



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN CIENCIAS QUÍMICAS

Método directo con reducción de accesos a memoria en la evaluación de integrales de repulsión electrónica moleculares mediante funciones auxiliares en EP2

TESIS

PARA OPTAR POR EL GRADO DE

MAESTRO EN CIENCIAS

PRESENTA

Q. Juan Felipe Huan Lew Yee

TUTOR

Dr. Jorge Martín del Campo Ramírez
Facultad de Química, Universidad Nacional Autónoma de México

Ciudad de México, Febrero, 2020



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Método directo con reducción de accesos a memoria
en la evaluación de integrales de repulsión electrónica
moleculares mediante funciones auxiliares en EP2

Juan Felipe Huan Lew Yee

Universidad Nacional Autónoma de México



Departamento de Física y Química Teórica

Facultad de Química

Método directo con reducción de accesos a memoria en la evaluación de integrales de repulsión electrónica moleculares mediante funciones auxiliares en EP2

Juan Felipe Huan Lew Yee

- Presidente* Dr. Emilio Orgaz Baqué
Facultad de Química
Universidad Nacional Autónoma de México
- Vocal* Dr. Serguei Fomine
Instituto de Investigación en Materiales
Universidad Nacional Autónoma de México
- Vocal* Dr. Andreas M. Köster
Centro de Investigaciones Avanzadas
Instituto Politécnico Nacional
- Vocal* Dr. J. Jesús Hernández Trujillo
Facultad de Química
Universidad Nacional Autónoma de México
- Secretario* Dr. José Eduardo Barrios Vargas
Facultad de Química
Universidad Nacional Autónoma de México
- Asesor* Dr. Jorge Martín del Campo Ramírez
Facultad de Química
Universidad Nacional Autónoma de México

Juan Felipe Huan Lew Yee

Método directo con reducción de accesos a memoria en la evaluación de integrales de repulsión electrónica moleculares mediante funciones auxiliares en EP2

2020

Jurado: Dr. Emilio Orgaz Baqué, Dr. Serguei Fomine, Dr. Andreas M. Köster, Dr. J. Jesús Hernández Trujillo, Dr. José Eduardo Barrios Vargas

Asesor: Dr. Jorge Martín del Campo Ramírez

Universidad Nacional Autónoma de México

Facultad de Química

Departamento de Física y Química Teórica

Av. Universidad 3000, Ciudad de México, C.P. 04510

Acrónimos

- ERI** Integral de Repulsión Electrónica / *Electron Repulsion Integral*
- AO-ERI** Integral de Repulsión Electrónica en Orbital Atómico / *Atomic-Orbital Electron Repulsion Integrals*
- MO-ERI** Integral de Repulsión Electrónica en Orbital Molecular / *Molecular-Orbital Electron Repulsion Integrals*
- NR** Método de Newton-Raphson
- EPT** Teoría del propagador del electrón / *Electron Propagator Theory*
- EP2** Propagador del electrón de segundo orden en su aproximación diagonal
- P3** Propagador del electrón de tercer orden parcial
- P3+** Propagador del electrón de tercer orden parcial renormalizado
- OVGF** *Outer Valence Green Function*
- ModChol** Descomposición Modificada de Cholesky / *Modified Cholesky Decomposition*
- SVD** Descomposición en Valores Singulares / *Singular Value Decomposition*
- TED** Descomposición Truncada de Valores Propios / *Truncated Eigenvalue Decomposition*
- DF** Ajuste de la densidad / *Density Fitting*
- LDF** Ajuste local de la densidad / *Local Density Fitting*
- EBE** Energía de unión del electrón / *Electron Binding Energy*
- RAM** Memoria de acceso aleatorio / *Random Access Memory*

Agradecimientos

A mi familia, Leticia Yee López, Felipe Lew León y Victoria Yee López, por su apoyo en todos los aspectos de mi vida, y por acompañarme a lo largo del camino académico que he recorrido.

A Dulce Consuelo Guzmán Ocampo, por su compañía en lo personal y en lo académico, por ser mi equipo en la vida, y por sus sugerencias durante la elaboración de este trabajo.

Al Dr. Jorge Martín del Campo Ramírez, por la detallada lectura de este trabajo, los debates académicos, por sus valiosos consejos y apoyo, así como por compartirme de sus conocimientos y guiarme durante la construcción de los míos.

A mis amigos, Luis Camacho, Carlos Fontes, David Colorado, Luis Valdez, Romina Soto, Stefani Paulin y Aura Fuentes, por la convivencia que hemos tenido a lo largo de estos años.

A mis amigos, Jose María Castillo, Alejandra del Río, Edith Leal y Andrés Marmolejo, por los buenos momentos y por las estimulantes charlas académicas.

A mis compañeros y amigos en el grupo, Xiaomin Huang, Rodrigo Cortés, Alfonso Esqueda, Neftalí, Victor y Andrés, por sus sugerencias y convivencia.

Al Dr. Roberto Flores Moreno, por su apoyo académico, la colaboración y por permitirme aprender de su código.

Al Dr. José Luis Morales Pérez, por el apoyo en la parte matemática de este trabajo.

A los miembros del jurado, por su cuidadosa revisión y gratas sugerencias respecto a este trabajo.

A mis profesores, por presentarme nuevos retos y orientarme durante mi formación profesional.

A la Universidad Nacional Autónoma de México, a la Facultad de Química y al Posgrado de Ciencias Químicas, por todos los recursos brindados durante mi formación.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca de maestría correspondiente al CVU 867718.

A la Dirección General de Cómputo y de Tecnologías de Información y Comunicación (DGETIC) por los recursos de cómputo otorgados bajo la clave LANCAD-UNAM-DGETIC-270.

Al proyecto CONACyT con clave CB-2016-282791.

Al Programa de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT) bajo el proyecto con clave IN114418.

Resumen

En este trabajo se presenta un algoritmo directo para el propagador del electrón de segundo orden en su aproximación diagonal, capaz de ser ejecutado cuando no hay suficiente memoria, sin hacer uso de disco, y generando de manera eficiente las integrales de repulsión electrónica en orbital molecular. Mediante la aproximación del ajuste de la densidad se introduce una base auxiliar, la cual permite evaluar integrales más simples, y genera arreglos más sencillos de manejar. La implementación se basa en una contracción intermedia que permite la división en lotes, al mismo tiempo que minimiza el proceso de recálculo de integrales. De manera adicional, se usa la descomposición modificada de Cholesky para dar estabilidad numérica a la metodología, y se aprovecha la localización orbital para disminuir el número de lotes cuando es posible. El algoritmo generado representa una oportunidad importante para trascender las limitaciones de memoria, aprovechar el paralelismo y poder analizar sistemas de interés de gran tamaño.

Abstract

In this work, a direct algorithm for second order electron propagator in its diagonal approximation is presented. The algorithm allows executing calculations when not enough memory is available, without using disk, and generating efficiently the molecular orbital electron repulsion integrals. Density fitting approximation is used to introduce an auxiliary basis, which leads to simpler integrals and easier to handle arrays. The implementation is based on an intermediate contraction that can be evaluated in batches, and minimize the reevaluation of integrals. In addition, the modified Cholesky decomposition is used to provide stability to the fitting process. The methodology takes advantage of the orbital localization to decrease the number of batches when possible. The algorithm represents an important opportunity to overcome memory limitations, to exploit parallelization, and to study huge systems of interest.

Producción

Los resultados aquí presentados contribuyeron a la generación de los siguientes trabajos:

- *Reduced RAM access algorithm for molecular orbital electron repulsion integrals with resolution of the identity applied in post Hartree-Fock methods*, el cuál fue presentado en modalidad de **póster** en la American Chemical Society FALL 2019 NATIONAL MEETING & EXPO llevada a cabo en San Diego, California, el 27 de agosto de 2019.
- *Implementación semidirecta de EPT2 con base auxiliar*, el cuál fue presentado en modalidad de **plática** en la XVIII Reunión Mexicana de Fisicoquímica Teórica en Toluca, Estado de México, el 24 de octubre de 2019.
- *Asymmetric density fitting with modified Cholesky decomposition applied to second order electron propagator*, el cuál ha sido enviado como **artículo** a la revista Journal of Chemical Theory and Computation y se encuentra en estado Accepted.
- El método generado se ha integrado al **software** Nagual para realizar cálculos de propagador del electrón de segundo orden en su aproximación diagonal.

Índice

1	Introducción	1
2	Teoría	5
2.1	Propagador del Electrón de segundo orden	6
2.2	Integrales de Repulsión Electrónica	6
2.3	Descomposición Modificada de Cholesky	10
3	Algoritmo directo-w	13
3.1	Pseudocódigo	13
3.2	Esquema de paralelización	17
4	Resultados	18
4.1	Estabilidad numérica de la matriz métrica	18
4.2	Localización orbital	21
4.3	Desempeño con respecto al número de lotes	24
4.4	Análisis de ventajas del algoritmo	25
5	Conclusiones y Perspectivas	28
6	Reflexión Personal	29
A	Newton-Raphson en EP2	31
B	Análisis de estabilidad numérica de la matriz métrica	33
C	Otras implementaciones	35
C.1	Implementaciones convencional e in-core	35
C.2	Implementación directa simple	37
	Referencias	39

Introducción

” *Advances in computer technology extend the applicability of computational procedures... They often necessitate the development of new algorithms better adapted to the new hardware... an example concerns the growing disparity of CPU and I/O performance, which in all probability will increase in the future.*

— R. Ahlrichs, M. Bär, M. Häser, H. Horn, y C. Kölmel
Chem. Phys. Lett. 162, 165 (1989).

La teoría del propagador del electrón (EPT) permite calcular energías de ionización y afinidades electrónicas que corrigen a las predichas por el teorema de Koopmans¹ al incluir efectos de relajación orbital y de correlación electrónica; así como orbitales de Dyson, que son funciones de una partícula que proporcionan información sobre probabilidades de transición.²⁻⁶ Las ecuaciones corresponden a la función de Green de muchos cuerpos, y el tratamiento se puede realizar introduciendo un potencial de autoenergía.⁷ La aproximación al potencial genera varias formulaciones de EPT, siendo las más populares la aproximación a segundo orden, la parcial a tercer orden⁸⁻¹⁰ (P3 y P3+), y la *outer valence Green's function*^{11,12} (OVGF); en particular, este trabajo se enfoca en la aproximación diagonal del propagador del electrón de segundo orden (EP2).

La EPT es un método post Hartree-Fock que utiliza integrales de repulsión electrónica en representación de orbital molecular (MO-ERIs), las cuales se calculan como contracciones de integrales de repulsión electrónica en representación de orbital atómico (AO-ERIs). La evaluación de las MO-ERIs constituye la parte más demandante de EP2; por lo tanto, este trabajo se centra en dar un manejo adecuado a la generación de las MO-ERIs requeridas. En el algoritmo propuesto se introduce una base auxiliar mediante la aproximación de ajuste de la densidad para utilizar integrales de tres y dos centros, las últimas se agrupan en una matriz métrica, \mathbf{G} , y formalmente debe de encontrarse su inversa, pero esto se evita utilizando la descomposición modificada

de Cholesky (ModChol)^a. También se aprovecha la localización orbital para reducir, cuando es posible, el escalamiento del algoritmo y el número de AO-ERIs a manejar, mediante la técnica de ajuste local de la densidad (LDF)¹⁶.

Los algoritmos de los cálculos de estructura electrónica caen en cuatro categorías generales: los algoritmos convencionales, que guardan las integrales en disco; los algoritmos *in-core*, que guardan las integrales en RAM; los algoritmos directos, que recalculan las integrales cuando es necesario; y los algoritmos semidirectos, que combinan el recálculo de las integrales con el guardado en disco.^{17,18} Los algoritmos *in-core* suelen ser muy eficientes, ya que calculan las integrales una sola vez y utilizan la memoria RAM que presenta una alta velocidad de entrada y salida (I/O); sin embargo, su ejecución solo es posible mientras haya suficiente memoria RAM. Los algoritmos convencionales permiten extender la ventaja de guardar las integrales y evitar su recálculo computacionalmente costoso, pero debido a la lenta I/O de disco pueden generar problemas de latencia en sistemas moleculares de gran tamaño donde se deben leer archivos de integrales muy grandes, lo que se traduce en una desventaja para la paralelización.^{19,20} En cambio, los algoritmos directos evitan el cuello de botella de disco intercambiándolo por una mayor carga de operaciones de punto flotante causadas por el recálculo de integrales.^{21,22} La posibilidad de implementar una teoría de varias formas distintas responde a la búsqueda de maneras de tomar el mayor provecho de las características computacionales de la época. En la actualidad, los sistemas computacionales poseen una cantidad de memoria grande, aunque finita, y procesadores con una alta capacidad de cómputo; además, la capacidad de cómputo en paralelo en la CPU ha crecido notablemente, encontrándose típicamente computadoras personales con 4, 8, y hasta 16 *cores* de CPU, y sistemas de cómputo profesionales con 16, 24 y hasta 72 *cores* de CPU. Dadas las características de los sistemas computacionales actuales, los algoritmos directos representan la alternativa más viable para el análisis de sistemas moleculares grandes que requieran más RAM de la disponible.

