2}{3}. \tag{B.11}$$

El orden L indica que la integración de armónicos esféricos con momento angular hasta L se pueden calcular exactamente con dicha malla. A partir de esta restricción se obtiene una serie de ecuaciones que permiten calcular los pesos. Los nodos y los pesos están generalmente tabulados ya que son independientes de la función a evaluar.

B.3 Mallados moleculares de Becke

El mallado de Becke consiste en descomponer una función que depende de las coordenadas de todos los núcleos en componentes independientes para cada núcleo. Esto se lleva a cabo partiendo el espacio molecular en celdas difuminadas en las fronteras y centradas en cada átomo [40].

Primero, se consideran funciones de peso $\omega_n(\mathbf{r})$ para cada núcleo n de la molécula de estudio, tal que

$$\sum_n \omega_n(\mathbf{r}) = 1. \tag{B.12}$$

Además estas funciones deben valer uno cerca del núcleo al que pertenecen y decaer a cero en la vecindad de otros núcleos. Esto es para que el espacio esté dividido en celdas que son difusas, es decir, celdas continuas que se traslapan en los bordes o celdas difuminadas.

Cualquier función de interés $F(\mathbf{r})$ se puede expresar así:

$$F(\mathbf{r}) = \sum_n \omega_n(\mathbf{r})F(\mathbf{r}) = \sum_n F_n(\mathbf{r}), \tag{B.13}$$

donde

$$F_n(\mathbf{r}) = \omega_n(\mathbf{r})F(\mathbf{r}). \tag{B.14}$$

Entonces, la integración de esta función es

$$I = \int F(\mathbf{r})\mathrm{d}\mathbf{r} = \int \sum_n F_n(\mathbf{r})\mathrm{d}\mathbf{r} = \sum_n I_n, \tag{B.15}$$

donde

$$I_n = \int F_n(\mathbf{r})\mathrm{d}\mathbf{r}. \tag{B.16}$$

Cada una de las integrales "atómicas" (para denotar que están centradas en un núcleo) I_n se realiza mediante una integración numérica en coordenadas esféricas empleando una cuadratura gaussiana y el mallado de Lebedev. La integral I_n en coordenadas esféricas es:

$$\begin{aligned} I_n &= \int_0^\infty \mathrm{d}r \int_0^\pi \mathrm{d}\theta \int_0^{2\pi} \mathrm{d}\phi\, F_n(r,\theta,\psi)\, r^2 \sin\theta \\ &= \int_0^\infty \mathrm{d}r \int_0^\pi \mathrm{d}\theta \int_0^{2\pi} \mathrm{d}\psi\, \omega_n(r,\theta,\psi)F(r,\theta,\psi)\, r^2 \sin\theta. \end{aligned} \tag{B.17}$$

Aquí, $r^2 \sin\theta$ es la determinante Jacobiana que resulta del cambio de coordenadas. La integración sobre r y los ángulos es independiente. La integral radial se realiza con una cuadratura gaussiana y el mallado de Lebedev se utiliza para la integración de la parte angular sobre una esfera unitaria. Entonces, la integral I_n se puede reescribir como:

$$I_n = \int_0^\infty \mathrm{d}r \int \mathrm{d}\Omega\, \omega_n(r,\Omega)F(r,\Omega)\, r^2, \tag{B.18}$$

donde Ω es el ángulo sólido. Integrando por Lebedev se tiene que

$$I_n = 4\pi \int_0^\infty \mathrm{d}r \sum_{i=1}^{N_\Omega} \omega_i \omega_n(r,\theta_i,\psi_i)F(r,\theta_i,\psi_i)\, r^2. \tag{B.19}$$

Si ahora se integra la parte radial, se tiene que

$$I_n = 4\pi \sum_{i=1}^{N_\Omega} \sum_{j=1}^{N_r} \omega_i \omega_j \omega_n(r_j, \theta_i, \psi_i) F(r_j, \theta_i, \psi_i) r_j^2. \tag{B.20}$$

