



UNIVERSIDAD
DE GRANADA

Escuela Técnica Superior de Ingenierías Informática y de
Telecomunicación y Facultad de Ciencias

GRADO EN INGENIERÍA INFORMÁTICA Y MATEMÁTICAS

TRABAJO DE FIN DE GRADO

Revisión bibliográfica sobre dinámica molecular. Modelización, algoritmos y aplicaciones

Presentado por:
Joaquín Arcila Pérez

Curso académico 2024-2025



Revisión bibliográfica sobre dinámica molecular. Modelización, algoritmos y aplicaciones

Joaquín Arcila Pérez

Joaquín Arcila Pérez *Revisión bibliográfica sobre dinámica molecular. Modelización, algoritmos y aplicaciones.*

Trabajo de fin de Grado. Curso académico 2024-2025.

**Responsable de
tutorización**

Juan Calvo Yagüe
Departamento de Matemática Aplicada

Lázaro René Izquierdo Fábregas
Departamento de Matemática Aplicada

Grado en Ingeniería
Informática y Matemáticas

Escuela Técnica Superior
de Ingenierías Informática
y de Telecomunicación y
Facultad de Ciencias

Universidad de Granada

DECLARACIÓN DE ORIGINALIDAD

D. Joaquín Arcila Pérez

Declaro explícitamente que el trabajo presentado como Trabajo de Fin de Grado (TFG), correspondiente al curso académico 2024-2025, es original, entendido esto en el sentido de que no he utilizado para la elaboración del trabajo fuentes sin citarlas debidamente.

En Granada a 14 de julio de 2025

Fdo: Joaquín Arcila Pérez

*A mis padres, por su esfuerzo, dedicación y la confianza que siempre han depositado en mí.
A mis hermanos, por ser un ejemplo a seguir y por haberme guiado a lo largo del camino.
A mi familia y a mis amigos, por estar siempre ahí y por animarme cada día a alcanzar mis metas.
A todos mis profesores, especialmente a mis tutores, Juan y René, y a mi tío Alberto, por su paciencia
y el interés mostrado en mi proceso de aprendizaje.*

Índice general

Summary	VII
Resumen	IX
Introducción	XI
Planificación y Presupuesto	XIII
1. Contexto histórico	1
2. Fundamentos de Dinámica Molecular	3
2.1. Principios de la mecánica clásica	3
2.1.1. Leyes de Newton	3
2.1.2. Principio de conservación de la energía	4
2.1.3. Introducción a las ecuaciones de movimiento de Newton	5
3. Modelización en Dinámica Molecular	7
3.1. Modelos de interacción	7
3.1.1. Campos de fuerza	7
3.2. Representación de sistemas moleculares	11
3.2.1. Modelos atomísticos vs coarse-graining	11
3.2.2. Cajas de simulación	12
3.2.3. Condiciones de contorno	12
3.2.4. Métodos de truncamiento	13
4. Algoritmos en Dinámica Molecular	15
4.1. Integración de ecuaciones de movimiento	15
4.1.1. Algoritmo de Verlet	19
4.1.2. Velocity-Verlet	20
4.1.3. Leap-Frog	21
4.2. Control de temperatura y presión	23
4.2.1. Termostatos	24
4.2.2. Barostatos	27
4.3. Algoritmos de optimización	29
4.3.1. Minimización de energía	30
4.3.2. Métodos de Monte Carlo	33
4.4. Simulaciones paralelas y aceleración	35
5. Implementación de una Simulación de Dinámica Molecular	37
5.1. Inicialización del sistema	39
5.2. Cálculo de fuerzas	41
5.3. Integración de las ecuaciones de movimiento	43
5.4. Programas de dinámica molecular	44

6. Aplicaciones de la Dinámica Molecular	47
6.1. Bioquímica y biología estructural	47
6.1.1. Moléculas biológicas: conceptos básicos	47
6.1.2. Aplicaciones en simulación biomolecular	48
6.2. Materiales y nanotecnología	50
6.3. Química y física de fluidos	52
6.4. Estudios de energía y catalizadores	52
7. Desafíos y futuras direcciones	55
7.1. Limitaciones actuales	55
7.1.1. Escalabilidad	55
7.1.2. Precisión de los modelos de fuerza	55
7.1.3. Tiempo de simulación	56
7.2. Nuevas tendencias	56
7.2.1. Métodos híbridos clásico-cuánticos (QM/MM)	56
7.2.2. Aprendizaje automático para generación de campos de fuerza	57
7.2.3. Simulaciones multiescala y a gran escala	57
7.3. Oportunidades de mejora	58
7.3.1. Nuevos algoritmos e integradores simplécticos	58
7.3.2. Avances en el diseño de campos de fuerza	59
8. Conclusiones	61
A. Código	63
Glosario	67
Bibliografía	71

Summary

This Final Degree Project has focused on the study of molecular dynamics (MD), a computational technique that enables the modeling of atomic and molecular systems based on their interactions. Throughout the project, a comprehensive analysis of this methodology has been carried out, covering everything from its theoretical foundations to its various practical applications, with particular attention to the algorithmic tools used in simulations.

First, the historical and scientific context in which molecular dynamics emerged has been introduced, from the pioneering experiments of Alder and Wainwright in the 1950s, through the fundamental contributions of Rahman and Verlet, to the technological advances that enabled the application of these methods to the study of complex biological systems and materials using high-performance computing (HPC) and graphics processing units (GPUs). In summary, the milestones that allowed the transition from simple hard-sphere gas simulations to advanced simulations of proteins, solvents, or crystalline networks have been identified, consolidating molecular dynamics as a key tool in materials science, biophysics, and molecular design.

Next, the project addresses the foundations of classical and statistical mechanics that form the basis of simulations. On one hand, it analyzes how Newton's laws lead to the equations of motion governing the evolution of a particle system. These equations allow the problem to be formulated as a Hamiltonian system, whose mathematical structure imposes certain properties that must be respected by numerical algorithms. One of the most relevant is symplecticity—a property that must be preserved by integration methods to ensure physically coherent long-term simulations. In this context, the Verlet algorithm and its variants have been studied, highlighting their ability to conserve system energy over time and their numerical stability compared to non-symplectic methods such as the Euler integrator. The importance of selecting an appropriate integration step to ensure both stability and accuracy has also been discussed.

On the other hand, the role of interaction potentials in molecular dynamics has been analyzed, particularly classical force fields. These models approximate intermolecular forces through terms representing bonds, angles, torsions, as well as Van der Waals and electrostatic interactions. It has been emphasized how the choice of force field directly affects the precision and applicability of the simulation, making proper parametrization essential to faithfully represent the studied systems. Additionally, the importance of the level of detail with which the system is represented has been highlighted, addressing the trade-off between accuracy and computational cost.

An essential part of the work has been the discussion of the different thermodynamic ensembles (NVE, NVT, NPT) and how they are implemented in practice through the use of thermostats and barostats. The most commonly used methods have been examined, such as the Berendsen and Nosé-Hoover thermostats, explaining their advantages, limitations, and their impact on the system's dynamics and statistics. Regarding barostats, the functioning of

Summary

the Berendsen and Parrinello-Rahman methods has been explained, the latter being especially useful in simulations of materials under stress.