La combinación de los métodos directos con aproximaciones que mejoren el uso de los recursos disponibles y disminuyan las operaciones aritméticas debe llevar a un proceso de cómputo más eficiente.²³ Las ventajas de la aproximación del ajuste de la densidad (*Density Fitting*, DF, por sus siglas en inglés) en el propagador del electrón ya han sido descritas y se relacionan con un mejor manejo de memoria, con la generación de intermediarios de contracción entre AO-ERIs y MO-ERIs más fáciles de manejar, y AO-ERIs más simples de evaluar.²⁴ La combinación de las AO-ERIs con base auxiliar

^a ModChol se refiere a la factorización de una matriz simétrica indefinida propuesta por Cheng y Higham¹³ y no a la descomposición de Cholesky de las AO-ERIs de cuatro centros propuesta por Beebe¹⁴ y ya aplicada al propagador del electrón¹⁵.

introducidas por el ajuste de la densidad con el manejo mediante un algoritmo directo representa la mejor forma de trascender las limitaciones computacionales de memoria RAM disponible.^{25,26} En este trabajo se usa una contracción intermedia que permite hacer un uso eficiente de la memoria, ya que requiere de menos accesos que las AO-ERIs. La contracción intermedia se almacena en un tensor generado en lotes independientes, tal que cada lote puede ajustarse a la memoria RAM disponible, esto permite tomar el máximo provecho de la RAM como lo haría un algoritmo *in-core*, al mismo tiempo que se evitan las limitaciones de tener una memoria finita²⁷. Para minimizar el proceso de recálculo, las integrales se calculan e inmediatamente se añaden a todos los elementos a los que contribuyen en las contracciones intermedias; esta es una forma particular de ejecución en la que las integrales guían las operaciones a realizar, y se denomina *integral driven*.²⁰

En el contexto de los tipos de algoritmos y el manejo de las integrales, deben mencionarse de manera especial algunos algoritmos que han sido reportados para la EPT. Han sido reportados algoritmos semidirectos con integrales de cuatro centros²⁸⁻³⁰, estos tienen la idea de aprovechar la capacidad de cómputo de los CPUs preservando el ahorro de cálculo al guardar las integrales en disco, pero con la capacidad de cómputo actual, la latencia de disco es más importante que las operaciones aritméticas ahorradas en sistemas moleculares grandes. Las implementaciones previas de la teoría del propagador con base auxiliar han sido del tipo convencional^{24,31}, así que si bien disminuyen el escalamiento aritmético y de memoria, también introducen el cuello de botella de I/O de disco; en particular una de ellas ya aprovecha la localización orbital para reducir el escalamiento³¹. Por último, las ventajas de DF para evitar o disminuir el I/O de disco ya han sido exploradas en otras teorías como *Coupled Cluster*²⁶ y *Moller-Plesset*²⁵, por lo que su aplicación en EPT representa una potencial ventaja que debe ser explotada.

El objetivo general de este trabajo es proponer un algoritmo que utilice integrales con base auxiliar para generar de manera eficiente y aprovechando los recursos computacionales disponibles, las MO-ERIs requeridas para los cálculos de EP2 en sistemas grandes que no quepan en memoria. Para lograr esto, deben considerarse como objetivos particulares el evitar el uso de disco, que se convierte en un cuello de botella en sistemas grandes; el generar un sistema de evaluación por lotes, para realizar el cálculo en porciones que se adapten a la memoria RAM disponible; evitar el guardado de las AO-ERIs, pues estas requieren demasiada memoria y su división en lotes no es eficiente; y el minimizar la reevaluación de las ERIs pues esto también representa un tiempo de cómputo importante. El esquema descrito ha sido programado en paralelo utilizando OpenMP para aprovechar la capacidad de cómputo disponible.

Ya que la implementación propuesta aprovecha de manera eficiente los recursos computacionales, al mismo tiempo que mantiene estabilidad numérica, resulta en una metodología robusta que permite realizar cálculos de sistemas moleculares que superen la memoria disponible. La capacidad de analizar sistemas de gran tamaño de forma eficiente desde la construcción del algoritmo facilita el uso de la teoría y aumenta sus aplicaciones potenciales. Aunque EP2 no es por sí misma una metodología con un alto grado de exactitud, es requerida para el cálculo de órdenes superiores que presentan resultados de mayor calidad. Por otra parte, los principios considerados en la implementación propuesta para EP2 pueden ser generalizados a tratamientos de orden superior.

“...it becomes increasingly complicated to design cost-effective direct algorithms, principally because direct schemes operate best when atomic orbital integrals are used... correlation procedures, however, are usually performed in the molecular orbital (MO) basis.

— A.P. Rendell and T.J. Lee
J. Chem. Phys. 101, 400 (1994).

En el presente capítulo se presenta el fundamento teórico de la teoría del propagador del electrón sobre el que se construye la implementación propuesta. Las ecuaciones pueden ser desarrolladas a partir de la función de Green de Hartree-Fock mediante la ecuación de Dyson, utilizando un potencial denominado de auto-energía. La evaluación del potencial en términos de la base lleva a una representación matricial Σ , en la que usualmente los términos diagonales son los más importantes. En la aproximación diagonal, las energías de ionización y afinidades electrónicas, o de forma general denominadas energías de unión de electrón (*Electron Binding Energies*, EBE's, por sus siglas en inglés), se calculan como los valores que hacen cero al determinante de la función de Green de muchos cuerpos considerando que Σ es una matriz diagonal. Mediante esta aproximación se pueden obtener las EBE's de manera eficiente evitando el cálculo de los orbitales de Dyson. Los elementos de Σ se pueden aproximar de varias maneras, una de ellas es con la aproximación perturbativa, que puede ser aplicada a varios órdenes, siendo la más sencilla la perturbación de segundo orden.

A continuación se expone el álgebra de EP2, considerando como punto central la generación de las MO-ERIs. Se introduce la aproximación de ajuste de la densidad para las integrales, y la teoría que respalda a la descomposición modificada de Cholesky para obtener estabilidad numérica en la implementación. La descripción sigue el orden de los pasos para la construcción de las MO-ERIs requeridas para evaluar EP2. Las ecuaciones aquí mostradas corresponden al desarrollo general con orbitales de espín, y debe considerarse que varias de ellas se vuelven cero al integrar el espín.

2.1 Propagador del Electrón de segundo orden

La teoría del propagador del electrón en su aproximación diagonal (EP2) calcula las EBE's, ω , como una corrección a la energía orbital, ε_p , dada por un término diagonal de la matriz de auto-energía, $\Sigma_{pp}(\omega)$, es decir

$$\omega = \varepsilon_p + \Sigma_{pp}(\omega) . \quad (2.1)$$

Como el potencial de auto-energía depende de las EBE's, la ecuación (2.1) se soluciona de manera iterativa. Usualmente se requieren pocas iteraciones de Newton-Raphson (NR) simple para lograr la convergencia utilizando ε_p como valor inicial. Las ecuaciones relacionadas con NR en EP2 se muestran en el Anexo A.

El término Σ_{pp} se encuentra de manera perturbativa, y en EP2 toma la forma³

$$\Sigma_{pp}^{(2)} = \sum_{i,a < b} \frac{|(pa|ib) - (pb|ia)|^2}{\omega + \varepsilon_i - \varepsilon_a - \varepsilon_b} + \sum_{a,i < j} \frac{|(pi|aj) - (pj|ai)|^2}{\omega + \varepsilon_a - \varepsilon_i - \varepsilon_j} , \quad (2.2)$$

donde el índice p corresponde al orbital a analizar, y los índices i, j y a, b indican orbitales ocupados y virtuales respectivamente; al primer término de la ecuación (2.2) se le denomina 2ph y contiene MO-ERIs del tipo p-virtual-ocupado-virtual, y al segundo se le denomina 2hp y contiene MO-ERIs del tipo p-ocupado-virtual-ocupado. Es importante destacar que no se necesitan todas las MO-ERIs, sino solo combinaciones específicas de orbitales ocupados y virtuales. Cuando las MO-ERIs no caben en memoria, entonces deben de evaluarse en lotes, para hacer esto debe de cumplirse que la MO-ERI $(pq|st)$ esté en el mismo lote que la MO-ERI $(pt|sq)$, donde q, s , y t representan orbitales que pueden ser tanto ocupados como virtuales; esta condición se requiere para poder hacer las restas del numerador de los términos 2ph y 2hp, que corresponden a la anti-simetrización de las integrales. Esto significa que dado un tercer índice, s , se debe tener acceso al par de MO-ERIs con el segundo y el cuarto índice, q y t , intercambiados.

2.2 Integrales de Repulsión Electrónica

Las MO-ERIs presentes en la ecuación (2.2) se pueden expresar como contracciones de AO-ERIs según

$$(pa|ib) = \sum_{\mu}^N C_{\mu p} \sum_{\nu}^N C_{\nu a} \sum_{\sigma}^N C_{\sigma i} \sum_{\lambda}^N C_{\lambda b} (\mu\nu|\sigma\lambda) ,$$

$$(pi|aj) = \sum_{\mu}^N C_{\mu p} \sum_{\nu}^N C_{\nu i} \sum_{\sigma}^N C_{\sigma a} \sum_{\lambda}^N C_{\lambda j} (\mu\nu|\sigma\lambda) , \quad (2.3)$$

donde los índices μ , ν , σ y λ representan orbitales atómicos, y N es el número de funciones de base. La AO-ERI dada por $(\mu\nu|\sigma\lambda)$ constituye una integral de cuatro centros que se definen por

$$(\mu\nu|\sigma\lambda) = \int d\mathbf{x}_1 d\mathbf{x}_2 \mu(\mathbf{x}_1) \nu(\mathbf{x}_1) r_{12}^{-1} \sigma(\mathbf{x}_2) \lambda(\mathbf{x}_2) , \quad (2.4)$$

aquí \mathbf{x} contiene las coordenadas espaciales y el espín, y se ha asumido que todos los orbitales son reales. Tanto las MO-ERIs como las AO-ERIs constituyen las partes más costosas del proceso de cómputo, la primera por su alto escalamiento, y la segunda por la alta cantidad de operaciones aritméticas que tienen que hacerse para evaluar cada una; por lo tanto, se han introducido diversas aproximaciones para reducir su costo computacional. En particular, aquí se utiliza el ajuste de la densidad para reemplazar las AO-ERIs de cuatro centros presentes en la ecuación (2.3) por AO-ERIs más sencillas de tres y dos centros.^{32,33} La evaluación de MO-ERIs se transforma en

$$\begin{aligned} (pa|ib) &= \sum_{\mu}^N C_{\mu p} \sum_{\nu}^N C_{\nu i} \sum_{\sigma}^N C_{\sigma a} \sum_{\lambda}^N C_{\lambda b} \sum_P^{N_{aux}} (\mu\nu|P) \sum_Q^{N_{aux}} G_{PQ}^{-1}(Q|\sigma\lambda) , \\ (pi|aj) &= \sum_{\mu}^N C_{\mu p} \sum_{\nu}^N C_{\nu a} \sum_{\sigma}^N C_{\sigma i} \sum_{\lambda}^N C_{\lambda j} \sum_P^{N_{aux}} (\mu\nu|P) \sum_Q^{N_{aux}} G_{PQ}^{-1}(Q|\sigma\lambda) , \end{aligned} \quad (2.5)$$

donde los índices P y Q denotan la base auxiliar, y N_{aux} es el número de funciones auxiliares. Las AO-ERIs de tres centros se definen por

$$(\mu\nu|P) = \int d\mathbf{x}_1 d\mathbf{x}_2 \mu(\mathbf{x}_1) \nu(\mathbf{x}_1) r_{12}^{-1} P(\mathbf{x}_2) , \quad (2.6)$$

y el término \mathbf{G} es una matriz métrica formada por las AO-ERIs de dos centros³⁴

$$G_{PQ} = (P|Q) = \int d\mathbf{x}_1 d\mathbf{x}_2 P(\mathbf{x}_1) r_{12}^{-1} Q(\mathbf{x}_2) . \quad (2.7)$$

Las sumas de la ecuación (2.5) se deben reordenar para adecuarse a la generación de MO-ERIs anti-simetrizadas. El esquema usado en este trabajo busca una evaluación por lotes que supere las limitaciones de RAM; por tanto, es necesario que dentro de un lote haya bloques del tipo p -virtual- i -virtual para poder realizar la operación del numerador en el término 2ph, y del tipo p -ocupado- a -ocupado para poder realizar la

operación del numerador en el término $2hp$; esto sugiere que los bloques de MO-ERIs deben construirse en función de la identidad del tercer índice de la MO-ERI.

Una consecuencia directa de introducir la base auxiliar en las MO-ERIs es que se separan los primeros dos índices, *bra*, de los últimos dos índices, *ket*; esto permite darle un tratamiento diferente a cada parte, como se muestra en la Figura 2.1.

El orbital estudiado, p , constituye un índice fijo en el *bra*, por tanto, su contracción a orbital molecular desde las AO-ERIs de tres centros se realiza primero, ya que reduce el escalamiento de los pasos subsecuentes. La contracción de este índice se representa mediante

$$(p\nu|P) = \sum_{\mu}^N C_{\mu p}(\mu\nu|P) . \quad (2.8)$$

Posteriormente, se contrae a orbital molecular el segundo índice del *bra* mediante

$$(pq|P) = \sum_{\nu}^N C_{\nu q}(p\nu|P) , \quad (2.9)$$

esto genera una matriz temporal, la cual se multiplica por el inverso de la matriz métrica

$$(\widetilde{pq}|Q) = \sum_P^{N_{aux}} (pq|P)G_{PQ}^{-1} . \quad (2.10)$$

La tilde en la ecuación (2.10) indica que formalmente el objeto no es una integral, ya que se ha multiplicado por la matriz métrica. Por conveniencia, las operaciones anteriores se renombrarán para futuras referencias, como

$$z_{\nu P}^p = (p\nu|P) , \quad (2.11)$$

$$y_{qP}^p = (pq|P) , \quad (2.12)$$

$$x_{qQ}^p = (\widetilde{pq}|Q) , \quad (2.13)$$

es importante notar que debido a que p es un índice fijo, \mathbf{x} , \mathbf{y} , \mathbf{z} , son matrices, y por tanto se pueden calcular y almacenar sin consideraciones adicionales.