Generalmente se usa la cuadratura de Gauss-Chebyshev porque la obtención de las abscisas y los pesos es más sencilla, pero es necesario realizar este cambio de variable propuesto por Treutler y Ahlrichs [49],

$$r = \frac{1}{\ln 2} \ln\left(\frac{2}{1-x}\right). \tag{B.21}$$

Esto permite transformar el intervalo de integración de $[0, \infty)$ a $[-1, 1]$. Las abscisas y los pesos de Gauss-Chebyshev de segundo tipo se obtienen así [50, 51]:

$$x_i^n = 1 + \frac{2}{\pi}\left(1 + \frac{2}{3}\sin^2\left[\frac{i\pi}{n+1}\right]\right) \cdot \cos\left[\frac{i\pi}{n+1}\right] \sin\left[\frac{i\pi}{n+1}\right] - \frac{2i}{n+1}, \tag{B.22}$$

$$w_i^n = \frac{16}{3(n+1)} \sin^4\left[\frac{i\pi}{n+1}\right]. \tag{B.23}$$

Hasta este punto no se ha descrito nada respecto a la función de peso ω_n. Esta función necesita cumplir ciertas características para que el espacio molecular se pueda separar en celdas independientes con fronteras suavizadas. Estas celdas son parecidas a los poliedros de Voronoi, pero las caras no están definidas por un plano, sino por un gradiente continuo. Las coordenadas elípticas (λ, μ, ψ) son:

$$\lambda_{ij} = \frac{r_i + r_j}{R_{ij}}, \qquad \mu_{ij} = \frac{r_i - r_j}{R_{ij}}, \tag{B.24}$$

donde r_i y r_j son las distancias del punto del mallado a los átomos i y j, respectivamente. La distancia entre los núcleos es R_{ij}. Cuando λ es constante, se obtienen elipsoides con focos en los núcleos i y j. Si μ es constante, se obtienen hiperboloides. La coordenada μ tiene un rango de $[-1, 1]$. Cuando $\mu = -1$, se tiene un rayo que origina del centro i hacia el infinito sobre el eje internuclear. Si $\mu = 0$, la función resultante es el bisector perpendicular del eje internuclear. Las superficies para $\mu \in (-1, 0)$ y $\mu \in (0, 1)$ corresponden a hiperboloides. Cuando $\mu = 1$, ocurre algo similar al caso en el que $\mu = -1$, pero centrado en el núcleo j. En

los poliedros de Voronoi, se considera una función escalera para delimitar la celda, tal que

$$s(\mu_{ij}) = \begin{cases} 1, & -1 \leq \mu_{ij} \leq 0 \\ 0, & 0 < \mu_{ij} \leq 1 \end{cases} \tag{B.25}$$

Si esto se realiza para cada par de átomos, entonces la celda de cada uno de ellos, se obtiene multiplicando todas las hipérbolas que involucran el núcleo i,

$$P_i(\mathbf{r}) = \Pi_{j \neq i} s(\mu_{ij}). \tag{B.26}$$

$P_i(\mathbf{r})$ es la función de celda que define los límites del poliedro. Ahora, para suavizar las fronteras de la celda, se busca una función contínua. Esta función debe cumplir con requisitos similares a la función escalera:

$$\begin{aligned} &s(-1) = 1, \\ &s(1) = 0, \\ &\frac{\mathrm{d}s}{\mathrm{d}\mu}(-1) = \frac{\mathrm{d}s}{\mathrm{d}\mu}(1) = 0. \end{aligned} \tag{B.27}$$

La función más sencilla que satisface esto es

$$s(\mu) = \frac{1}{2}[1 - f(\mu)], \tag{B.28}$$

donde $f(\mu)$ es un polinomio, que es una composición de funciones $p(\mu)$ triple,

$$f(\mu) = p(p(p(\mu))). \tag{B.29}$$

Esto es para que se eliminen las cúspides de los otros centros, manteniendo fronteras suaves,

$$p(\mu) = \frac{3}{2}\mu - \frac{1}{2}\mu^3. \tag{B.30}$$

Para que se cumpla la ecuación 3.21, los pesos de Becke deben estar divididos por la suma de todos los P_i, incluyendo P_n,

$$\omega_n(\mathbf{r}) = \frac{P_n(\mathbf{r})}{\sum_m^M P_m(\mathbf{r})}. \tag{B.31}$$

C

Conjunto base auxiliar

A continuación se presentan los conjuntos de funciones gaussianas de la base auxiliar. La primera fila indica el número total de funciones en el conjunto base y las filas subsecuentes, de manera alternada, indican el momento angular de la función en una fila y el exponente y coeficiente de esta función en la siguiente.