Subsequently, the issue of simulation initialization has been addressed, highlighting the importance of performing a prior energy minimization step to avoid artificial configurations with high energies or particle overlaps. This step relies on optimization methods, such as gradient descent or conjugate gradient, which allow for the identification of stable configurations corresponding to local minima on the potential energy surface. In complex systems, where this surface presents multiple local minima, the role of metaheuristics has been emphasized as tools to escape from these minima and approach the global minimum. In this context, Monte Carlo methods such as the Metropolis algorithm have also been introduced, offering an alternative to molecular dynamics by generating random system configurations based on probabilistic criteria.

A key aspect addressed in the project has been the use of reduced units in simulations. This technique allows physical quantities to be expressed in a dimensionless form by using characteristic parameters of the system as scaling factors. This not only simplifies the physical expressions used but also improves numerical stability and facilitates comparison between different simulations. Furthermore, it allows code generalization, since results can later be rescaled to real physical units if desired. Finally, a pseudo-algorithm has been designed to perform a molecular dynamics simulation using this unit system.

The great potential of molecular dynamics in real-world applications is also highlighted. Its uses in materials science, computational chemistry, structural biology, and particularly in drug design have been discussed. In the latter field, simulations help understand ligand-protein binding, evaluate the stability of molecular complexes, and investigate interactions at the atomic level. These capabilities are essential to accelerate the discovery of new treatments for diseases such as cancer, HIV, and neurodegenerative disorders.

Finally, the project dedicates a section to the current limitations of molecular dynamics. It analyzes how computational scalability remains a challenge, especially for large or long-duration simulations. It also points out the inability of classical force fields to accurately describe quantum phenomena, which has motivated the development of hybrid approaches such as the QM/MM method. Based on these limitations, several opportunities for improvement are identified, focusing on the application of artificial intelligence techniques, the integration of multiscale methods, and the exploitation of high-performance architectures—thus reflecting the great potential of molecular dynamics.

In summary, this work provides an overview of molecular dynamics, covering its theoretical foundations, computational methods, techniques for regulating thermodynamic conditions, optimization strategies, and major applications in modern science. It concludes that, despite its limitations, molecular dynamics is a fundamental tool for studying complex systems at the atomic scale, with significant room for advancement.

Resumen

Este Trabajo Fin de Grado se ha centrado en el estudio de la dinámica molecular (MD), una técnica computacional que permite modelar sistemas atómicos y moleculares a partir de sus interacciones. A lo largo del trabajo, se ha llevado a cabo un análisis exhaustivo de esta metodología, abordando desde sus fundamentos teóricos hasta sus distintas aplicaciones prácticas, con especial atención a las herramientas algorítmicas empleadas en las simulaciones.

En primer lugar, se ha introducido el contexto histórico y científico en el que surge la dinámica molecular, desde los experimentos pioneros de Alder y Wainwright en los años 50, pasando por las contribuciones fundamentales de Rahman y Verlet, hasta los avances tecnológicos que posibilitaron la aplicación de estos métodos al estudio de sistemas biológicos complejos y materiales mediante el uso de ordenadores de alto rendimiento (HPC) y unidades de procesamiento gráfico (GPUs). En resumen, se han identificado los hitos que permitieron pasar de simulaciones simples de gases duros a simulaciones avanzadas de proteínas, disolventes o redes cristalinas, consolidando la dinámica molecular como una herramienta clave en ciencia de materiales, biofísica y diseño molecular.

Seguidamente, el trabajo aborda los fundamentos de la mecánica clásica y estadística que sustentan las simulaciones. Por un lado, se ha analizado cómo, a partir de las leyes de Newton, se establecen las ecuaciones de movimiento que gobiernan la evolución de un sistema de partículas. Estas ecuaciones permiten formular el problema como un sistema hamiltoniano, cuya estructura matemática impone ciertas propiedades que deben ser respetadas por los algoritmos numéricos. Una de las más relevantes es la simplecticidad, propiedad que debe preservarse en los métodos de integración para garantizar una simulación físicamente coherente a largo plazo. En este contexto, se han estudiado el algoritmo de Verlet y sus variantes, destacando su capacidad para conservar la energía del sistema a largo plazo y su estabilidad numérica frente a otros métodos no simplécticos como el integrador de Euler. Asimismo, se ha explicado la importancia de seleccionar un paso de integración adecuado para asegurar tanto la estabilidad como la precisión de la simulación.

Por otro lado, se ha analizado el papel de los potenciales de interacción en dinámica molecular, en particular los campos de fuerza clásicos. Estos modelos permiten aproximar las fuerzas intermoleculares mediante términos que representan enlaces, ángulos, torsiones, así como interacciones de Van der Waals y electrostáticas. Se ha destacado cómo la elección del campo de fuerza condiciona directamente la precisión y aplicabilidad de la simulación, siendo imprescindible una parametrización adecuada para representar con fidelidad los sistemas estudiados. Además, se ha puesto de manifiesto la importancia del nivel de detalle con el que se representa el sistema, abordando el *trade-off* entre precisión y coste computacional.

Una parte esencial del trabajo ha sido la discusión de los diferentes colectivos termodinámicos (NVE, NVT, NPT) y cómo se implementan en la práctica mediante el uso de termostatos y barostatos. Se ha profundizado en los métodos más utilizados, como los termostatos de Berendsen y Nosé-Hoover, explicando sus ventajas, limitaciones y su impacto sobre la dinámica

y la estadística del sistema. En el caso de los barostatos, se ha explicado el funcionamiento del método de Berendsen y del método de Parrinello-Rahman, este último especialmente útil en simulaciones de materiales sometidos a tensiones.

Posteriormente, se ha abordado el problema de la inicialización de las simulaciones, destacando la importancia de realizar una etapa previa de minimización de energía para evitar configuraciones artificiales con altas energías o solapamientos entre partículas. Esta etapa se basa en métodos de optimización, como el descenso por gradiente o el gradiente conjugado, que permiten encontrar configuraciones estables correspondientes a mínimos locales de la superficie de energía potencial. En sistemas complejos, donde dicha superficie presenta múltiples mínimos locales, se ha resaltado el papel de las metaheurísticas como herramientas para escapar de estos mínimos y aproximarse al mínimo global. En este contexto, también se han presentado los métodos de Monte Carlo, como el algoritmo de Metropolis, los cuales ofrecen una alternativa a la dinámica molecular al generar configuraciones aleatorias del sistema según criterios probabilísticos.

Un aspecto clave tratado en el trabajo ha sido el uso de unidades reducidas en las simulaciones. Esta técnica permite expresar las magnitudes físicas de forma adimensional, utilizando como escala parámetros característicos del sistema. Esto no solo simplifica las expresiones físicas empleadas, sino que también mejora la estabilidad numérica y facilita la comparación entre diferentes simulaciones. Además, permite generalizar el código, ya que los resultados pueden reescalarsen posteriormente a unidades físicas reales si se desea. Finalmente, se ha diseñado un pseudo-algoritmo que realiza una simulación de dinámica molecular que pone en práctica este sistema de unidades.