Por otra parte, el *ket* no se relaciona con ningún índice fijo, y tiene el escalamiento aritmético y de memoria más importante del procedimiento. Sea la MO-ERI $(pq|st)$, su tercer índice, s , está en el *ket*, y se contrae de las AO-ERIs de tres centros generando una ERI intermedia dada por

$$(Q|s\lambda) = \sum_{\sigma}^N C_{\sigma s}(Q|\sigma\lambda) . \quad (2.14)$$

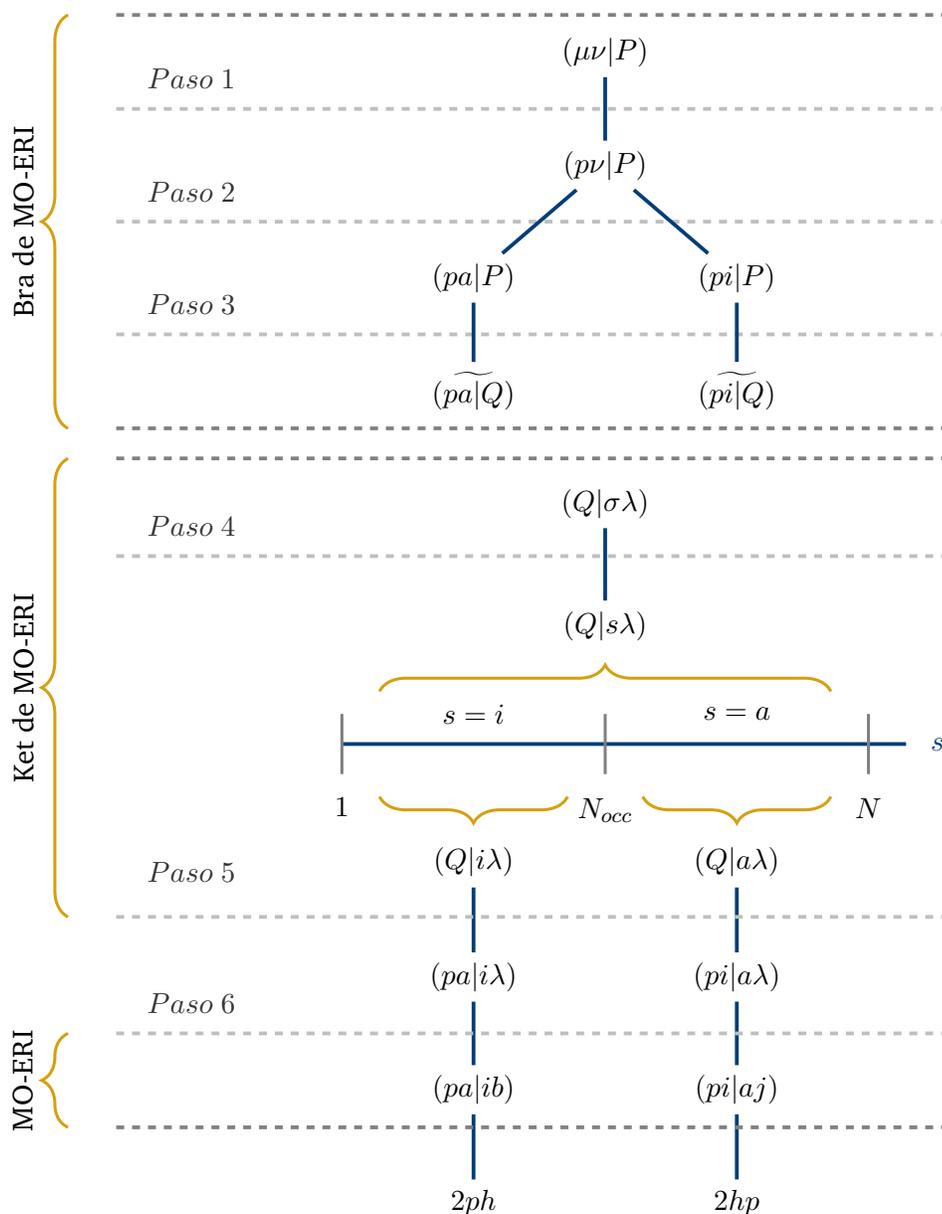


Figura 2.1. Esquema de evaluación de las MO-ERIs anti-simetrizadas requeridas para EP2 utilizando integrales con base auxiliar. Los pasos del 1 al 3 corresponden a la contracción del bra de las MO-ERIs, y la multiplicación por la matriz métrica. Los pasos del 4 al 6 corresponden a la contracción del ket y a la generación de los bloques de MO-ERIs requeridos por los numeradores de los términos 2ph y 2hp. Es importante destacar que el paso 4 se realiza por lotes, donde cada lote contiene un conjunto de valores del índice s , y en función de estos valores se realiza la contracción del segundo y cuarto índice de las MO-ERIs en el paso 5 y 6, esto se describe con más detalle en el siguiente capítulo.

La ERI intermedia ($Q|s\lambda$) es clave en la implementación, por lo que se le denominará tensor w para referencia, dado por

$$w_{\lambda Q}^s = (Q|s\lambda) , \quad (2.15)$$

en particular, la notación de los índices de w se ha utilizado para resaltar que el tensor de tres dimensiones será tratado como un conjunto de matrices.

Al tomar en cuenta las consideraciones analizadas y sustituirlas en la ecuación (2.5), la evaluación de las MO-ERIs requeridas para EP2 usando base auxiliar se realiza mediante

$$\begin{aligned} (pa|ib) &= \sum_{\lambda}^N C_{\lambda b} \sum_Q^{N_{aux}} x_{aQ}^p w_{\lambda Q}^i , \\ (pi|aj) &= \sum_{\lambda}^N C_{\lambda j} \sum_Q^{N_{aux}} x_{iQ}^p w_{\lambda Q}^a , \end{aligned} \quad (2.16)$$

donde se ha hecho explícito que los pasos a seguir dependen de si el tercer índice, s , corresponde a un orbital ocupado, i , o a un orbital virtual, a .

2.3 Descomposición Modificada de Cholesky

La introducción de la base auxiliar tiene como consecuencia la evaluación de AO-ERIs de dos centros contenidas en la matriz métrica \mathbf{G} , cuya inversa debe de usarse en la ecuación (2.10). Sin embargo, debido a dependencias lineales y a que la precisión numérica es finita, la representación computacional de la matriz \mathbf{G} puede estar mal condicionada y estar cerca de ser singular, es decir que su inversa no exista o que su cálculo genere un error numérico importante. Una solución común es descomponer la matriz métrica en sus valores propios (λ) y vectores propios, y volver cero los inversos de aquellos valores negativos o menores a una tolerancia junto con sus vectores propios asociados. Este método es conocido como descomposición truncada de valores propios (*Truncated Eigenvalue Decomposition, TED, por sus siglas en inglés*). Otro método común es obtener la inversión por medio de la descomposición en valores singulares (*Singular Value Decomposition, SVD, por sus siglas en inglés*), en la que se vuelven cero los inversos de los valores singulares (σ) que estén por debajo de cierta tolerancia. Aunque estas soluciones garantizan la inversión de la matriz métrica, también generan una matriz singular, \mathbf{G}^{-1} , en consecuencia, no existe una correspondencia entre la

\mathbf{G} y la \mathbf{G}^{-1} generada. Un análisis de esto se muestra en el Anexo B. En este caso se recurrirá a una alternativa, la descomposición modificada de Cholesky, la cual mejora el condicionamiento de la matriz métrica manteniendo la información de la matriz original.

La ecuación (2.10) requiere de invertir la matriz \mathbf{G} , y de realizar una multiplicación matricial por la matriz \mathbf{y} . Esto se puede reordenar como

$$\mathbf{G}\mathbf{x} = \mathbf{y} , \quad (2.17)$$

lo que es equivalente a resolver un sistema lineal de ecuaciones que puede hacerse de manera eficiente mediante alguna factorización de la matriz métrica. Dado que \mathbf{G} es simétrica, la factorización de Cholesky suele ser una opción rápida de calcular. Sin embargo, cuando la matriz métrica está mal condicionada, la factorización de Cholesky puede generar un error numérico grande, o su ejecución computacional puede terminar en error debido a la presencia de valores propios negativos. Como alternativa se emplea la factorización de Cholesky modificada¹³, ModChol, la cual consiste en obtener la factorización \mathbf{LDL}^T de la matriz métrica^a

$$\mathbf{G} = \mathbf{LDL}^T , \quad (2.18)$$

donde \mathbf{L} es una matriz triangular, \mathbf{D} es una matriz diagonal por bloques, y los bloques son de (1×1) y (2×2) . ModChol descompone los bloques D de la matriz \mathbf{D} en sus valores propios, y en caso de estar por debajo de una tolerancia les asigna una perturbación ΔD , de esta forma corrige el mal condicionamiento sin tener que obtener directamente los valores propios de la matriz. Al nuevo bloque, obtenido de la perturbación, se le simboliza como \hat{D} para diferenciarlo del bloque original. El proceso se representa según

$$\hat{D} = D + \Delta D . \quad (2.19)$$

La forma de realizar la perturbación se esquematiza en la Figura 2.2, en donde se muestra que si el bloque es de (1×1) , entonces solo se revisa que el único elemento esté por arriba de la tolerancia establecida; pero si el bloque es de 2×2 , entonces se descompone en sus valores y vectores propios para revisar que estén por arriba de la tolerancia. Si alguno de los valores propios fuera menor a la tolerancia, se reemplaza por la tolerancia y se reconstruye el bloque.

^a En la práctica se usó la factorización de Bunch-Kaufman³⁵, la cual es más eficiente. Esta tiene la forma $\mathbf{PGP}^T = \mathbf{LDL}^T$, donde \mathbf{P} es una matriz de permutación. Las ecuaciones se han escrito sin la matriz de permutación por claridad, y porque \mathbf{P} no es estrictamente necesaria para ModChol.

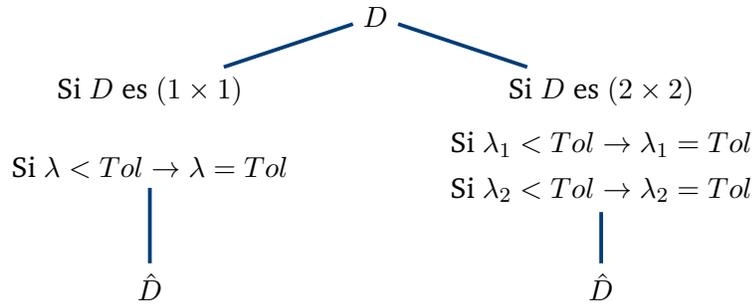


Figura 2.2. Esquema de perturbación de los bloques diagonales D en ModChol. Si se trata de un bloque de (1×1) , se tiene directamente de λ y si esta por debajo de la tolerancia se le asigna la tolerancia. Si se trata de un bloque de (2×2) , se descompone en sus vectores y valores propios, y si λ_1 o λ_2 están por debajo de la tolerancia, se les asigna la tolerancia, luego se reconstruye el bloque. El bloque original se denota por D , y el modificado por \hat{D} .

La matriz diagonal por bloques tras realizar la perturbación, \hat{D} , puede usarse para reconstruir la matriz métrica; sin embargo, como no es la matriz diagonal original, se reconstruye una matriz métrica perturbada, \tilde{G} . Esto es equivalente a agregar una pequeña perturbación E a la matriz G para volverla positivo definida y mejorar su condicionamiento; lo que se representa mediante la expresión

$$\tilde{G} = \mathbf{L}\hat{\mathbf{D}}\mathbf{L}^T = \mathbf{G} + \mathbf{E} \approx \mathbf{G} . \quad (2.20)$$

En la práctica, la perturbación es tan pequeña que \tilde{G} es esencialmente idéntica a la matriz métrica original, pero con mayor estabilidad numérica. De hecho se cumple que \tilde{G} conserva más información de la matriz métrica original que SVD y TED.

Al considerar ModChol, la ecuación (2.17) se transforma en

$$\mathbf{L}\hat{\mathbf{D}}\mathbf{L}^T \mathbf{x} = \mathbf{y} . \quad (2.21)$$

La ecuación (2.21) implica resolver varios sistemas de ecuaciones lineales de izquierda a derecha. Finalmente, se utiliza la ecuación (2.16) para generar las MO-ERIs y contribuir al término $\Sigma_{pp}^{(2)}$.

Algoritmo directo-w

” *For best performance a direct algorithm should be integral driven, i.e., integral evaluation concerns should dictate the order of events in the calculation. When an integral has been calculated, it should be used to the maximum extent possible, as long as no external storage is invoked.*

— **H.B. Schlegel**

Geometry Optimization on Potential Energy Surfaces
(1995).

Las ecuaciones de EP2 con base auxiliar pueden ser implementadas de varias maneras, tal que cada una aprovecha diferentes aspectos de los recursos computacionales disponibles. Este trabajo se orienta a analizar eficientemente sistemas moleculares de gran tamaño, y por tanto se asume que requerirán más memoria RAM de la disponible; el I/O de disco debe evitarse porque genera un cuello de botella y constituye una etapa de latencia que destruye la paralelización. Para superar las limitaciones de memoria, la generación de MO-ERIs se acopla a las iteraciones de EP2. A continuación se expone la implementación del esquema de la Figura 2.1, y cómo se relaciona cada paso con el manejo de los recursos computacionales. El algoritmo propuesto es del tipo directo ya que recalcula integrales, sin embargo, también aprovecha contracciones intermedias mediante el tensor w ; por lo tanto, para diferenciarlo de un algoritmo directo simple se le denominará algoritmo directo-w. Ya que en la sección de resultados se hará referencia a otros algoritmos, estos se muestran en el anexo C.

3.1 Pseudocódigo

La implementación propuesta se distingue por guardar el tensor w , que corresponde a contracciones intermedias de las AO-ERIs de tres centros, en lugar de las AO-ERIs mismas; esto permite que la ejecución solamente dependa de la memoria RAM y no del disco. El algoritmo optimiza la generación de las MO-ERIs necesarias para EP2, y se divide en cuatro etapas, la primera es la generación de la matriz métrica G , la

segunda considera la contracción sobre el *bra*, la tercera es la contracción sobre el *ket*, y finalmente, en la cuarta etapa se obtienen bloques de MO-ERIs y se realiza la contribución a los términos 2ph y 2hp de Σ_{pp} para obtener la energía de EP2. Los pasos a seguir junto con su escalamiento aritmético se muestran en el Algoritmo 1.