El conjunto base del hidrógeno está conformado por 8 funciones gaussianas, 4 con momento angular 0 (s) y 4 con momento angular 1 (p). La primera función del conjunto del hidrógeno es una función s, con un exponente de 30.8223000 y un coeficiente de expansión de 1.00000000. El conjunto base completo del hidrógeno es:

```
H
    8
    0
    30.8223000    1.00000000
    1
    30.8223000    1.00000000
    0
    5.13705000    1.00000000
    1
    5.13705000    1.00000000
    0
    1.23289200    1.00000000
    1
    1.23289200    1.00000000
    0
    0.20548200    1.00000000
    1
    0.20548200    1.00000000
```

El conjunto base del carbono está conformado por 14 funciones gaussianas, 7 funciones s y 7 funciones p. La primera función del conjunto del carbono es una función s, con un exponente de 1490.79040000000 y un coeficiente de expansión de 1.00000000. El conjunto base completo del carbono es:

```
C
    14
    0
    1490.79040000000    1.00000000
    1
    1490.79040000000    1.00000000
    0
    298.158080000000    1.00000000
    1
    298.158080000000    1.00000000
    0
    74.5395200000000    1.00000000
    1
    74.5395200000000    1.00000000
    0
    23.2936000000000    1.00000000
    1
    23.2936000000000    1.00000000
    0
    4.65872000000000    1.00000000
    1
    4.65872000000000    1.00000000
    0
    1.45585000000000    1.00000000
    1
    1.45585000000000    1.00000000
    0
    0.29117000000000    1.00000000
    1
    0.29117000000000    1.00000000
```

Bibliografía

[1] P. Dirac, Proc. R. Soc. Lond. A **123**, 714 (1929).

[2] P. Hohenberg, W. Kohn, Phys. Rev. **136**, B864 (1964).

[3] A. Simões, K. Gavroglu, *Neither Chemistry nor Physics*, The MIT Press, London, 2012.

[4] I. S. Ufimtsev, T. J. Martínez, Computing in Science & Engineering **10**, 26 (2008).

[5] N. Luehr, I. Ufimtsev, T. Martinez, Dynamical Quadrature Grids: Applications in Density Functional Calculations, en *GPU Computing Gems*, editado por W.-M. W. Hwu, Capítulo 2, págs. 35--42, Elsevier, USA, 2011.

[6] K. Yasuda, Journal of Comput. Chem. **29**, 334 (2008).

[7] I. S. Ufimtsev, T. J. Martínez, J. Chem. Theory Comput. **5**, 2619 (2009).

[8] A. V. Titov, I. S. Ufimtsev, T. J. Martínez, J. Chem. Theory Comput. **9**, 213 (2013).

[9] H.-J. Werner, F. R. Manby, P. Knowles, J. Chem. Phys. **118**, 8149 (2003).

[10] T. Helgaker, P. Jørgensen, J. Olsen, *Molecular Electronic-Structure Theory*, John Wiley & Sons, Chichester, 2002.

[11] W. Koch, M. C. Holthausen, *A Chemist´s Guide to Density Functional Theory*, Wiley-VCH, 2001.

[12] R. G. Parr, W. Yang, *Density-Functional Theory of Atoms and Molecules*, Oxford University Press, New York, 1989.

[13] R. G. Woolley, B. T. Sutcliffe, Chem. Phys. Lett. **45**, 393 (1977).

[14] M. Levy, J. P. Perdew, V. Sahni, Phys. Rev. A **30**, 2745 (1984).

[15] V.V. Karasiev. S.B. Trickey, Frank discussion of the status of ground-state orbital-free {DFT}, en *Concepts of Mathematical Physics in Chemistry: A Tribute to Frank E. Harris - Part A*, editado por J. R. Sabin y R. Cabrera-Trujillo, volumen 71 of *Advances in Quantum Chemistry*, págs. 221--245, Academic Press, 2015.

[16] M. Chen, X.-W. Jiang, H. Zhuang, L.-W. Wang, E. A. Carter, J. Chem. Theory. Comput. **12**, 2950 (2016).