A continuación, se señala el gran potencial de la dinámica molecular en aplicaciones reales. Se han comentado sus usos en ciencia de materiales, química computacional, biología estructural y, en particular, en el diseño de fármacos. En este último ámbito, se ha discutido cómo las simulaciones permiten entender el acoplamiento entre ligandos y proteínas, evaluar la estabilidad de complejos moleculares e investigar las interacciones a nivel atómico. Estas capacidades son fundamentales para acelerar el descubrimiento de nuevos tratamientos frente a enfermedades como el cáncer, el VIH o enfermedades neurodegenerativas.

Finalmente, el trabajo dedica una sección a las limitaciones actuales de la dinámica molecular. Se ha analizado cómo la escalabilidad computacional continúa siendo un desafío, especialmente en simulaciones de gran tamaño o de larga duración temporal. También se ha señalado la incapacidad de los campos de fuerza clásicos para describir con precisión fenómenos de naturaleza cuántica, lo que ha motivado el desarrollo de enfoques híbridos como el método QM/MM. A partir de estas limitaciones, se identifican diversas oportunidades de mejora, centradas en la aplicación de técnicas de inteligencia artificial, integración de métodos multiescala y en el aprovechamiento de arquitecturas de alto rendimiento, reflejando así el gran potencial de la dinámica molecular.

En resumen, este trabajo ofrece una visión general de la dinámica molecular, abarcando sus bases teóricas, sus métodos computacionales, las técnicas para regular condiciones termodinámicas, las estrategias de optimización y las principales aplicaciones en ciencia moderna. Se concluye que, pese a sus limitaciones, la dinámica molecular es una herramienta fundamental para el estudio de sistemas complejos a escala atómica, con un amplio margen de mejora.

Introducción

La dinámica molecular (MD) es una técnica computacional que permite estudiar el comportamiento de sistemas atómicos y moleculares mediante la simulación de sus interacciones y movimientos a lo largo de un tiempo determinado. Dicha técnica se basa en la resolución de las ecuaciones de movimiento de Newton, a partir de las cuales consigue proporcionar información detallada sobre la estructura, las propiedades y la evolución de los sistemas planteados.

Las simulaciones computacionales actúan como puente (ver **Figura 1**) entre las escalas microscópicas de longitud y tiempo y el mundo macroscópico del laboratorio, así como entre la teoría y la experimentación, ya que permiten validar modelos teóricos, contrastándolos con datos experimentales, y explorar fenómenos en condiciones extremas de temperatura o de presión, a las que sería prácticamente imposible llegar en el laboratorio. Además, la MD ofrece grandes ventajas frente a otros métodos de simulación, como el de Monte Carlo (MC) [1], al proporcionar acceso a propiedades dinámicas del sistema, esencial en el escenario planteado.

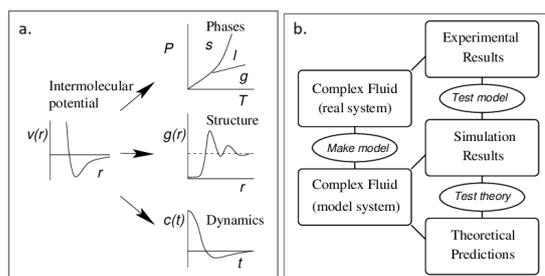


Figura 1.: Simulaciones como puente entre (a) lo microscópico y lo macroscópico; (b) la teoría y el experimento [2].

El objetivo principal de este trabajo es proporcionar al lector un conocimiento detallado sobre la dinámica molecular, abordando tanto sus fundamentos teóricos como su aplicación práctica. Además de ampliar la comprensión sobre esta técnica, se busca enfatizar la relevancia de la investigación y el desarrollo en este campo, así como mostrar su evolución histórica y las tendencias actuales, incluyendo la integración de inteligencia artificial.

El contenido del trabajo se estructura en siete capítulos:

- El **Capítulo 1** ofrece un recorrido histórico por el desarrollo de la dinámica molecular, donde se presentan los hitos más relevantes, desde las primeras simulaciones realizadas por Alder y Wainwright hasta las recientes mejoras basadas en inteligencia artificial. Este capítulo contextualiza la evolución de la técnica en paralelo con los avances en computación y teoría física.

- En el **Capítulo 2** se introducen los fundamentos físicos que sustentan la dinámica molecular, analizándose las leyes y ecuaciones de Newton, así como los conceptos clave relacionados con la energía, la temperatura y las condiciones iniciales del sistema.
- El **Capítulo 3** trata sobre la modelización en dinámica molecular. Se profundiza en los distintos potenciales de interacción y condiciones de contorno, y se evidencia la importancia de la representación del sistema en MD.
- El **Capítulo 4** está dedicado al estudio y análisis de los principales algoritmos empleados en dinámica molecular. Se presentan los métodos numéricos utilizados para integrar las ecuaciones de movimiento, con especial atención al algoritmo de Verlet y sus variantes. Asimismo, se abordan técnicas para el control de las condiciones termodinámicas del sistema mediante termostatos (como los de Berendsen y Nosé–Hoover) y barostatos (como Andersen y Parrinello–Rahman). También se exploran algoritmos de optimización, cuyo objetivo es hallar configuraciones de mínima energía. Finalmente, se discuten estrategias para mejorar el rendimiento computacional mediante el uso de simulaciones paralelas y técnicas de aceleración por hardware, como el empleo de unidades de procesamiento gráfico (GPU) y arquitecturas de alto rendimiento (HPC).
- En el **Capítulo 5** se presenta una posible implementación de un programa de dinámica molecular, exponiendo un pseudo-algoritmo. Este capítulo permite ilustrar de forma concreta los aspectos técnicos tratados en los anteriores.
- En el **Capítulo 6** se exploran algunas de las principales aplicaciones de la dinámica molecular, con especial atención a su uso en biología computacional, simulación de proteínas y diseño de fármacos. Se destaca cómo la MD ha contribuido significativamente a avances científicos y tecnológicos.
- Finalmente, el **Capítulo 7** aborda las tendencias actuales y futuras del campo. Se discute la integración de la inteligencia artificial y el aprendizaje automático en la mejora de potenciales y análisis de resultados, así como los enfoques híbridos que combinan dinámica molecular clásica con métodos cuánticos y redes neuronales.

A lo largo del documento, se verá reflejado el carácter interdisciplinar de la dinámica molecular, combinando física, química, matemáticas e informática con el fin de afrontar problemas complejos en ciencia e ingeniería.

Planificación y Presupuesto

La planificación de este proyecto se ha organizado dividiendo el trabajo en diferentes tareas, las cuales se han distribuido a lo largo de las semanas de duración del proyecto. Para mejorar la claridad, dicha planificación se ha representado mediante un diagrama de Gantt, permitiendo así visualizar las dependencias entre tareas y asegurar una gestión eficiente del tiempo.



Figura 2.: Diagrama de Gantt del proyecto

En el diagrama se puede observar cómo se han planificado y distribuido las distintas tareas del proyecto a lo largo de 22 semanas. En primer lugar, se lleva a cabo un estudio general de los objetivos del proyecto y, a continuación, se realiza una revisión bibliográfica relacionada con dichos objetivos.

Luego, se estudian las bases teóricas de la dinámica molecular, profundizando en la modelización, modelos de interacción y representación de sistemas moleculares. Estas tareas se extienden, aproximadamente, hasta la semana 9, y sirven como base para los siguientes bloques del proyecto.