Algoritmo 1 Implementación del algoritmo directo-w para EP2

1. Reducir la dimensión de la base auxiliar de N_{aux} a N_{aux}^{LDF} mediante LDF.
 2. Calcular la matriz métrica \mathbf{G} $N_{aux}^{LDF^2}$
 3. Aplicar ModChol a la matriz métrica $\tilde{\mathbf{G}} = \mathbf{L}\mathbf{D}\mathbf{L}^T$ $N_{aux}^{LDF^3}$
 4. Calcular $z_{\nu P}^p = \sum_{\mu}^N C_{\mu p}(\mu\nu|P)$ $N_{aux}^{LDF} N^2$
 5. Calcular $y_{qP}^p = \sum_{\nu}^N C_{\nu q} z_{\nu P}^p$ $N_{aux}^{LDF} N^2$
 6. Resolver para \mathbf{x} el sistema $\mathbf{L}\mathbf{D}\mathbf{L}^T \mathbf{x}^{pT} = \mathbf{y}^{pT}$ $N_{aux}^{LDF^2} N$
 7. Determinar el número de lotes requerido de $w_{\lambda Q}^s$ dividiendo el índice s .
- for** cada lote **do**
- for** σ en N orbitales atómicos **do**
- for** λ en N orbitales atómicos **do**
- for** Q en N_{aux} funciones auxiliares **do**
8. Calcular AO-ERI ($Q|\sigma\lambda$).
 9. Contribuir $w_{\lambda Q}^s \leftarrow C_{\sigma s}(Q|\sigma\lambda)$ $N_{aux}^{LDF} N^3$
- end for**
- end for**
- end for**
- for** s en N Orbitales Moleculares **do**
- if** s es un Orbital Ocupado i **then**
- 10a. Tomar q en Orb. Virt. y $(pa|i\lambda) = \sum_Q (x^p)_{aQ} (w^{iT})_{Q\lambda} N_{aux}^{LDF} N_{occ} N_{vir} N$
 - 11a. Contraer λ a Orb. Virt. $(pa|ib) = \sum_{\lambda}^N C_{\lambda b} (pa|i\lambda)$ $N_{occ} N_{vir}^2 N$
 - 12a. Realizar la contribución a 2ph.
- end if**
- if** s es un Orbital Virtual a **then**
- 10b. Tomar q en Orb. Occ. y $(pi|a\lambda) = \sum_Q (x^p)_{iQ} (w^{aT})_{Q\lambda} N_{aux}^{LDF} N_{occ} N_{vir} N$
 - 11b. Contraer λ a Orb. Occ. $(pi|aj) = \sum_{\lambda}^N C_{\lambda j} (pi|a\lambda)$ $N_{occ}^2 N_{vir} N$
 - 12b. Realizar la contribución a 2hp.
- end if**
- end for**
- end for**
13. Calcular la energía de EP2
 14. Revisar convergencia
-

La primera etapa involucra tres pasos relacionados con la construcción de la matriz métrica. En el paso uno se provecha la localización orbital para remover del conjunto

de base auxiliar las funciones que no contribuyen al cálculo, de esta forma, se reduce la dimensión de la base auxiliar de N_{aux} a N_{aux}^{LDF} . La aproximación LDF tiene como consecuencia la generación de una matriz métrica más pequeña, la reducción del escalamiento de todos los pasos que involucren la base auxiliar y la disminución de la cantidad de memoria requerida. En el paso dos se evalúa la matriz métrica que contiene las AO-ERIs de dos centros. En el paso tres se aplica la descomposición de Cholesky Modificada a la matriz métrica, la cual se guarda en memoria para su posterior uso. Otras técnicas como SVD y TED podrían ser usadas en el paso tres, pero en este trabajo solo serán usadas para justificar la elección de ModChol.

En la segunda etapa se realiza la contracción del *bra*. Como pasos cuatro y cinco se calculan las matrices \mathbf{z} y \mathbf{y} dadas en las ecuaciones (2.8) y (2.9), estas matrices corresponden a la contracción del primer y segundo centro del bra respectivamente. Posteriormente, en el paso seis se obtiene la matriz \mathbf{x} al resolver el sistema de ecuaciones lineales de la ecuación (2.21). Alternativamente se puede utilizar la multiplicación por la inversa dada en la ecuación (2.10), sin embargo esto solo se realiza con fines comparativos puesto que se busca evitar la inversión de la matriz métrica.

La tercera etapa consiste en la contracción del *ket*. Es en esta etapa donde se expresa la parte de manejo de memoria del algoritmo, y el guardado de \mathbf{w} en lugar de las AO-ERIs. A diferencia de todos los demás arreglos temporales, \mathbf{w} tiene tres índices, por lo que el escalamiento aritmético y de memoria de su evaluación es el más importante, a causa de esto, se vuelve crucial darle un manejo eficiente. Las AO-ERIs y el tensor \mathbf{w} tienen las mismas dimensiones, $N^2 N_{aux}^{LDF}$; por tanto, cuando no hay suficiente memoria para guardar todas las AO-ERIs, tampoco la hay para guardar todo el tensor \mathbf{w} . Sin embargo, la clave está en que para generar una MO-ERI se necesitan todas las AO-ERIs, pero solo una porción de \mathbf{w} ; esta es la razón por la que el guardado de la contracción intermedia genera un manejo de memoria más eficiente y permite la división en lotes.

La generación de \mathbf{w} corresponde a contraer un índice de las AO-ERIs de tres centros a orbital molecular, y la evaluación en lotes corresponde en dividir en porciones los orbitales moleculares de la contracción. La partición se hace como sigue: si s representa a los orbitales moleculares, entonces a un orbital molecular determinado le corresponde la matriz $w_{\lambda Q}^s$ de dimensión $(N \times N_{aux}^{LDF})$; dependiendo de la identidad de s , esta matriz da origen al bloque de MO-ERIs p -ocupado- a -ocupado si s es un orbital virtual, o al bloque p -virtual- i -virtual si s es un orbital ocupado. Para aprovechar la memoria disponible, se generan y guardan simultáneamente tantas de estas matrices como quepan en memoria, el conjunto de estas matrices generan un lote; en el paso siete se calcula cuántos de estos lotes se necesitan para cubrir todos los orbitales

moleculares. En el paso ocho se calculan las AO-ERIs requeridas para contribuir a la contracción intermedia en w , la cual se realiza en el paso nueve. Los pasos ocho y nueve se realizan de manera conjunta, ya que se calcula una AO-ERI e inmediatamente se realiza su contribución a w en el paso nueve sin guardar el tensor completo de AO-ERIs en memoria, y se repiten los pasos ocho y nueve hasta haber calculado todas las AO-ERIs. La consecuencia de este esquema es que las AO-ERIs tienen que ser evaluadas tantas veces como el número de lotes, sin embargo el escalamiento de la evaluación de las AO-ERIs, $N^2 N_{aux}^{LDF} N_{lotes}$ es menor al de la evaluación del tensor w , $N^3 N_{aux}^{LDF}$, así que en tanto se cumpla que $N_{lotes} \ll N$, la reevaluación de integrales no es un problema significativo. En la cuarta etapa se genera la MO-ERI y se realiza la contribución a Σ_{pp} de la ecuación (2.2) para calcular EP2. Puesto que solo se requieren las MO-ERIs p -ocupado-virtual-ocupado y p -virtual-ocupado-virtual, el procedimiento a seguir depende del orbital molecular que represente s , es decir si la contribución se hará a 2ph o 2hp. Si s representa un orbital ocupado, entonces se toma la subporción de la matriz x que representa orbitales virtuales y se genera la ERI intermedia $(pa|i\lambda)$ en el paso diez. En el paso once se contrae el último índice del *ket* a orbitales virtuales y en el paso doce se realiza la contribución al término 2ph. El procedimiento equivalente se realiza si s representa un orbital virtual, se toma la subporción de x que representa orbitales ocupados y se genera la ERI intermedia $(pi|a\lambda)$ en el paso diez, en el paso once se contrae el último índice del *ket* a orbitales ocupados y en el paso doce se realiza la contribución al término 2hp. Posteriormente, se calcula la energía de unión del electrón, ω , en el paso trece y finalmente se revisa la convergencia en el paso catorce para decidir si reiniciar el ciclo o terminar.

Es importante destacar que el esquema descrito no genera una MO-ERI a la vez, sino un bloque de MO-ERIs anti-simetrizadas. Además, el esquema de evaluación por lotes solo divide en etapas la contracción, la cual es el paso de mayor coste de operaciones aritméticas, a cambio de repetir la evaluación de las AO-ERIs, que no son cuello de botella. El aumentar la memoria disponible en el sistema de cómputo contribuye a disminuir el número de lotes y por tanto las reevaluaciones de AO-ERIs, sin embargo, como no son cuello de botella, el efecto se verá amortiguado a menos que el número de lotes sea demasiado grande. Para mejorar el desempeño del algoritmo resulta más importante aumentar el número de procesadores para realizar la contracción a MO-ERIs de manera más eficiente.

3.2 Esquema de paralelización

La implementación en paralelo del algoritmo fue realizada utilizando OpenMP, un modelo de memoria compartida. Se utilizó un esquema de paralelización para la generación de las AO-ERIs y su contribución al tensor w , y otro para la contracción final a MO-ERIs y su contribución al término Σ_{pp} . El primer esquema de paralelización, relacionado con la construcción del tensor w , abarca los pasos ocho y nueve del Algoritmo 1, y se aplica directamente sobre el barrido de la base auxiliar. Los átomos se reparten entre los procesadores, tal que a cada uno le toca generar las AO-ERIs que contengan las funciones auxiliares de los átomos que le corresponden, y realizar su contribución a w . El segundo esquema de evaluación, relacionado con la generación final y uso de las MO-ERIs, comprende los pasos del diez al doce del Algoritmo 1, en este caso, el reparto de trabajo en los procesadores se realiza directamente sobre las funciones de base. Los otros algoritmos, con los que se compara el propuesto en la sección de resultados, siguieron esquemas de paralelización similares.

El esquema de paralelización se eligió de tal forma que se maximice la distribución de operaciones entre los procesadores, y cumple con el propósito de poder analizar sistemas grandes en un tiempo corto con un uso aceptable de los recursos de cómputo. Además, se han aplicado los mismos principios para paralelizar todos los algoritmos comparados, de tal forma que no haya gran influencia en las comparaciones al momento de hacer cálculos en paralelo. No obstante, es importante mencionar que existen otros esquemas de paralelización aplicables.

Resultados

En este capítulo se prueba la validez y la influencia de las aproximaciones en el algoritmo directo-w, así como su mejora respecto a otras alternativas. Primero, se analiza el efecto de las aproximaciones ModChol y LDF, posteriormente, se explora el comportamiento del algoritmo ante un número de lotes variable. Finalmente, se exponen los beneficios de la generación de MO-ERIs a través del manejo del tensor w contra otros algoritmos cuyo fundamento es el manejo de las AO-ERIs, ya sea guardándolas o recalculándolas. Detalles sobre la implementación de estos algoritmos se muestran en el anexo C.

Previo al cálculo de EP2, la geometría de las moléculas analizadas fue optimizada en el *software* Psi4³⁶ utilizando B3LYP^{37,38} y la misma base que se indica en el correspondiente cálculo del propagador. Posteriormente, se realizó un cálculo de Hartree-Fock (HF) en el *software* Nagual³⁹ para obtener las energías orbitales y los coeficientes de transformación de orbital atómico a orbital molecular, en todos los casos el cálculo de Hartree-Fock se realizó con ajuste de la densidad ya que ha sido reportado que esto tiene un efecto aceptable en el resultado de EP2.³¹ El tiempo de HF no se considera en los resultados, sino únicamente el tiempo de cálculo de EP2. Excepto cuando se indique lo contrario, todos los cálculos de EP2 fueron realizados a 12 *cores* de un procesador Intel Xeon CPU E5-2680 a 2.70GHz. En todos los casos se utilizó la base auxiliar GEN-A2*.⁴⁰⁻⁴²

4.1 Estabilidad numérica de la matriz métrica

El efecto de ModChol en la estabilidad numérica de la matriz métrica se analiza primero, puesto que será usado en el resto de los cálculos. La introducción de la base auxiliar mediante el ajuste de la densidad lleva a tener que invertir la matriz métrica, G , la cual es positivo definida. Debido a la precisión finita de la computadora, su representación computacional puede estar mal condicionada; en este caso la inversión genera problemas de inestabilidad numérica y el valor final de energía no es confiable.

Para elucidar el efecto de ModChol al evitar la inversión de la matriz métrica, y justificar su elección frente a otros posibles alternativas como SVD o TED, se analizó la matriz métrica de los fulerenos C_{60} , C_{70} y C_{82} . En la Tabla 4.1 se muestra que en todos los casos las matrices métrica analizadas presentan valores propios negativos, lo cual demuestra el error numérico inducido por la limitada precisión de la computadora. También se muestra el número de condición^a de la matriz métrica, \mathbf{G} , y de la matriz generada tras aplicar ModChol ($\tilde{\mathbf{G}}$); la aplicación de ModChol reduce el número de condición en varios órdenes de magnitud respecto a la matriz métrica original, esto es prueba de que ModChol mejora la estabilidad numérica de la matriz.^b

Tabla 4.1. Análisis de la matriz métrica de fulerenos seleccionados, con base 6-311+G(d)^{43,44} y base auxiliar GEN-A2*. Se muestra el número de condición de la matriz métrica original, \mathbf{G} , y de la matriz métrica tras aplicar ModChol, $\tilde{\mathbf{G}}$, junto con los valores propios (λ) negativos de la matriz métrica original. En todos los casos ModChol mejora el condicionamiento de la matriz métrica.

Molécula	N_{aux}	$Cond(\mathbf{G})$	$Cond(\tilde{\mathbf{G}})$	$\lambda < 0$
C_{60}	5640	9.06×10^{19}	8.61×10^{17}	210
C_{70}	6580	1.07×10^{21}	6.59×10^{17}	248
C_{82}	7708	4.31×10^{20}	1.03×10^{18}	295

La SVD y TED son técnicas que aseguran el obtener una inversa, pero son inferiores en estabilidad numérica a ModChol. La SVD y TED sí mejoran la estabilidad numérica respecto a la matriz métrica original ya que niegan el uso de valores singulares o de valores propios muy pequeños, por debajo de una tolerancia, los cuales crean problemas de precisión numérica al momento de invertirlos. El descarte de valores por debajo de la tolerancia implica poner un cero en lugar del inverso, sin embargo, la matriz \mathbf{G}^{-1} generada es singular puesto que tiene ceros en sus valores propios o en sus valores singulares. Por tanto, la inversa generada no tiene conexión con la matriz métrica original.^c Por otra parte, el número de valores propios o singulares descartados por estar debajo de la tolerancia establecida puede ser bastante grande, como se muestra en la Tabla 4.2, llegando hasta el 15% en el caso del fullereno de 82 carbonos, esto se traduce en pérdida de la información del sistema. En cambio, ya que ModChol aplica una muy pequeña perturbación a \mathbf{G} , la nueva matriz $\tilde{\mathbf{G}}$ es prácticamente la

^a El número de condición se define por $cond(\mathbf{G}) = \|\mathbf{G}\| \|\mathbf{G}^{-1}\|$, es decir, la norma de una matriz multiplicada por la norma de su inversa. Este es un número mayor o igual a uno que permite evaluar el condicionamiento de una matriz. Una matriz bien condicionada tiene un número de condición de uno, mientras que una matriz está peor condicionada entre más grande sea su número de condición.