[17] W. Kohn, J. Sham, Phys. Rev. **137**, A1697 (1965).

[18] R. Flores-Moreno, Parakata versión xiv-v, http://gts.sourceforge.net/, 2015.

[19] E. J. Baerends, D. E. Ellis, P. Ros, Chem. Phys. **2**, 41 (1973).

[20] H. Sambe, R. H. Felton, J. Chem. Phys. **62**, 1122 (1975).

[21] B. I. Dunlap, J. W. D. Connolly, J. R. Sabin, J. Chem. Phys. **71**, 4993 (1979).

[22] B. I. Dunlap, N. Rösch, S. B. Trickey, Mol. Phys. **108**, 3167 (2010).

[23] R. van Leeuwen, E. J. Baerends, Phys. Rev. A **49**, 2421 (1994).

[24] J. P. Perdew, K. Schmidt, Jacob's ladder of density functional approximations for the exchange-correlation energy, en *Density Functional Theory and Its Application to Materials*, editado por P. G. V. V. Doren, C. V. Alsenoy, págs. 1--20, AIP, Melville, New York, 2001.

[25] P. A. M. Dirac, Proc. Camb. Phil. Soc. **26**, 376 (1930).

[26] A. D. Becke, Phys. Rev. A **38**, 3098 (1988).

[27] D. C. Langreth, M. J. Mehl, Phys. Rev. B **28**, 1809 (1983).

[28] P. M. W. Gill, Adv. Quantum Chem. **25**, 141 (1994).

[29] J. W. Mintmire, B. I. Dunlap, Phys. Rev. A **25**, 88 (1982).

[30] A. M. Köster, J. U. Reveles, J. M. del Campo, J. Chem. Phys. **121**, 3417 (2004).

[31] A. M. Köster, J. Chem. Phys. **104**, 4114 (1996).

[32] A. M. Köster, J. M. del Campo, F. Janetzko, B. Zuniga-Gutierrez, J. Chem. Phys. **130**, 114106 (2009).

[33] D. N. Laikov, Chem. Phys. Lett. **281**, 151 (1997).

[34] A. Szabo, N. S. Ostlund, *Modern Quantum Chemistry*, Dover Publications Inc., Mineola, 1996.

[35] J. A. Pople, R. K. Nesbet, J. Chem. Phys. **22**, 571 (1954).

[36] A. M. Köster, J. Chem. Phys. **118**, 9943 (2003).

[37] S. Obara, A. Saika, J. Chem. Phys. **84**, 3963 (1986).

[38] L. E. McMurchie, E. R. Davidson, J. Comput. Phys. **44**, 289 (1981).

[39] M. Head-Gordon, J. A. Pople, J. Chem. Phys. **89**, 5777 (1988).

[40] A. D. Becke, J. Chem. Phys. **88**, 2547 (1988).

[41] G. E. Moore, Procs. of the IEEE **86**, 82 (1994).

[42] J. Sanders, E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, Pearson Education, Inc., Michigan, 2010.

[43] Cuda toolkit v8.0.61, http://docs.nvidia.com/cuda/index.html.

[44] J. Rys, M. Dupuis, H. F. King, J. Comput. Chem. **4**, 154 (1983).

[45] A. Asadchev, V. Allada, J. Felder, B. M. Bode, M. S. Gordon, T. L. Windus, J. Chem. Theory Comput. **6**, 696 (2010).

[46] K. F. Riley, M. P. Hobson, S. J. Bence, *Mathematical Methods for Physics and Engineering*, Cambridge University Press, New York, USA, 2002.

[47] I. V. Lebedev, Siberian Math. J. **18**, 99 (1977).

[48] X. Wang, T. Carrington Jr., J. Theo. Comput. Chem. **2**, 599 (2003).

[49] R. Ahlrichs, O. Treutler, J. Chem. Phys. **102**, 346 (1994).

[50] J. M. Pérez-Jordá, E. San-Fabián, F. Moscardó, Comput. Phys. Comm. **70**, 271 (1992).

[51] J. M. Pérez-Jordá, A. D. Becke, E. San-Fabián, J. Chem. Phys. **100**, 6520 (1994).