Seguidamente, se introduce la etapa en la que se invierte más tiempo: el estudio de los algoritmos empleados en dinámica molecular. Esta fase se desarrolla durante 4 semanas, diviendo su contenido en el análisis de los métodos de integración utilizados, los mecanismos de control de temperatura y presión, los algoritmos de optimización y una revisión de las simulaciones paralelas y el uso de aceleración por hardware.

Entre las semanas 14 y 17 se diseña un algoritmo para realizar una simulación de dinámica molecular, incluyendo el desarrollo del pseudo-código correspondiente y un análisis de las unidades físicas utilizadas en dichas simulaciones.

Finalmente, se lleva a cabo un estudio de las aplicaciones de la dinámica molecular, así como de los desafíos actuales y futuras posibles líneas de investigación. Tras dicho estudio,

Planificación y Presupuesto

se completan las secciones restantes de la memoria y se realiza una revisión general del proyecto.

Esta planificación organizada ayuda a que el proyecto avance de forma clara y ordenada, permitiendo ir incorporando conocimientos de forma progresiva.

En cuanto al presupuesto del proyecto, se tienen en cuenta los siguientes factores:

- **Coste de personal:** El proyecto ha sido desarrollado por una única persona, asumiendo el rol de ingeniero informático junior. Considerando un salario bruto medio de 2.250€ mensuales y una duración estimada del proyecto de 5 meses, el coste total de personal asciende a **11.250€**.
- **Coste de Hardware y Software:** El único coste a considerar en este caso es el del equipo empleado para desarrollar el proyecto, un MSI GF63 Thin 10SCXR, cuyo precio aproximado es de **800€**.

Por tanto, el coste total del proyecto es de, aproximadamente, **12.050€**.

1. Contexto histórico

El desarrollo de la dinámica molecular ha estado ligado al avance de la computación y la física teórica. Las primeras simulaciones se dieron en el año 1957, cuando Alder y Wainwright simularon un gas de esferas duras, demostrando la existencia de transiciones de fase mediante métodos computacionales [3]. Seguidamente, en el año 1964, Rahman realizó la primera simulación con un potencial de Lennard-Jones, modelando el comportamiento del argón líquido [4].

En los años 70 se introdujeron los primeros métodos de integración numérica, como el algoritmo de Verlet [5], que permitió cálculos más precisos y estables. Con ello, la dinámica molecular se empezó a aplicar en sólidos, líquidos y sistemas biológicos en los años 1976-1979. Luego, en el año 1985, Car y Parrinello desarrollaron la dinámica molecular *ab initio*, integrando métodos cuánticos en la simulación clásica [6]. Dicho método es útil si se busca comparar directamente los resultados de la simulación con mediciones experimentales en materiales específicos, sin embargo, tiene un gran costo computacional. Por ello, para lograr un equilibrio entre precisión y eficiencia computacional, nos centraremos en la dinámica molecular clásica, adecuada para el análisis de fenómenos generales y la comparación de diferentes teorías, siempre que el modelo utilizado represente correctamente los principios físicos esenciales del sistema en estudio.

En la década de 1990 y principios de los 2000, la MD se convirtió en una herramienta fundamental para la simulación de proteínas y ADN, impulsando avances en biomedicina y el desarrollo de fármacos. A partir de 2010, la incorporación de la computación de alto rendimiento (HPC) y los aceleradores GPU ha revolucionado el campo, permitiendo la simulación de sistemas con millones de átomos y reduciendo drásticamente los tiempos de cálculo. Mientras que una simulación de 10 nanosegundos solía requerir aproximadamente una semana, ahora puede completarse en tan solo 12 horas, lo que ha posibilitado simulaciones más precisas y extendidas hasta escalas del orden de los microsegundos.

Debido al gran impacto que ha tenido la dinámica molecular, siendo clave en proyectos galardonados con el Premio Nobel de Química, como el de Karplus, Levitt y Warshel en 2013 "*for the development of multiscale models for complex chemical systems*", hoy en día se están integrando la inteligencia artificial y el aprendizaje automático con el objetivo de mejorar la precisión de los potenciales de interacción, además de explorar nuevos enfoques híbridos, combinando MD con métodos cuánticos y redes neuronales.

2. Fundamentos de Dinámica Molecular

Para comprender el funcionamiento de la dinámica molecular y los algoritmos presentados en este trabajo, es fundamental conocer sus principios básicos. En este capítulo, se introducen los conceptos esenciales que sustentan la dinámica molecular.

2.1. Principios de la mecánica clásica

La dinámica molecular se fundamenta en la mecánica clásica, una rama de la física desarrollada principalmente por Isaac Newton en el siglo XVII, que estudia el movimiento de los cuerpos bajo la acción de fuerzas. Su formulación se apoya en las leyes de Newton [7], así como en los principios de conservación de la energía y del momento, y en las ecuaciones de movimiento que rigen la evolución de un sistema en el tiempo.

2.1.1. Leyes de Newton

En MD, cada átomo o molécula de un sistema se modela como una partícula clásica que se mueve según las leyes de Newton, también conocidas como leyes del movimiento de Newton. Por tanto, las tres leyes del movimiento de Newton son los fundamentos sobre los cuales se basa la dinámica molecular:

Primera Ley de Newton

La Primera Ley de Newton, conocida como el Principio de Inercia, establece que un cuerpo mantiene su estado de reposo o de movimiento rectilíneo uniforme a menos que una fuerza externa actúe sobre él. En [8], Newton lo enuncia como “*Corpus omne perseverare in statu suo quiescendi vel movendi uniformiter in directum, nisi quatenus illud a viribus impressis cogitur statum suum mutare*”, lo que implica que un objeto no cambiará su estado de movimiento a menos que una fuerza lo obligue a hacerlo. Aplicado al contexto de la dinámica molecular, esto significa que, en ausencia de interacciones con otras partículas, los átomos seguirían trayectorias rectilíneas con velocidad constante dentro de la simulación.

Segunda Ley de Newton

Dado que en la dinámica molecular clásica la masa de cada partícula se considera un parámetro constante, la Segunda Ley de Newton, también conocida como el Principio Fundamental de la Dinámica, establece que la aceleración de un cuerpo es proporcional a la fuerza neta aplicada e inversamente proporcional a su masa. En palabras de Newton: “*Mutationem motus proportionalem esse vi motrici impressæ, & fieri secundum lineam rectam qua vis illa imprimitur.*” [8].

Por tanto, la relación fundamental que rige el movimiento es:

$$\vec{F} = m \frac{d\vec{v}}{dt} = m \frac{d^2\vec{r}}{dt^2} = m\vec{a}, \quad (1)$$

2. Fundamentos de Dinámica Molecular

donde:

- \vec{F} representa la fuerza aplicada sobre el cuerpo,
- m es la masa del cuerpo,
- \vec{v} es la velocidad del cuerpo,
- \vec{r} es la posición del cuerpo,
- \vec{a} es la aceleración resultante del cuerpo.