^b En el anexo se muestra un análisis sobre el por qué reducir el número de condición mejora la estabilidad numérica de la metodología.

^c No es posible invertir la matriz inversa para recuperar la matriz original si se truncan valores propios o singulares, es decir que $(\mathbf{G}^{-1})^{-1} \neq \mathbf{G}$.

matriz G original y se evita tener que manejar una matriz singular. Cabe señalar que aunque la matriz resultante no sea la matriz métrica original, la diferencia es mucho menor al error numérico introducido al invertir una matriz mal condicionada, y menor que el que inducen otras alternativas como SVD y TED. Un análisis exhaustivo de las diferencias entre las matrices métrica original y las obtenidas tras aplicar SVD, TED y ModChol a los fulerenos se muestra en el anexo B.

Otra ventaja de ModChol es el tiempo de cómputo que requiere. En la Tabla 4.2 se muestra el tiempo para la operación matemática de G , sea esta la inversión en el caso de SVD y TED, o la descomposición en el caso de ModChol. En el caso de los fulerenos, el tiempo requerido para realizar ModChol es sustancialmente menor a las alternativas, siendo entre 25 y 40 veces más rápido que SVD, y entre 15 y 20 veces más rápido que TED, al mismo tiempo que corrige de manera correcta el espectro de valores propios. Por tanto, el método de ModChol es más rápido, estable y seguro frente a sus contrapartes.

Tabla 4.2. Análisis de aplicar truncamiento con SVD y TED para invertir la matriz métrica, y de aplicar ModChol para factorizarla, usando base 6-311+G(d) y base auxiliar GEN-A2*. Los valores singulares (σ) y los valores propios (λ) menores a la tolerancia representan una cantidad importante, en este contexto la metodología de ModChol logra hacer una buena corrección de los valores propios en un lapso de tiempo sustancialmente menor que SVD y TED. La tolerancia utilizada fue de 10^{-10} y el tiempo se da en minutos.

Molécula	N_{aux}	t_{SVD}	$\sigma < Tol$	t_{TED}	$\lambda < Tol$	$t_{ModChol}$	$\lambda < Tol$
C ₆₀	5640	18.36	768	8.88	768	0.6	748
C ₇₀	6580	28.3	946	13.5	946	0.9	907
C ₈₂	7708	46.7	1158	22.15	1158	1.4	1122

4.2 Localización orbital

En esta sección se estudia la localización de varios orbitales, y su efecto sobre el proceso de división por lotes del tensor w causado por la aplicación de la aproximación LDF¹⁶. Como moléculas de prueba se utilizan el artesunato, la coelenteracina, el mirabegron, y la narciclasina, las cuales se muestran en la Figura (4.1); estas moléculas fueron elegidas por presentar diversos tipos de átomos, y fragmentos estructurales variados como anillos, cadenas lineales sustituidas y aromaticidad.

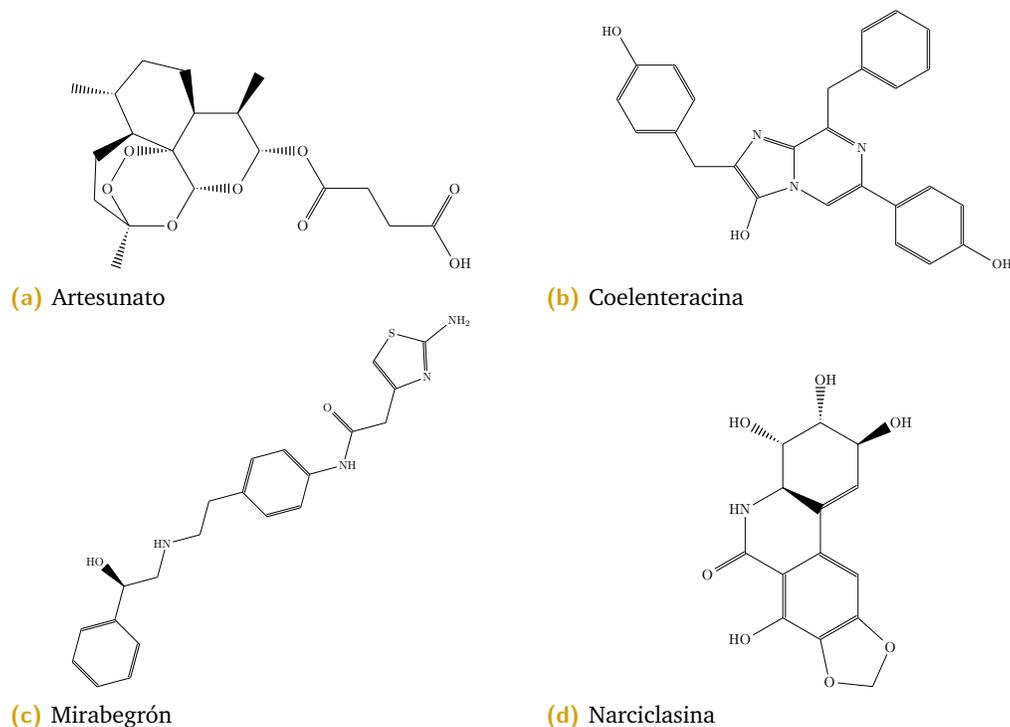


Figura 4.1. Moléculas de prueba para estudiar la localización en varios orbitales y su efecto en el algoritmo directo- w , las moléculas son las siguientes: a) artesunato, un fármaco para la malaria, b) coelenteracina, una molécula bioluminiscente encontrada en organismos marinos, c) mirabegrón, un fármaco para tratar la vejiga hiperactiva, y d) narciclasina, molécula con propiedades antitumorales y antiinflamatorias.

En particular, se analiza la localización de los orbitales ocupados, estos fueron utilizados en su forma canónica sin considerar ninguna transformación adicional que induzca localización. Para evaluar la localización orbital, se agrupan las funciones de base por átomos, y se reordenan los conjuntos en orden decreciente acorde a su contribución a la ocupación orbital. Posteriormente, se descartan las funciones de base auxiliar pertenecientes a los átomos que contribuyen en menos del 0.01% a la ocupación,

esto permite obtener un conjunto de funciones de base auxiliar reducido (N_{aux}^{LDF}) respecto al conjunto de funciones de base auxiliar completo (N_{aux}). El número de funciones auxiliares en la aproximación LDF para cada molécula se muestra en la Figura (4.2), donde se observa que en general los orbitales de *core* presentan una mayor localización que los orbitales de valencia; esto es representado en las gráficas por la región de puntos de la parte inferior izquierda que pertenece al *core*, la cual posee menos funciones auxiliares que la sección derecha que pertenece a valencia. Sin embargo, también es posible que se presente localización significativa en los orbitales de valencia, como se ve con claridad en el ejemplo de mirabegron de la Figura (4.2c), también se presenta una localización en menor grado en algunos orbitales de valencia de narciclasina y coelenterasina en las Figuras (4.1b) y (4.1d) respectivamente.

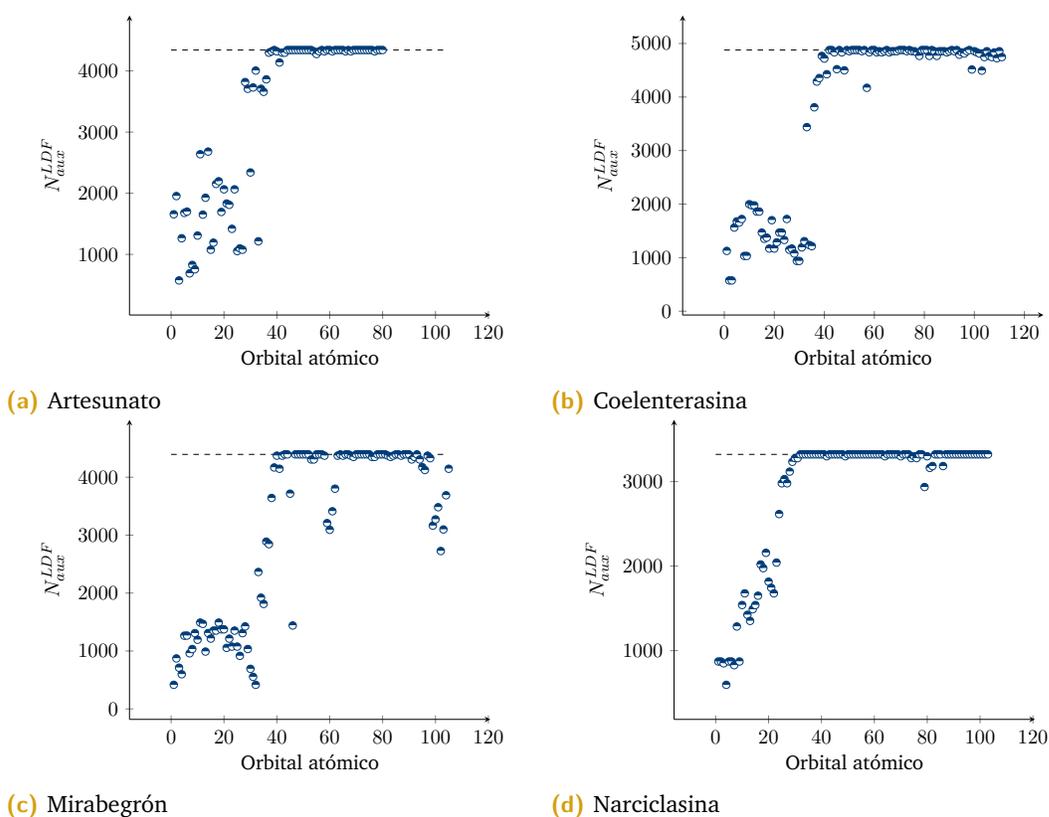


Figura 4.2. Número de funciones de base auxiliar tras considerar la localización en varios orbitales de a) artesunato, b) coelenterasina, c) mirabegron, d) narciclasina; en todos los casos es más importante la localización de los orbitales de *core*, sin embargo, en algunos orbitales de valencia la localización también puede ser significativa. La línea punteada corresponde al conjunto de funciones auxiliares completo, N_{aux} . Se utilizó la base aug-cc-pVDZ^{45,46} y la base auxiliar Gen-A2*

La aplicación de LDF mediante el criterio descrito ha sido previamente reportada⁴⁷ y aplicada a EP2³¹, con efectos como reducción de las dimensiones de la matriz métrica, de la memoria involucrada, y del escalamiento de las operaciones. Sin embargo, ya que la implementación propuesta no depende de disco, y usa ModChol, se tienen algunas consecuencias adicionales. Aunque la matriz métrica reduce su dimensión, el efecto sobre las operaciones de la matriz métrica se vuelve menos importante ya que ModChol hace que las operaciones involucradas sean sustancialmente más rápidas en comparación a los otros pasos del algoritmo. Por otra parte, se necesita menos memoria para el tensor w , lo cual se traduce en un menor número de lotes, y en consecuencia en un menor número de revaluaciones de las AO-ERIs.

Para probar el efecto de la aproximación LDF sobre el algoritmo directo- w , se evaluaron varios orbitales de mirabegrón con la base aug-cc-pVTZ^{45,46}. Esta base fue elegida para crear una alta demanda de memoria y en consecuencia forzar la necesidad de varios lotes de w . El análisis de localización de orbitales se muestra en la Figura (4.3), el cual es similar al de la Figura (4.2c), pero con una distribución de funciones auxiliares distinta ya que provienen de otra base. Se cumple la tendencia ya descrita, es decir, la localización de los orbitales de *core* en el extremo izquierdo es mayor que la de los orbitales de valencia en el extremo derecho, pero aún hay localización aprovechable en los últimos; los puntos dorados corresponden a orbitales seleccionados usados para analizar el efecto de la localización en el algoritmo directo- w .

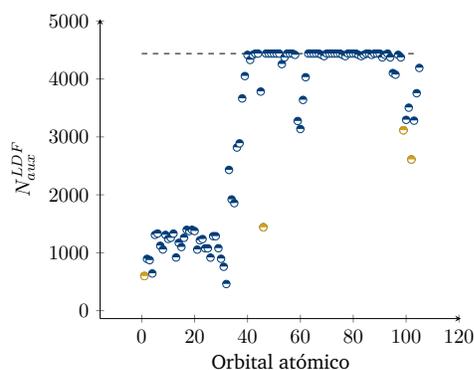


Figura 4.3. Número de funciones de base auxiliar tras considerar la localización a varios orbitales de mirabegrón con base aug-cc-pVTZ y base auxiliar Gen-A2*. Los puntos dorados indican orbitales representativos seleccionados para analizar su comportamiento en el algoritmo propuesto.

Los puntos dorados de la Figura (4.3) se calcularon con LDF y sin LDF para comparar el tiempo de cómputo y los valores energéticos mostrados en la Tabla 4.3, también se muestra el número de lotes requeridos para cada punto considerando que se utilizó

una memoria de 36 GB. En la Tabla se evidencia que la localización orbital reduce el número de lotes necesarios. La reducción en la base auxiliar también disminuye la dimensión de las AO-ERIs, por tanto, la localización orbital se traduce en evaluar menos AO-ERIs en una menor cantidad de revaluaciones, este efecto es más pronunciado en los orbitales de *core* donde la localización es más importante, y en el mejor de los casos se ha ahorrado hasta el 32% del tiempo de cómputo. Por otra parte, el error introducido por la aproximación LDF sobre la energía es pequeño, del orden de 10^{-6} para el mirabegrón estudiado, lo cual es varios órdenes de magnitud menor que el error generado por introducir la base auxiliar; en consecuencia, el uso de la aproximación LDF en EP2 es aceptable, lo que concuerda con lo ya reportado en el caso del algoritmo convencional³¹.

Tabla 4.3. Efecto de la aproximación LDF sobre orbitales seleccionados de mirabegrón usando el algoritmo directo-w. La localización orbital es más importante en los orbitales de *core*, y lleva una reducción en el tiempo de cómputo y en el número de lotes requeridos sin afectar significativamente la energía. El tiempo de cómputo se da en horas, y la energía en eV. Para referencia, el número de funciones de base auxiliar original es $N_{aux} = 4438$ con un tiempo promedio de cómputo de $t_{EP2} = 6.09h$. Se utilizó la base aug-cc-pVTZ y la base auxiliar Gen-A2*.

Orbital Molecular	N_{aux}^{LDF}	t_{EP2}	N_{lotes}	E_{EP2}	$ \Delta E_{EP2} $
1	598	4.11	1	-2478.636	7.7×10^{-6}
46	1440	5.04	2	-27.374	6.3×10^{-6}
99	3114	5.87	4	-9.555	1.5×10^{-5}
102	2610	5.99	3	-8.783	2.0×10^{-6}

4.3 Desempeño con respecto al número de lotes

La cantidad de lotes depende de la memoria disponible, por lo que es importante conocer cómo las variaciones en el número de lotes afectan al tiempo de cómputo. Para esto, se estudió la molécula de fenantreno a 2, 4, 6, 8, 16 y 32 lotes con el algoritmo directo-w, los tiempos de cómputo se muestran en la figura 4.4. El tiempo de cómputo aumenta con el número de lotes, esto se debe a que hay que reevaluar las AO-ERIs en cada lote. No obstante, tener que evaluar 32 lotes, no significa que el tiempo de cómputo sea 32 veces más largo, esto es positivo puesto que significa que el aumento en el número de lotes está amortiguado, y la razón es que la evaluación de las AO-ERIs tiene un menor escalamiento aritmético que las contracciones de AO-ERIs a MO-ERIs. Adicionalmente, la división en lotes solo parte la contracción y por tanto no aumenta el cuello de botella.

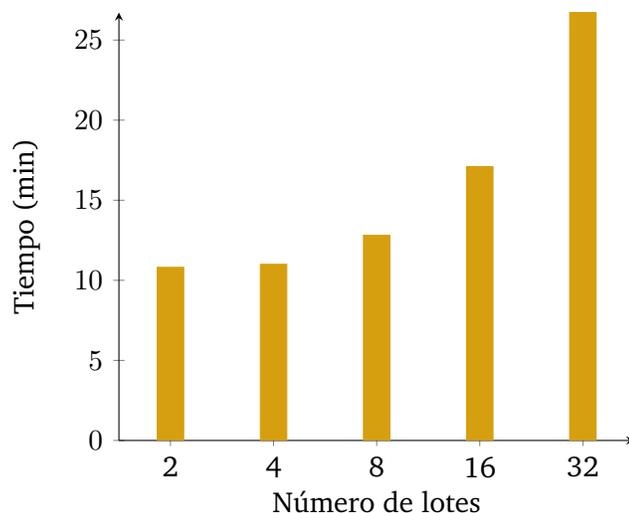


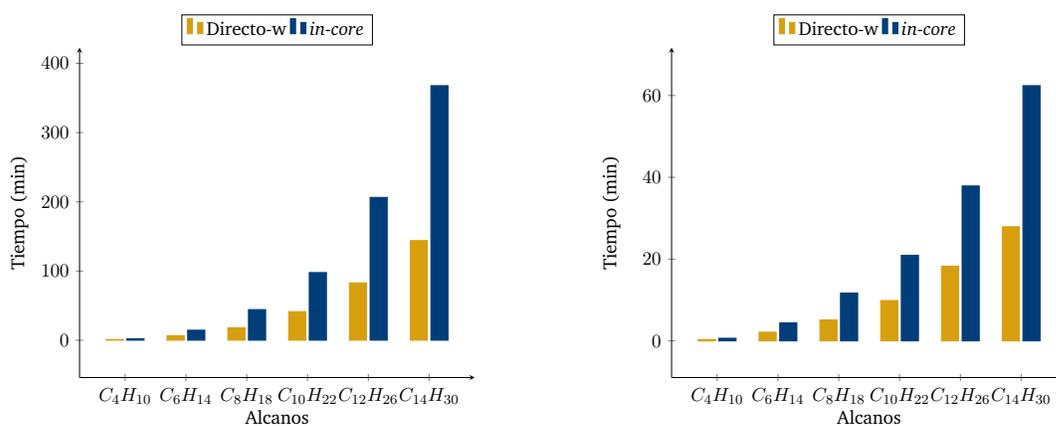
Figura 4.4. Cálculo de la molécula de fenantreno variando el número de lotes a 2, 4, 8, 16 y 32, con el algoritmo directo-w. Se utilizó base cc-pVTZ y base auxiliar GEN-A2*.

4.4 Análisis de ventajas del algoritmo

El algoritmo directo-w representa una forma de superar las limitaciones de memoria con base en regenerar las MO-ERIs por partes en cada iteración del NR de EP2. Para exponer las razones y ventajas del diseño del algoritmo, es necesario compararlo con otros posibles algoritmos en el contexto de integrales con base auxiliar aplicadas a EP2. Las comparaciones se realizan utilizando alcanos lineales, ya que son moléculas muy parecidas entre sí que solo aumentan en tamaño.

La primera comparación se realiza con un algoritmo *in-core*, en el que se guardan todas las AO-ERIs en RAM, y se utilizan para generar la contracción a MO-ERIs en cada iteración. El algoritmo *in-core* no permite superar las limitaciones de memoria RAM, pero representa un punto de comparación entre el esquema de guardar las AO-ERIs contra el esquema de usar el tensor w . Para comparar ambos algoritmos, se permitió suficiente memoria RAM tal que pudiesen ser guardadas todas las AO-ERIs, o que se use un solo lote del tensor w , esto evita la influencia de la división por lotes. Dos observaciones deben hacerse antes de analizar los resultados, primero, el algoritmo propuesto no es necesario si se puede usar un solo lote, pues esto indica que no hay limitaciones de RAM, y segundo, si hay suficiente RAM para guardar las AO-ERIs en un algoritmo *in-core*, entonces resulta mejor guardar las MO-ERIs desde el inicio. No obstante, esta comparación pretende demostrar que guardar w permite un manejo de memoria correcto y que contribuye de forma positiva al algoritmo.

La Figura (4.5) muestra el tiempo de cómputo del algoritmo *in-core* comparado con el del algoritmo *directo-w*, donde se observa que usar el tensor *w* es un poco más de dos veces más rápido que guardar las AO-ERIs. El experimento se hizo tanto con un algoritmo serial, que se muestra en la Figura (4.5a), como con un algoritmo paralelo que se muestra en la Figura (4.5b); la similitud entre ambas Figuras muestra que la mejora es efectiva sin gran influencia del proceso de paralelización. La mejor gestión de la memoria se debe a que mientras el algoritmo *in-core* tiene que acceder a la memoria y barrer todas las AO-ERIs para cada uno de los bloques de MO-ERIs a generar, el algoritmo propuesto accede a una AO-ERI y la utiliza inmediatamente tantas veces como sea necesario para todas las contribuciones a *w*. En cuanto a *hardware*, esto significa que el algoritmo *in-core* guarda las AO-ERIs en RAM, y cada vez que se va a generar una MO-ERI, tiene que cargar cada AO-ERI en caché, utilizarla, y transferir la próxima AO-ERI de RAM a caché, la transferencia de RAM a caché se hace tantas veces como la cantidad de AO-ERIs, y el proceso debe repetirse tantas veces como los orbitales moleculares. Por otra parte, el esquema de generación del tensor *w* genera una AO-ERI, la cual se carga en caché, y se usa varias veces para todas las contribuciones a *w*, en este esquema todas las veces que se usa una AO-ERI son consecutivas y por tanto solo requieren de una sola transferencia de RAM a caché por AO-ERI⁴⁸. Una última razón para que el algoritmo propuesto sea más rápido que guardar las integrales es que la caché es más rápida que la memoria RAM. Por lo tanto, el uso del tensor *w* lleva a una reducción en la cantidad de accesos a memoria y a gestionarla de manera más eficiente.



(a) Comparación Serial

(b) Comparación Paralela

Figura 4.5. Comparación entre el algoritmo *in-core*, que guarda las AO-ERIs en RAM, y el algoritmo *directo-w* para el cálculo de EP2 de alcanos lineales utilizando la base aug-cc-pVDZ y la base auxiliar Gen-A2*.

El algoritmo *in-core* es ejecutable siempre que haya suficiente RAM para guardar las integrales. La limitación de RAM se puede solucionar al guardar las AO-ERIs en disco y leerlas cada vez que se vaya a generar un bloque de MO-ERIs, este es el algoritmo convencional. Sin embargo, este algoritmo tiene que hacer una gran cantidad de escrituras y lecturas de disco, que representan el cuello de botella que el algoritmo directo-w busca evitar. Para mostrar la importancia de evitar el cuello de botella de disco, se analizó el tetradecano con el algoritmo convencional y con el algoritmo directo-w, los resultados se muestran en la Tabla 4.4, donde se demuestra que el algoritmo propuesto es 48 veces más rápido que el convencional. En este caso se ha intercambiado el I/O de disco por una mayor cantidad de operaciones, las cuáles son pagadas de manera eficiente al utilizar varios *cores* de CPU. La implementación de un algoritmo directo no depende del tensor w , de hecho basta recalcular cada AO-ERI conforme se necesite, este es un algoritmo directo simple que no minimiza el recálculo de las integrales ni toma provecho de la RAM; la Tabla 4.4 muestra que hasta este algoritmo es más rápido que el algoritmo convencional por evitar la lectura de disco, no obstante el algoritmo directo-w es 22 veces más rápido que el algoritmo directo simple ya que aprovecha la RAM para minimizar las revaluaciones de AO-ERIs.

Tabla 4.4. Comparación entre el tiempo de cálculo de EP2 utilizando el algoritmo directo-w, el algoritmo convencional, y el algoritmo directo simple en tetradecano con base aug-cc-pVDZ y base auxiliar GEN-A2*. En todos los casos el algoritmo directo-w es más rápido que sus contrapartes.

$C_{14}H_{30}$	Directo-w	Convencional	Directo Simple
Tiempo (min)	28	1345	610
$t/t_{Directo-w}$	1	48	22

Conclusiones y Perspectivas

Se ha generado una implementación robusta para EP2 que permite estudiar sistemas moleculares grandes más allá de las limitaciones de memoria, el cuello de botella de disco es evitado y se hace un balance entre uso de memoria y revaluación de las integrales. El uso de la base auxiliar ha sido clave en el diseño del algoritmo por los arreglos que genera, los cuales simplifican el manejo de memoria. El hecho de calcular contracciones intermedias por lotes en lugar de simplemente guardar las integrales agrega complejidad a la implementación, pero representa una optimización importante que mejora la ejecución computacional de la teoría. La técnica de ModChol también ha sido importante para la implementación, pues reduce el tiempo de cómputo al mismo tiempo que aporta estabilidad numérica, por otra parte, la localización orbital puede ser útil para el algoritmo en algunos casos, sobre todo al analizar orbitales de *core*. Se espera que este algoritmo favorezca que la teoría sea aplicada a sistemas reales y que tenga un impacto en la resolución de problemas de interés.

Los principios aplicados en el algoritmo directo-w de EP2 son generalizables a órdenes superiores de aproximación del propagador del electrón; en particular existe interés en explorar su adaptación a las ecuaciones de P3. Aunque este trabajo se ha centrado en usar la base auxiliar para generar un algoritmo directo que supere las limitaciones de RAM, también se ha explorado y el uso de la base auxiliar para disminuir el escalamiento aritmético y de memoria de la generación de MO-ERIs en algoritmos *in-core* en EP2; así que también se desea explorar este beneficio en el caso de P3. Otra área de oportunidad a explorar es la implementación del algoritmo directo-w en arquitecturas de cómputo heterogéneo, es decir, aprovechar el uso conjunto de CPU/GPU para reducir aún más el tiempo de cómputo. La cantidad de memoria disponible ha crecido en las GPUs más recientes, pero sigue representando una cantidad limitada, por tanto el algoritmo directo-w es potencialmente aplicable para aprovechar el alto paralelismo que presentan cuando no hay suficiente memoria.

Reflexión Personal

Esta última parte del texto corresponde a un pensamiento final y a una reflexión sobre la tendencia a futuro en el diseño de algoritmos para la química computacional. La discusión sobre los algoritmos usados en química cuántica está íntimamente motivada por el deseo de poder analizar sistemas moleculares más grandes en un menor tiempo, y proviene desde los años 70's. Desde entonces se ha modificado el balance de cual es el algoritmo más prometedor conforme cambian las capacidades computacionales de la época. Cuando la capacidad de cómputo era limitada y la mayor memoria residía en el disco, la opción a elegir eran los algoritmos convencionales que minimizaban el cómputo; cuando la memoria RAM aumentó, los algoritmos *in-core* se volvieron una opción viable, y cuando las operaciones de punto flotante por segundo, flops, que puede realizar una computadora aumentó, el estudio de nuevos algoritmos volteó a ver a los algoritmos semidirectos y directos.

Desde el surgimiento de artículos que analizaban este tipo de aspectos, muchas cosas han cambiado en el ámbito de la química computacional; mientras que al final de los años 70's la supercomputadora CRAY-1 alcanzaba velocidades de reloj de 80 MHz con 8MB de memoria y un procesamiento del orden de los cientos de megaflops, y en los 80 tener 2 Gb de disco se consideraba una enormidad, en la actualidad una *laptop* de uso común puede llegar con facilidad a las decenas y cientos de gigaflops, con memoria de disco del orden de terabytes e incluso la computadora más barata tiene CPUs con mínimo 1GHz. La predicción hecha en los 80's se cumplió y sigue siendo válida en el 2020, *la capacidad de cómputo ha crecido más rápido que la I/O de disco*, esta idea es la que ha motivado a hacer un algoritmo directo en este trabajo. Sin embargo, no está garantizado que esto se quede así; la aparición de los discos de estado sólido con una alta eficiencia de I/O puede representar un renacimiento de los métodos convencionales, o al menos ampliar su aplicabilidad. No obstante, esto entra en competencia con la popularización de los GPUs en cálculos de química cuántica, que traen una paralelización masiva y en la que los métodos directos pueden resultar nuevamente prometedores. De hecho, la supercomputadora de mayor poder de procesamiento en la actualidad basa su poder de cómputo en un uso masivo de GPUs. Aunque el desarrollo de los GPUs los ha vuelto más eficientes tanto en memoria como en procesamiento, la memoria que tienen disponible es poca en comparación con

la gran capacidad que puede tener la RAM, aquí los algoritmos directos representan un área amplia de oportunidad para la química cuántica para llevar a cabo cálculos más complejos pero también con mayor precisión química.