Tercera Ley de Newton

La Tercera Ley de Newton, o Principio de Acción y Reacción, establece que si un cuerpo ejerce una fuerza sobre otro, entonces el segundo cuerpo ejerce una fuerza de igual magnitud, pero dirección opuesta, sobre el primero:

“Actioni contrariam semper & æqualem esse reactionem: sive corporum duorum actiones in se mutuo semper esse æquales & in partes contrarias dirigi.” [8]

Es decir, sea \vec{F}_{ij} la fuerza ejercida por un cuerpo i sobre un cuerpo j , y sea \vec{F}_{ji} la fuerza ejercida por el cuerpo j sobre el cuerpo i , entonces

$$\vec{F}_{ij} = -\vec{F}_{ji}. \quad (2)$$

2.1.2. Principio de conservación de la energía

El principio de conservación de la energía establece que, en un sistema cerrado y aislado, la energía no puede crearse ni destruirse, solo transformarse entre sus distintas formas [9]. En el contexto de la MD, este principio es fundamental para comprender cómo se comportan los sistemas a lo largo del tiempo.

En particular, este principio se manifiesta claramente en simulaciones que se realizan bajo las condiciones del colectivo¹ microcanónico (NVE) [10]. En este tipo de simulaciones, el número de partículas (N), el volumen (V) y la energía total (E) permanecen constantes durante toda la simulación, ya que el sistema está completamente aislado del entorno.

Sin embargo, en ocasiones resulta necesario simular condiciones más cercanas a la realidad, lo que requiere introducir mecanismos de control externos, que regulen la temperatura o la presión, por ejemplo. Con este fin, se recurre a los siguientes acomplamientos externos, que permiten la entrada o salida de energía del sistema:

- Termostato: regula la temperatura (T), permitiendo realizar simulaciones en el colectivo canónico (NVT), donde se mantiene constante la temperatura media del sistema [11, 12].
- Barostato: regula la presión (P). Combinado con un termostato, permite simulaciones en el colectivo isóbaro-isoterma (NPT), en el que se conservan tanto la presión media como la temperatura [13, 14].

¹En mecánica estadística una colectividad representa todas las posibles configuraciones microscópicas de un sistema bajo determinadas condiciones externas.

En estos casos, como se ha comentado, el sistema puede intercambiar energía con el exterior, por lo que la energía interna total deja de ser constante. Por tanto, el balance energético general se describe mediante la expresión:

$$U_T = U_i + W + Q,$$

donde:

- U_T es la energía interna total del sistema,
- U_i es la energía interna inicial del sistema,
- W es el trabajo realizado por o sobre el sistema,
- Q es el calor añadido o eliminado del sistema.

En la [Tabla 2.1](#) se muestra un resumen de los colectivos estadísticos más empleados en dinámica molecular, junto con las variables que permanecen constantes, los mecanismos de control necesarios, y algunos ejemplos típicos de aplicación.

Colectivo	Variables constantes	Control externo	Ejemplos de uso
NVE	N, V, E	Ninguno	Análisis energético puro, validación de integradores, estudios teóricos sin influencia externa [15]
NVT	N, V, T	Termostato	Procesos biológicos a temperatura constante, simulaciones de proteínas, análisis estructural en equilibrio térmico [11]
NPT	N, P, T	Termostato y barostato	Estudio de fases (cristalización, fusión), compresión, materiales a presión ambiente, simulaciones biomoleculares con entorno acuoso [14]

Tabla 2.1.: Colectividades estadísticas comunes en dinámica molecular.

Por último, es importante señalar que la elección del colectivo depende del tipo de proceso que se desea simular, de las propiedades físicas que se quieren medir y de la disponibilidad de datos experimentales con los que contrastar los resultados. En la práctica, los colectivos NVT y NPT son los más utilizados, ya que permiten replicar con mayor realismo las condiciones experimentales habituales.

2.1.3. Introducción a las ecuaciones de movimiento de Newton

Las ecuaciones de movimiento constituyen el núcleo matemático de la dinámica molecular, ya que, a partir de ellas, se obtiene la evolución de las posiciones y velocidades de las partículas a lo largo del tiempo bajo la acción de fuerzas. Dichas ecuaciones derivan directamente de la Segunda Ley de Newton (1).

En la práctica, debido a la complejidad que supone resolver estas ecuaciones de forma analítica para sistemas que involucran un gran número de partículas, se recurre a métodos

2. *Fundamentos de Dinámica Molecular*

numéricos, que permiten alcanzar soluciones aproximadas mediante la discretización temporal. Estos métodos, fundamentales para el desarrollo de simulaciones computacionales en dinámica molecular, se abordan en detalle en el [Capítulo 4](#).

3. Modelización en Dinámica Molecular

La modelización constituye una etapa fundamental en cualquier simulación de dinámica molecular, ya que determina el grado de fidelidad con el que el sistema físico real será representado computacionalmente. En esta sección se abordan los principales componentes que conforman la modelización: los modelos de interacción, la representación de los sistemas moleculares y las condiciones de contorno.

3.1. Modelos de interacción

Los sistemas están compuestos por partículas (átomos, iones o moléculas), que interactúan entre sí mediante funciones matemáticas llamadas potenciales de interacción, las cuales están diseñadas para aproximar fuerzas físicas entre las partículas. Estos potenciales describen cómo varía la energía potencial en función de la distancia entre las partículas. Las fuerzas se obtienen a partir de dichos potenciales mediante la siguiente expresión matemática:

$$\vec{F}_{ij} = -\nabla V(r_{ij}), \quad (3)$$

donde:

- \vec{F}_{ij} representa la fuerza ejercida por un cuerpo i sobre un cuerpo j ,
- $V(r_{ij})$ es el valor del potencial de interacción según la distancia entre los cuerpos i y j .

3.1.1. Campos de fuerza

Un campo de fuerza agrupa los distintos potenciales de interacción presentes en un sistema con el objetivo de describir todas las fuerzas internas y externas del mismo. Estas fuerzas se dividen, en general, en dos categorías: interacciones de corto alcance e interacciones de largo alcance.

Lennard-Jones

Las fuerzas de Van der Waals [16] son aquellas que se producen entre átomos y moléculas. Tienen carácter atractivo y repulsivo. Las moléculas y átomos se atraen hasta cierta distancia, pero si se acercan demasiado, se repelen. El potencial más comúnmente usado para modelar dichas interacciones es el potencial de Lennard-Jones. Este potencial capta dos contribuciones principales: una repulsiva a distancias muy cortas y otra atractiva a distancias intermedias. Su forma típica es:

$$V_{LJ}(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right], \quad (4)$$

donde:

- ϵ representa la profundidad del pozo de potencial,

3. Modelización en Dinámica Molecular

- σ representa el diámetro efectivo ($V_{LJ}(\sigma) = 0$),
- r es la distancia entre partículas.

Se denomina profundidad del pozo de potencial al valor ϵ , pues representa la energía mínima del potencial, donde la atracción entre partículas es máxima. Con el objetivo de calcular a qué distancia interatómica se alcanza dicha energía mínima, se iguala la derivada del potencial a cero:

$$\begin{aligned}\frac{dV_{LJ}(r)}{dr} = 0 &\iff 4\epsilon \left[-12 \cdot \frac{\sigma^{12}}{r^{13}} + 6 \cdot \frac{\sigma^6}{r^7} \right] = 0 \\ &\iff -12 \cdot \frac{\sigma^{12}}{r^{13}} + 6 \cdot \frac{\sigma^6}{r^7} = 0.\end{aligned}$$

Despejando r , se obtiene:

$$r = 2^{1/6}\sigma = r_{min}.$$

Por otro lado, se verifica que:

$$V_{LJ}(r) = 0 \iff \left(\frac{\sigma}{r}\right)^{12} = \left(\frac{\sigma}{r}\right)^6 \iff r = \sigma.$$

En la **Figura 3** se pueden observar estos parámetros, representados gráficamente sobre el potencial de Lennard-Jones.

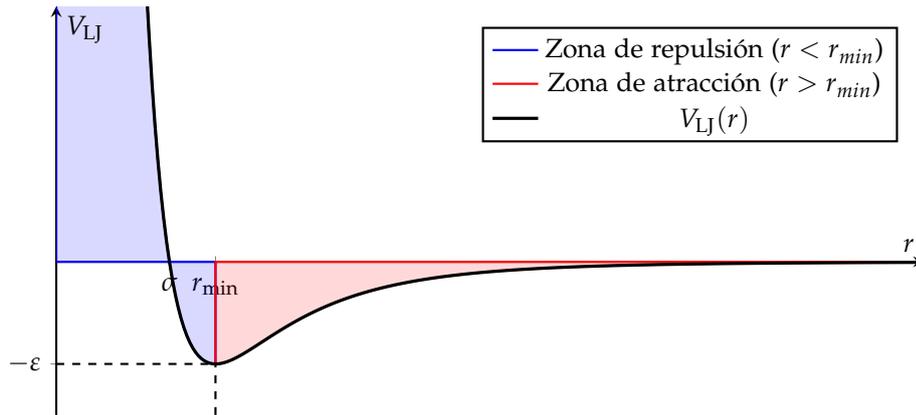


Figura 3.: Representación del potencial de Lennard-Jones.

Potenciales electrostáticos

Las interacciones electrostáticas se modelan mediante la ley de Coulomb, que define la fuerza entre dos cargas puntuales mediante la siguiente ecuación:

$$V_{\text{Coulomb}}(r) = \frac{Q_1 Q_2}{4\pi\epsilon_0\epsilon_r r'}$$

donde:

- Q_1 y Q_2 representan las cargas,

- ϵ_0 es la permitividad del vacío ($\approx 8.854 \cdot 10^{-12} \frac{C^2}{N \cdot m^2}$),
- ϵ_r representa la permitividad relativa del medio,
- r es la distancia entre cargas.

En muchas simulaciones de dinámica molecular se asume que las interacciones se producen en el vacío, tomando $\epsilon_r = 1$.

Estas interacciones, al tener un alcance teóricamente infinito (no se anulan completamente a ninguna distancia finita), requieren tratamientos computacionales especializados, como el método de Ewald [15] o la técnica de mallas de partículas (PME, *Particle-Mesh Ewald*) [17], que permiten mantener la eficiencia sin comprometer la precisión.

Campos de fuerza biomoleculares

Uno de los enfoques más utilizados para representar la energía potencial en sistemas biológicos es el uso de funciones empíricas definidas por campos de fuerza como AMBER [18] o CHARMM [19]. En estos modelos, además de las interacciones entre moléculas, se modelan también las interacciones entre átomos dentro de una misma molécula (ver Sección 5.4). Esto es especialmente necesario en moléculas con muchos grados de libertad internos. La forma general del potencial total viene dada por:

$$\begin{aligned}
 V_{\text{total}} &= V_{\text{enlace}} + V_{\text{ángulo}} + V_{\text{torsión}} + V_{\text{LJ}} + V_{\text{Coulomb}} = \\
 &= \sum_{\text{enlaces}} k_b (r - r_0)^2 + \sum_{\text{ángulos}} k_\theta (\theta - \theta_0)^2 + \sum_{\text{torsiones}} \frac{V_n}{2} [1 + \cos(n\phi - \gamma)] + \\
 &+ \sum_{i < j} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \sum_{i < j} \frac{Q_i Q_j}{4\pi\epsilon_0\epsilon_r r_{ij}},
 \end{aligned}$$

donde, además de los parámetros ya presentados, se tienen:

- k_b : representa la constante de fuerza del enlace,
- r : es la distancia actual entre los átomos considerados,
- r_0 : es la distancia de equilibrio del enlace,
- k_θ : representa la constante de fuerza angular,
- θ : es el ángulo actual entre los átomos considerados,
- θ_0 : es el ángulo de equilibrio,
- V_n : representa la barrera de torsión,
- n : es la periodicidad de la función,
- ϕ : es el ángulo de torsión actual entre los átomos considerados,
- γ : representa la fase de desplazamiento.

3. Modelización en Dinámica Molecular

El ajuste de las constantes presentadas puede plantearse como un problema de optimización, donde se busca minimizar la diferencia entre los valores obtenidos en la simulación y los de referencia. El proceso de parametrización se basa en una combinación de datos experimentales y cálculos de estructura electrónica, garantizando un equilibrio entre precisión física y viabilidad computacional. De esta forma, constituyen uno de los pilares fundamentales para asegurar la fiabilidad de la simulación.

Este tipo de representación permite modelar con gran precisión tanto las interacciones internas de una molécula (enlaces covalentes, ángulos de enlace y torsiones), representadas en la **Figura 4**, como las interacciones no enlazantes (fuerzas de Van der Waals y electrostáticas) entre diferentes moléculas o partes de la misma.

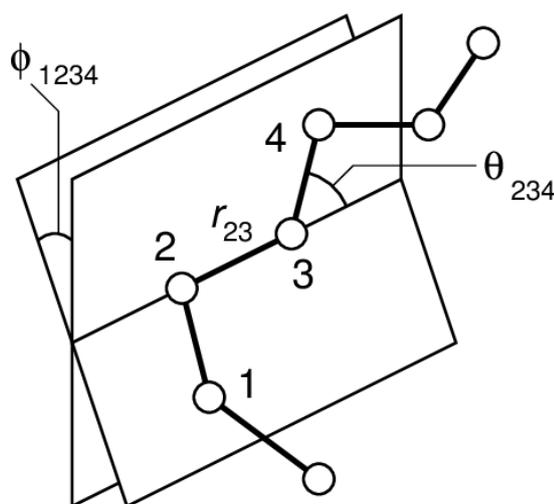


Figura 4.: Representación geométrica de una molécula en cadena simple, que ilustra la definición de la distancia interatómica r_{23} , el ángulo de flexión o ángulo de enlace θ_{234} , y el ángulo de torsión ϕ_{1234} [2].

Como ejemplo ilustrativo, en la práctica existen distintos modelos para representar, por ejemplo, una molécula de agua. La elección entre estos modelos depende del nivel de precisión y coste computacional que se desee.

En primer lugar, se tiene el modelo TIP3P, donde la distancia de enlace O-H es de $r_{OH} = 0.9572 \text{ \AA}$, el ángulo H-O-H es de $\theta = 104.52^\circ$, y se utilizan interacciones de tipo Lennard-Jones sobre el oxígeno junto con interacciones electrostáticas entre los tres átomos [20].