Probablemente un nuevo cambio en la balanza de los algoritmos vaya a surgir en los próximos años con el desarrollo, la regularización y la reducción de costos tanto de las GPUs como de los discos duros de estado sólido, en tanto que el desarrollo de las CPUs momentáneamente parece estancado. Podría ser que el cómputo en CPU pueda dar un nuevo salto con la prometedora computación cuántica, aunque también esta podría llevar al surgimiento de una arquitectura no considerada hasta ahora y que deba necesitar un rediseño de los algoritmos. Personalmente considero que el aumento en las *flops* seguirá siendo más importante y por tanto los algoritmos directos serán la opción por defecto para poder estudiar sistemas moleculares cada vez más grandes con teorías cada vez más exactas, lo que llevará a la expansión de la química teórica a campos que son aún prohibitivos en la actualidad.

Newton-Raphson en EP2

La ejecución de EP2 se lleva a cabo mediante un proceso iterativo guiado por Newton-Raphson, para ello se parte de la ecuación para calcular las energías de unión de electrón (EBE's), dada por

$$\omega = \varepsilon_p + \Sigma_{pp}(\omega) . \quad (\text{A.1})$$

Se define la función auxiliar

$$f(\omega) = \omega - \varepsilon_p - \Sigma_{pp}(\omega) , \quad (\text{A.2})$$

la cuál tiene como raíz los valores de EBE's. Si ω_i es un valor de EBE en la iteración i , entonces el valor de ω_{i+1} se calcula con Newton-Raphson mediante

$$\omega_{i+1} = \omega_i - \frac{f(\omega_i)}{f'(\omega_i)} . \quad (\text{A.3})$$

La derivada f' se calcula según

$$f'(\omega_i) = \frac{df(\omega_i)}{d\omega} = 1 - \frac{d\Sigma_{pp}(\omega_i)}{d\omega} , \quad (\text{A.4})$$

al tomar la expresión para el término diagonal de la matriz de auto-energía, se obtiene

$$\frac{d\Sigma_{pp}^{(2)}(\omega_i)}{d\omega} = - \sum_{i,a < b} \frac{|(pa|ib) - (pb|ia)|^2}{(\omega + \varepsilon_i - \varepsilon_a - \varepsilon_b)^2} - \sum_{a,i < j} \frac{|(pi|aj) - (pj|ai)|^2}{(\omega + \varepsilon_a - \varepsilon_i - \varepsilon_j)^2} . \quad (\text{A.5})$$

En la práctica esta expresión se evalúa con facilidad al calcular el término de la matriz de auto-energía, ya que una vez que se tiene una contribución al término 2ph o al término 2hp, solo hay que dividir otra vez por el denominador y volver el resultado negativo. Al sustituir la ecuación (A.5) y la ecuación (A.2) en la ecuación (A.3) se obtiene el valor de la nueva predicción de EBE dada por

$$\omega_{i+1} = \left(1 - \frac{d\Sigma_{pp}(\omega_i)}{d\omega} \right)^{-1} \left(\varepsilon_p + \Sigma_{pp}(\omega_i) - \omega_i \frac{d\Sigma_{pp}(\omega_i)}{d\omega} \right) . \quad (\text{A.6})$$

La forma de la ecuación (A.6) es particularmente importante ya que contiene el término

$$P_s = \left(1 - \frac{d\Sigma_{pp}(\omega_i)}{d\omega}\right)^{-1}, \quad (\text{A.7})$$

el cuál se denomina fuerza del polo o *pole strength* en inglés, y es un número entre cero y uno que provee información sobre la validez de la aproximación usada para obtener las EBEs.

Análisis de estabilidad numérica de la matriz métrica

La aproximación de ajuste de la densidad implica la inversión de la matriz métrica, \mathbf{G} , la cual es positivo definida y por tanto su inversión no debería ser problemática. Sin embargo, la representación computacional de la matriz puede no ser positivo definida y estar cerca de ser singular debido a la precisión numérica finita de la computadora, esto puede generar que la inversión no esté garantizada, o que haya problemas de estabilidad numérica. Para superar esto, usualmente la inversión se realiza mediante técnicas como la descomposición en valores singulares (SVD) o la descomposición en valores propios (TED). Estas técnicas descomponen la matriz métrica en sus valores singulares, $\mathbf{G} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, o en sus valores propios, $\mathbf{G} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T$, respectivamente, donde las matrices \mathbf{U} y \mathbf{V} son matrices unitarias, $\mathbf{\Sigma}$ es una matriz diagonal con los valores singulares, σ , \mathbf{P} es la matriz de vectores propios, y $\mathbf{\Lambda}$ es una matriz diagonal con los valores propios, λ . La inversa se encuentra como $\mathbf{G}^{-1} = \mathbf{V}\mathbf{\Sigma}^{-1}\mathbf{U}^T$ o $\mathbf{G}^{-1} = \mathbf{P}\mathbf{\Lambda}^{-1}\mathbf{P}^T$, por tanto solo hay que evaluar λ^{-1} o σ^{-1} . Sin embargo, las σ o las λ muy pequeñas pueden generar inestabilidad numérica durante su inversión, así que en la práctica, el inverso de las σ o λ que estén por debajo de una tolerancia se reemplaza por un cero, así como los vectores propios correspondientes; esta práctica se denomina *quenching*.

Aunque el *quenching* permite obtener la inversa para cualquier matriz, también descarta información de la matriz métrica original cuando remueve valores singulares o valores propios. De hecho, la \mathbf{G}^{-1} que se obtiene es una matriz singular ya que contiene ceros en algunos de sus valores propios o singulares, y estrictamente no puede reinvertirse para recuperar la matriz métrica original. Sin embargo, sí es posible llevar a cabo la reinversión debido a que la precisión numérica finita hace que haya un cero de máquina y no un cero formal. La matriz métrica obtenida tras la reinversión se denota como $\tilde{\mathbf{G}}$, y su diferencia respecto a la matriz métrica original es un indicador de la información perdida en el proceso de *quenching*. Un ejemplo se muestra para el caso de fullerenos en la tabla B.1, donde también se compara la matriz métrica perturbada tras aplicar la descomposición modificada de Cholesky. La descomposición de Cholesky simple no puede ser mostrada ya que su ejecución termina en error por los problemas de condicionamiento de la matriz métrica. En todos los casos la matriz recuperada por

ModChol es más semejante a la original que la recuperada por SVD o TED, como se muestra en su diferencia promedio. El número de condición de la matriz recuperada por ModChol es más bajo que el de la matriz métrica original y de TED, lo que muestra que se mejora la estabilidad numérica al aplicar ModChol. Aunque SVD tiene un número de condición más bajo, la matriz recuperada difiere significativamente respecto a la matriz métrica original, lo que demuestra que se perdió información durante el proceso de *quenching*. La idea de la pérdida de información se complementa con que SVD y TED descartan hasta el 15% de los valores propios en el caso del fullereno de 82 carbonos.

Tabla B.1. Comparación entre matrices métrica de fullerenos tras aplicar *quenching* con SVD y TED, y perturbar con ModChol, para después recuperar la matriz métrica modificada. “Vals.” representa valores singulares para SVD, y valores propios para TED y ModChol. En todos los casos ModChol mantiene más información de la matriz métrica original que SVD o TED. Para la comparación se utilizó base 6-311+G(d) y base auxiliar GEN-A2*.

C ₆₀			
$N_{aux} = 5640$	SVD	TED	ModChol
Número de Condición	6×10^{15}	1×10^{24}	9×10^{17}
Diferencia Absoluta Promedio	0.73	6.81×10^{-2}	1.9×10^{-12}
Desviación Estándar	1.90	0.12	2.8×10^{-11}
Vals < 0	–	210	212
Vals < Tol.	768	768	748
Tiempo (min)	18.36	8.88	0.6
C ₇₀			
$N_{aux} = 6580$	SVD	TED	ModChol
Número de Condición	2×10^{15}	1×10^{25}	7×10^{17}
Diferencia Absoluta Promedio	0.24	0.21	2×10^{-12}
Desviación Estándar	0.52	0.92	3×10^{-11}
Vals < 0	–	248	250
Vals < Tol.	946	946	907
Tiempo (min)	28.3	13.5	0.9
C ₈₂			
$N_{aux} = 7708$	SVD	TED	ModChol
Número de Condición	5×10^{15}	2×10^{25}	1×10^{18}
Diferencia Absoluta Promedio	0.35	6.1×10^{-2}	2.7×10^{-12}
Desviación Estándar	0.78	0.14	3.6×10^{-11}
Vals < 0	–	295	297
Vals < Tol.	1158	1158	1122
Tiempo (min)	46.7	22.15	1.4

Otras implementaciones

En el capítulo de resultados se comparó el algoritmo directo-w con una implementación convencional, *in-core* y directa-simple, cuyos pseudocódigos se describen aquí. Estas implementaciones se basan en el manejo directo de las AO-ERIs. Se ha mantenido la aproximación LDF y ModChol para que los algoritmos sean lo más cercanos al algoritmo directo-w y que la única diferencia sea respecto al manejo de las integrales; sin embargo la aproximación LDF puede no usarse y ModChol puede reemplazarse por la inversión de la matriz métrica para tener la versión elemental de estos algoritmos.

C.1 Implementaciones convencional e *in-core*

La implementación convencional e *in-core* comparten el hecho de que ambas guardan las AO-ERIs, por lo que su algoritmo es muy similar, sin embargo difieren en que la primera las guarda en disco y la segunda en RAM. Ambas ofrecen como ventaja el ahorro de operaciones de punto flotante ya que evalúan las integrales una sola vez, y posteriormente solo las leen y las usan varias veces; el algoritmo *in-core* solo es aplicable cuando hay suficiente RAM. Por otra parte, el algoritmo convencional es aplicable mientras haya suficiente memoria en disco, la cual suele ser bastante mayor, aunque también mucho más lenta. El pseudocódigo se muestra en el Algoritmo 2. Parte del código se comparte con el algoritmo directo-w, siendo la mayor diferencia los pasos 7 y 8 referentes a la evaluación de las AO-ERIs, y la contracción de un índice del ket.

Se inicia con la generación de la matriz métrica en los pasos del 1 al 3. Posteriormente, se realiza la contracción del *bra* en los pasos del 4 al 6. El paso 7 corresponde a las AO-ERIs, las cuales son generadas y guardadas en el paso 7a, o leídas en el paso 7b, en disco o en RAM según corresponda. En el paso 8 se genera la matriz $w_{\lambda Q}^s$ para un valor de s .^a Los pasos del 9 al 12 corresponden a la generación de la MO-ERI, y la contribución a Σ_{pp} . Finalmente se calcula ω y se revisa la convergencia.

^a El tensor de AO-ERIs tiene las mismas dimensiones que el tensor w de la implementación propuesta en este trabajo. No se considera el guardado de w en la implementación convencional mostrada puesto que se estaría usando el doble de memoria, y la finalidad es comparar la eficiencia de guardar el tensor w contra guardar las AO-ERIs.

Algoritmo 2 Implementación convencional e *in-core* de EP2

1. Reducir la dimensión de la base auxiliar de N_{aux} a N_{aux}^{LDF} mediante LDF.
 2. Calcular la matriz métrica \mathbf{G} $N_{aux}^{LDF^2}$
 3. Aplicar ModChol a la matriz métrica $\tilde{\mathbf{G}} = \mathbf{LDL}^T$ $N_{aux}^{LDF^3}$
 4. Calcular $z_{\nu P}^p = \sum_{\mu}^N C_{\mu p}(\mu\nu|P)$ $N_{aux}^{LDF} N^2$
 5. Calcular $y_{qP}^p = \sum_{\nu}^N C_{\nu q} z_{\nu P}^p$ $N_{aux}^{LDF} N^2$
 6. Resolver para \mathbf{x} el sistema $\mathbf{LDL}^T \mathbf{x}^{pT} = \mathbf{y}^{pT}$ $N_{aux}^{LDF^2} N$
 - for** s en N Orbitales Moleculares **do**
 - for** σ en N Orbitales Atómicos **do**
 - for** λ en N Orbitales Atómicos **do**
 - for** Q en N_{aux} Funciones Auxiliares **do**
 - if** no hay AO-ERI guardada **then**
 - 7a. Calcular y guardar AO-ERI ($Q|\sigma\lambda$).
 - else**
 - 7b. Leer AO-ERI ($Q|\sigma\lambda$).
 - end if**
 8. Calcular $w_{\lambda Q}^s \leftarrow \sum_{\sigma}^N C_{\sigma s}(Q|\sigma\lambda)$ $N_{aux} N^3$
 - end for**
 - end for**
 - end for**
 - if** s es un Orbital Ocupado i **then**
 - 9a. Tomar q en Orb. Virt.- a y $(pa|i\lambda) = \sum_Q (x^p)_{aQ} (w^{iT})_{Q\lambda}$ $N_{aux}^{LDF} N_{occ} N_{vir} N$
 - 10a. Contraer λ a Orb. Virt. mediante $(pa|ib) = \sum_{\lambda}^N C_{\lambda b}(pa|i\lambda)$ $N_{occ} N_{vir}^2 N$
 - 11a. Realizar la contribución a 2ph.
 - end if**
 - if** s es un Orbital Virtual a **then**
 - 9b. Tomar q en Orb. Occ.- i y $(pi|a\lambda) = \sum_Q (x^p)_{iQ} (w^{aT})_{Q\lambda}$ $N_{aux}^{LDF} N_{occ} N_{vir} N$
 - 10b. Contraer λ a Orb. Occ. mediante $(pi|aj) = \sum_{\lambda}^N C_{\lambda j}(pi|a\lambda)$ $N_{occ}^2 N_{vir} N$
 - 11b. Realizar la contribución a 2hp.
 - end if**
 - end for**
 12. Calcular la energía de EP2
 13. Revisar convergencia
-

C.2 Implementación directa simple

La implementación directa comparte la mayoría de los pasos con la implementación convencional, siendo la mayor diferencia que las AO-ERIs nunca se guardan, sino que se recalculan cada vez. Esta implementación evita el I/O disco, pero no considera las optimizaciones relativas al tensor w como ocurre en el caso del algoritmo directo- w ; sin embargo, su implementación a partir de los algoritmos convencional e *in-core* es muy fácil porque solo hay que cambiar la lectura/escritura de la AO-ERI por su reevaluación, a diferencia del algoritmo directo- w que está muy optimizado pero requiere reestructurar el algoritmo.