Por otro lado, el modelo TIP4P mejora la representación del agua desplazando la carga negativa desde el átomo de oxígeno hacia un punto virtual sin masa (denominado *massless site* o sitio M), que se coloca a una distancia de 0.15 \AA del átomo de oxígeno sobre el bisector del ángulo H-O-H. Esta modificación permite obtener mejores resultados a la hora de predecir propiedades físicas como la densidad del agua o su calor de vaporización, pero el coste computacional aumenta [20].

Finalmente, el modelo TIP5P va un paso más allá incorporando dos puntos virtuales adicionales que representan los pares de electrones no enlazantes del oxígeno, que se sitúan

a 0.7 \AA del oxígeno, con un ángulo O-L-O de 109.47° (ver Figura 5). Esto da lugar a una geometría más cercana a la real que mejora la descripción del comportamiento del agua en estado líquido, permitiendo obtener estimaciones más precisas de sus propiedades térmicas y estructurales anómalas. Sin embargo, el coste computacional aumenta aún más [20].

Los modelos planteados son rígidos, es decir, que las distancias de enlace y los ángulos se mantienen constantes durante la simulación. Como consecuencia, no se incluyen términos de torsión en el cálculo del potencial total. Esta simplificación no solo reduce el coste computacional, sino que también ha demostrado ser suficiente para reproducir propiedades estructurales y termodinámicas del agua cuando los parámetros están correctamente ajustados. Sin embargo, en moléculas más complejas, como proteínas, sí se incorpora el término ϕ [20].

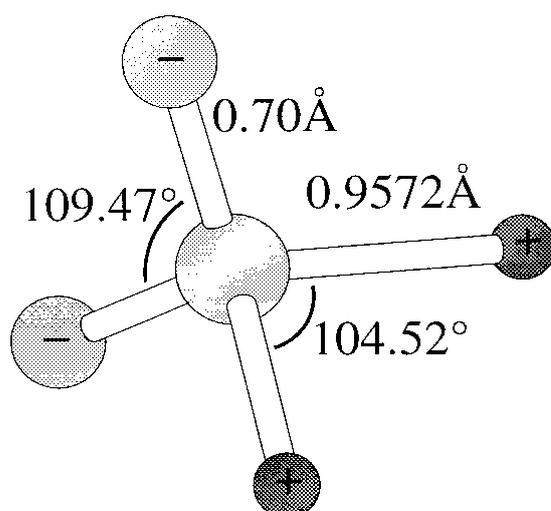


Figura 5.: Representación de una molécula de agua según el modelo TIP5P [20].

3.2. Representación de sistemas moleculares

3.2.1. Modelos atomísticos vs coarse-graining

El nivel de detalle con el que se representa un sistema molecular depende del objetivo de la simulación y de los recursos computacionales a disposición. Existen principalmente dos enfoques:

Modelos atomísticos

En primer lugar, los modelos atomísticos representan cada átomo de forma explícita, incluyendo su masa, carga, tipo de enlace y posición. Esta aproximación es la más común y es esencial en estudios en los que se requiere precisión en los detalles estructurales y dinámicos, como reacciones químicas o interacciones específicas entre proteínas y ligandos. Aunque proporcionan una gran precisión, su coste computacional es elevado, ya que el número de grados de libertad crece linealmente con el número de átomos.

Coarse-graining

Por otro lado, los modelos *coarse-grained* (CG) simplifican el sistema representando grupos de átomos como una única partícula efectiva. De esta forma, se reduce considerablemente el número de grados de libertad del sistema, permitiendo así estudiar sistemas más grandes y durante escalas temporales mucho más largas, haciendo posible el estudio de procesos como el plegamiento de proteínas. A pesar de perder resolución, este tipo de modelización es adecuado cuando se busca un comportamiento general en lugar de precisión local, ya que su coste computacional es menor, introduciendo así un *trade-off* entre eficiencia computacional y precisión.

Este tipo de modelización requiere asegurar que el modelo conserva las propiedades macroscópicas relevantes del sistema original, representando así una frontera entre la física estadística, el modelado matemático y la simulación computacional.

Para llevar a cabo ambas representaciones, es habitual hacer uso de herramientas de pre-procesado como *tLeap* (en AMBER) o *psfgen* (en NAMD [21]), que permiten construir y modificar estructuras moleculares, definir la topología del sistema, generar las coordenadas iniciales y asignar los parámetros del campo de fuerza. Estas herramientas preparan los archivos de entrada necesarios para ejecutar simulaciones de dinámica molecular. Puede encontrarse más información sobre sus usos en [22] y [23], respectivamente.

3.2.2. Cajas de simulación

Tras elegir los modelos de interacción y el nivel de resolución, se debe decidir cómo organizar el sistema dentro del espacio simulado. Esta representación se ve reflejada tanto en la validez física de la simulación como en la viabilidad computacional.

Para ello, las moléculas que conforman el sistema se encierran en una caja de simulación, que se comporta como un contenedor tridimensional. Dicha caja puede adoptar distintas geometrías en función de la naturaleza del sistema y de las condiciones que se deseen reproducir.

Por otro lado, además de las moléculas de interés, se suelen incluir un número determinado de moléculas de solvente, como puede ser el agua, con el fin de imitar el entorno biológico o experimental.

3.2.3. Condiciones de contorno

Las condiciones de contorno definen cómo se comportan las moléculas cuando alcanzan los límites del sistema simulado. Para evitar efectos de borde no físicos, como acumulación en las paredes o vacío exterior, se imponen ciertas condiciones de contorno:

Condiciones periódicas (*Periodic Boundary Conditions, PBC*)

La caja se replica infinitamente en todas las direcciones, de modo que cuando una partícula sale por un lado, regresa por el lado opuesto (ver [Figura 6](#)), permitiendo así simular un entorno infinito a partir de un número finito de partículas.

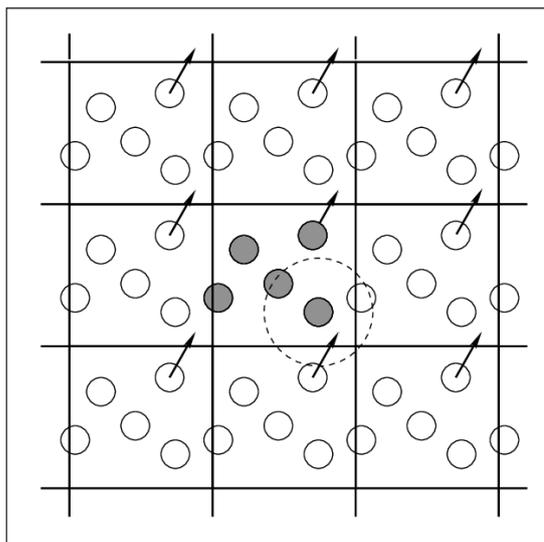


Figura 6.: PBC. A medida que una partícula sale de la caja de simulación, una imagen de la partícula entra para reemplazarla [2].

Condiciones no periódicas

En este caso, se asume que el sistema está aislado y que los bordes representan límites físicos reales. Por tanto, las partículas pueden escapar del sistema (condiciones de contorno abiertas) o rebotar en los bordes (reflectantes). Estas condiciones se emplean en sistemas donde la periodicidad no es deseable, como simulaciones de interfases, superficies o sistemas donde el entorno exterior no se repite.