La representación en pseudocódigo del algoritmo directo simple se muestra en el Algoritmo 3. Los pasos del 1 al 3 se encargan de lo referente a la matriz métrica y los pasos del 4 al 6 realizan la contracción del *bra*. En el paso 7 se evalúa la AO-ERI requerida, esto se hace siempre puesto que nunca se guarda ni en disco ni en RAM. En el paso 8 se calcula la matriz $w_{\lambda Q}^s$ para un valor de s , y para cada valor se obtiene la correspondiente ERI intermedia en el paso 9. En los pasos 10 y 11 se realiza las contracciones que generan a las MO-ERIs, y estas se usan para la expresión de Σ_{pp} . Finalmente, en el paso 13 se calcula ω y en el paso 14 se revisa la convergencia.

Algoritmo 3 Implementación directa-simple de EP2

1. Reducir la dimensión de la base auxiliar de N_{aux} a N_{aux}^{LDF} mediante LDF.
 2. Calcular la matriz métrica \mathbf{G} $N_{aux}^{LDF^2}$
 3. Aplicar ModChol a la matriz métrica $\tilde{\mathbf{G}} = \mathbf{LDL}^T$ $N_{aux}^{LDF^3}$
 4. Calcular $z_{\nu P}^p = \sum_{\mu}^N C_{\mu p}(\mu\nu|P)$ $N_{aux}^{LDF} N^2$
 5. Calcular $y_{qP}^p = \sum_{\nu}^N C_{\nu q} z_{\nu P}^p$ $N_{aux}^{LDF} N^2$
 6. Resolver para \mathbf{x} el sistema $\mathbf{LDL}^T \mathbf{x}^{pT} = \mathbf{y}^{pT}$ $N_{aux}^{LDF^2} N$
 - for** s en N Orbitales Moleculares **do**
 - for** σ en N Orbitales Atómicos **do**
 - for** λ en N Orbitales Atómicos **do**
 - for** Q en N_{aux} Funciones Auxiliares **do**
 7. Calcular AO-ERIs $(Q|\sigma\lambda)$.
 8. Calcular $w_{\lambda Q}^s \leftarrow \sum_{\sigma}^N C_{\sigma s}(Q|\sigma\lambda)$ $N_{aux}^{LDF} N^3$
 - end for**
 - end for**
 - end for**
 - if** s es un Orbital Ocupado i **then**
 - 9a. Tomar q en Orb. Virt.- a y $(pa|i\lambda) = \sum_Q (x^p)_{aQ} (w^{iT})_{Q\lambda}$ $N_{aux}^{LDF} N_{occ} N_{vir} N$
 - 10a. Contraer λ a Orb. Virt. $(pa|ib) = \sum_{\lambda}^N C_{\lambda b}(pa|i\lambda)$ $N_{occ} N_{vir}^2 N$
 - 11a. Realizar la contribución a 2ph.
 - end if**
 - if** s es un Orbital Virtual a **then**
 - 9b. Tomar q en Orb. Occ.- i y $(pi|a\lambda) = \sum_Q (x^p)_{iQ} (w^{aT})_{Q\lambda}$ $N_{aux}^{LDF} N_{occ} N_{vir} N$
 - 10b. Contraer λ a Orb. Occ. $(pi|aj) = \sum_{\lambda}^N C_{\lambda j}(pi|a\lambda)$ $N_{occ}^2 N_{vir} N$
 - 11b. Realizar la contribución a 2hp.
 - end if**
 - end for**
 12. Calcular la energía de EP2
 13. Revisar convergencia
-

Referencias

- ¹T. Koopmans, “Über die zuordnung von wellenfunktionen und eigenwerten zu den einzelnen elektronen eines atoms,” *Physica*, 104–113.
- ²A. Szabo and N. S. Ostlund, *Modern quantum chemistry : introduction to advanced electronic structure theory* (Dover Publications, 1996), p. 466.
- ³J. V. Ortiz, “Electron propagator theory: an approach to prediction and interpretation in quantum chemistry,” *Wiley Interdisciplinary Reviews: Computational Molecular Science* **3**, 123–142 (2013).
- ⁴H. H. Corzo and J. V. Ortiz, “Electron Propagator Theory: Foundations and Predictions,” *Advances in Quantum Chemistry* **74**, 267–298 (2017).
- ⁵C. M. Oana and A. I. Krylov, “Dyson orbitals for ionization from the ground and electronically excited states within equation-of-motion coupled-cluster formalism: Theory, implementation, and examples,” *Journal of Chemical Physics* **127**, 1–14 (2007).
- ⁶M. Díaz-Tinoco, H. H. Corzo, F. Pawłowski, and J. V. Ortiz, “Do Dyson Orbitals resemble canonical Hartree–Fock orbitals?” *Molecular Physics* **8976** (2018).
- ⁷L. S. Cederbaum, “One-body Green’s function for atoms and molecules: Theory and application,” *Journal of Physics B: Atomic and Molecular Physics* **8**, 290–303 (1975).
- ⁸J. V. Ortiz, “Partial third-order quasiparticle theory: Comparisons for closed-shell ionization energies and an application to the Borazine photoelectron spectrum,” *Journal of Chemical Physics* **104**, 7599–7605 (1996).
- ⁹J. V. Ortiz, “A nondiagonal, renormalized extension of partial third-order quasiparticle theory: Comparisons for closed-shell ionization energies,” *Journal of Chemical Physics* **108**, 1008–1014 (1998).
- ¹⁰J. V. Ortiz, “An efficient, renormalized self-energy for calculating the electron binding energies of closed-shell molecules and anions,” *International Journal of Quantum Chemistry* **105**, 803–808 (2005).
- ¹¹W. von Niessen, J. Schirmer, and L. S. Cederbaum, “Computational methods for the one-particle green’s function,” *Computer Physics Reports* **1**, 57–125 (1984).
- ¹²I. Thanopoulos, “Outer-valence Green’s function method using natural orbitals for ultrafast electron density dynamics,” *Computational and Theoretical Chemistry* **970**, 42–47 (2011).
- ¹³S. H. Cheng and N. J. Higham, “A Modified Cholesky Algorithm Based on a Symmetric Indefinite Factorization,” *SIAM Journal on Matrix Analysis and Applications* **19**, 1097–1110 (1998).

- ¹⁴N. H. F. Beebe and J. Linderberg, "Simplifications in the generation and transformation of two-electron integrals in molecular calculations," *International Journal of Quantum Chemistry* **12**, 683–705 (1977).
- ¹⁵V. P. Vysotskiy and L. S. Cederbaum, "On the Cholesky decomposition for electron propagator methods: General aspects and application on C60," *The Journal of Chemical Physics* **132**, 044110 (2010).
- ¹⁶R. Polly, H.-J. Werner, F. R. Manby, and P. J. Knowles, "Fast Hartree–Fock theory using local density fitting approximations," *Molecular Physics* **102**, 2311–2321 (2004).
- ¹⁷D. Young, *Computational Chemistry: A Practical Guide for Applying Techniques to Real World Problems* (2001), pp. 322–359.
- ¹⁸M. J. Frisch, M. Head-Gordon, and J. A. Pople, "Semi-direct algorithms for the MP2 energy and gradient," *Chemical Physics Letters* **166**, 281–289 (1990).
- ¹⁹R. Ahlrichs, M. Bär, M. Häser, H. Horn, and C. Kölmel, "Electronic structure calculations on workstation computers: The program system turbomole," *Chemical Physics Letters* **162**, 165–169 (1989).
- ²⁰H. B. Schlegel, *Geometry Optimization on Potential Energy Surfaces* (1995), pp. 459–500.
- ²¹M. W. Feyereisen, R. A. Kendall, J. Nichols, D. Dame, and J. T. Golab, "Implementation of the direct SCF and RPA methods on loosely coupled networks of workstations," *Journal of Computational Chemistry* **14**, 818–830 (1993).
- ²²J. Almlöf, K. Faegri, and K. Korsell, "Principles for a direct SCF approach to LICAO–MOab-initio calculations," *Journal of Computational Chemistry* **3**, 385–399 (1982).
- ²³M. Häser and R. Ahlrichs, "Improvements on the direct SCF method," *Journal of Computational Chemistry* **10**, 104–111 (1989).
- ²⁴R. Flores-Moreno and J. V. Ortiz, "Integral approximations in ab initio, electron propagator calculations," *Journal of Chemical Physics* **131** (2009).
- ²⁵U. Bozkaya, "Derivation of general analytic gradient expressions for density-fitted post-Hartree-Fock methods: An efficient implementation for the density-fitted second-order Møller–Plesset perturbation theory," *The Journal of Chemical Physics* **141**, 124108 (2014).
- ²⁶A. P. Rendell and T. J. Lee, "Coupled-cluster theory employing approximate integrals: An approach to avoid the input/output and storage bottlenecks," *The Journal of Chemical Physics* **101**, 400–408 (1994).
- ²⁷D. Cremer and J. Gauss, "An unconventional scf method for calculations on large molecules," *Journal of Computational Chemistry* **7**, 274–282 (1986).
- ²⁸V. G. Zakrzewski and J. V. Ortiz, "Semidirect algorithms in electron propagator calculations," *International Journal of Quantum Chemistry* **52**, 23–27 (1994).
- ²⁹V. G. Zakrzewski and J. V. Ortiz, "Semidirect algorithms for third-order electron propagator calculations," *International Journal of Quantum Chemistry* **53**, 583–590 (1995).

- ³⁰V. Zakrzewski and J. Ortiz, “Semidirect electron propagator calculations on chlorobenzene ionization energies,” *Journal of Molecular Structure: THEOCHEM* **388**, 351–357 (1996).
- ³¹J. Villalobos-Castro, J. A. Guerrero-Cruz, B. Zuniga-Gutierrez, and R. Flores-Moreno, “Local density fitting applied to second order propagator equations for core electron binding energies,” *Computational and Theoretical Chemistry* **1136-1137**, 29–33 (2018).
- ³²J. L. Whitten, “Coulombic potential energy integrals and approximations,” *The Journal of Chemical Physics* **58**, 4496–4501 (1973).
- ³³B. I. Dunlap, J. W. D. Connolly, and J. R. Sabin, “On some approximations in applications of $X\alpha$ theory,” *The Journal of Chemical Physics* **71**, 3396–3402 (1979).
- ³⁴O. Vahtras, J. Almlöf, and M. Feyereisen, “Integral approximations for LCAO-SCF calculations,” *Chemical Physics Letters* **213**, 514–518 (1993).
- ³⁵J. R. Bunch and L. Kaufman, “Some stable methods for calculating inertia and solving symmetric linear systems,” *Mathematics of Computation* **31**, 163–163 (1977).
- ³⁶R. M. Parrish, L. A. Burns, D. G. A. Smith, et al., “Psi4 1.1: An Open-Source Electronic Structure Program Emphasizing Automation, Advanced Libraries, and Interoperability,” *Journal of Chemical Theory and Computation* **13**, 3185–3197 (2017).
- ³⁷A. D. Becke, “Density-functional thermochemistry. iii. the role of exact exchange,” en, *The Journal of Chemical Physics* **98**, 5648–5652 (1993).
- ³⁸C. Lee, W. Yang, and R. G. Parr, “Development of the colle-salvetti correlation-energy formula into a functional of the electron density,” en, *Physical Review B* **37**, 785–789 (1988).
- ³⁹R. Flores-Moreno, J. A. Guerrero-Cruz, H. González-Ramírez, et al., *Nagual*, Guadalajara, México, 2018.
- ⁴⁰J. Andzelm, E. Radzio, and D. R. Salahub, “Compact basis sets for LCAO-LSD calculations. Part I: Method and bases for Sc to Zn,” *Journal of Computational Chemistry* **6**, 520–532 (1985).
- ⁴¹J. Andzelm, N. Russo, and D. R. Salahub, “Ground and excited states of group IVA diatomics from local-spin-density calculations: Model potentials for Si, Ge, and Sn,” *The Journal of Chemical Physics* **87**, 6562–6572 (1987).
- ⁴²P. Calaminici, F. Janetzko, A. M. Köster, R. Mejia-Olvera, and B. Zuniga-Gutierrez, “Density functional theory optimized basis sets for gradient corrected functionals: 3d transition metal systems,” *The Journal of Chemical Physics* **126**, 044108 (2007).
- ⁴³R. Krishnan, J. S. Binkley, R. Seeger, and J. A. Pople, “Self-consistent molecular orbital methods. XX. A basis set for correlated wave functions,” *The Journal of Chemical Physics* **72**, 650–654 (1980).
- ⁴⁴T. Clark, J. Chandrasekhar, G. W. Spitznagel, and P. V. R. Schleyer, “Efficient diffuse function-augmented basis sets for anion calculations. III. The 3-21+G basis set for first-row elements, Li-F,” *Journal of Computational Chemistry* **4**, 294–301 (1983).

- ⁴⁵T. H. Dunning, “Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen,” *The Journal of Chemical Physics* **90**, 1007–1023 (1989).
- ⁴⁶R. A. Kendall, T. H. Dunning, and R. J. Harrison, “Electron affinities of the first-row atoms revisited. {Systematic} basis sets and wave functions,” *The Journal of Chemical Physics* **96**, 6796–6806 (1992).
- ⁴⁷D. Mejía-Rodríguez and A. M. Köster, “Robust and efficient variational fitting of Fock exchange,” *The Journal of Chemical Physics* **141**, 124114 (2014).
- ⁴⁸M. Kupferschmid, *Classical Fortran* (CRC Press, Jan. 2009).