3.2.4. Métodos de truncamiento

En simulaciones de MD, es fundamental tratar eficientemente las interacciones entre partículas con el fin de reducir el coste computacional. Dado que las interacciones de largo alcance decrecen con la distancia, se introduce un radio de corte r_c tal que $V(r_{ij}) = 0$ si $r_{ij} > r_c$, para dos átomos i, j . Esto permite al programa omitir el cálculo de fuerzas entre átomos alejados más allá de esa distancia, ahorrando una cantidad significativa de cálculos.

No obstante, determinar en cada paso temporal qué partículas se encuentran dentro de ese radio para todas las demás puede seguir siendo muy costoso. En efecto, en un sistema compuesto por N partículas, se deben comprobar un total de $\frac{1}{2}N(N-1)$ pares posibles. Para mejorar esta eficiencia, Verlet introdujo las llamadas listas de vecinos (o *Verlet lists*). Estas listas mantienen un registro de las partículas que están dentro de un radio extendido $r_v > r_c$, llamado radio de vecindad, alrededor de cada partícula (ver [Figura 11](#) para una representación del entorno de una partícula).

En el primer paso de la simulación, se construye la lista de vecinos de cada átomo, y en los siguientes pasos solo se calculan las fuerzas de las parejas que figuran en dicha lista. Cada cierto número de pasos, antes de que una pareja inicialmente fuera del radio de vecindad

3. Modelización en Dinámica Molecular

pueda entrar en el radio de corte, la lista debe ser reconstruida para mantener la precisión del modelo.

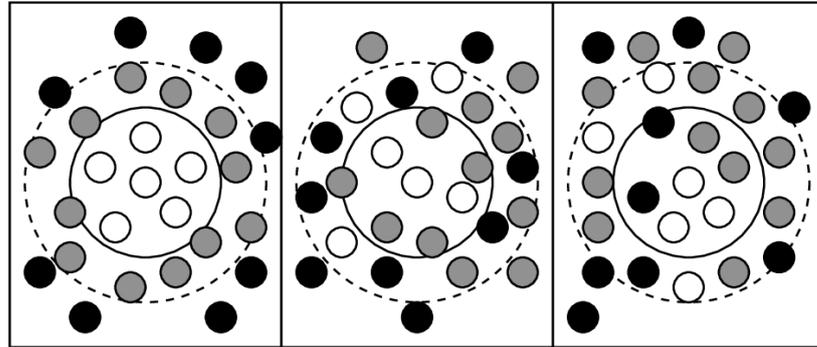


Figura 7.: Distintas etapas de la formación de la lista de Verlet. Se puede observar el radio de corte (círculo sólido) y el radio de vecindad (círculo discontinuo). La lista ha de ser reconstruida antes de que las partículas negras (inicialmente fuera de la lista) entren en el radio de corte [2].

4. Algoritmos en Dinámica Molecular

En dinámica molecular, como ya se comentó en el [Capítulo 2](#), el comportamiento de las partículas que componen un sistema se determina mediante la resolución de un conjunto de ecuaciones diferenciales ordinarias que derivan de (1). Estas ecuaciones describen la evolución de las posiciones y velocidades de dichas partículas a lo largo del tiempo. Sin embargo, ante la imposibilidad de resolverlas de forma analítica debido a la complejidad de los sistemas estudiados y el gran número de grados de libertad implicados, se recurre a algoritmos numéricos que permiten calcular la evolución de las partículas paso a paso, obteniendo así una simulación del sistema.

4.1. Integración de ecuaciones de movimiento

Tomando como punto de partida las ecuaciones (1) y (3), la base matemática de toda simulación de dinámica molecular viene dada por la integración de las siguientes ecuaciones diferenciales:

$$m_i \frac{d^2 \vec{r}_i}{dt^2} = \vec{F}_i = -\nabla_{\vec{r}_i} V(\vec{r}_1, \dots, \vec{r}_N).$$

La elección de un método de integración adecuado resulta fundamental, ya que debe garantizar la conservación de la energía, estabilidad numérica a largo plazo y un coste computacional razonable.

Los métodos numéricos que se presentarán discretizan el tiempo en intervalos de tamaño Δt , permitiendo así obtener una evolución aproximada de las trayectorias. Por ello, para garantizar la estabilidad numérica a largo plazo, es necesario que el método sea capaz de controlar los errores de redondeo y truncamiento a lo largo de la simulación. Si el tamaño del paso temporal Δt es demasiado grande, el error acumulado puede aumentar, provocando inestabilidades, como explosiones de energía o trayectorias físicamente irreales.

Por tanto, la elección del parámetro Δt es crucial, ya que debe ser lo suficientemente pequeño como para capturar las dinámicas rápidas del sistema, como las vibraciones intramoleculares, pero lo bastante grande como para optimizar el tiempo de cálculo, pues a menor valor de Δt , mayor coste computacional.

En general, se asume que las aceleraciones (y, por tanto, las fuerzas) se mantienen aproximadamente constantes durante cada paso temporal Δt , lo que permite que los integradores numéricos utilicen la fuerza calculada al inicio del paso para predecir la evolución del sistema [24]. De esta forma, cuanto más ligeros sean los átomos, más rápidas serán sus oscilaciones, y menor deberá ser el paso Δt para evitar errores significativos.

En particular, el límite superior para Δt está determinado por las frecuencias más altas del sistema, normalmente asociadas a vibraciones intramoleculares rápidas. Una práctica común

4. Algoritmos en Dinámica Molecular

consiste en seleccionar Δt al menos 10 veces menor que el período de oscilación más corto del sistema [24]. Por ejemplo, para una vibración de período $\tau \approx 10^{-14}$ s, se recomienda tomar $\Delta t \leq 10^{-15}$ s = 1 femtosegundo (fs). Superar este umbral puede provocar un crecimiento exponencial de los errores numéricos.

A pesar de ello, existe una cierta tensión entre la necesidad de utilizar pasos de tiempo pequeños para asegurar la precisión numérica y el deseo de alcanzar escalas temporales largas, típicas de muchos procesos microscópicos. Una simulación de 1 microsegundo con $\Delta t = 1$ fs requiere del orden de 10^6 ciclos de integración, lo que implica un coste computacional considerable, ya que en cada paso es necesario recalculer todas las fuerzas. Por ello, se busca elegir el mayor valor de Δt posible que mantenga la estabilidad numérica y la conservación de la energía [24].

Además, aunque se utilicen representaciones de alta precisión (por ejemplo, doble precisión en punto flotante), los errores de redondeo se acumulan gradualmente a lo largo de la trayectoria, especialmente si se ejecutan simulaciones con millones de pasos. Por ello, la elección adecuada del integrador y del tamaño del paso de integración es fundamental para garantizar la estabilidad y precisión global de la simulación a largo plazo.

Propiedades geométricas: métodos simplécticos

La dinámica molecular se enmarca dentro del contexto de los sistemas hamiltonianos [25], los cuales describen la evolución temporal de un sistema mediante ecuaciones que dependen de las coordenadas y los momentos generalizados. Estas ecuaciones derivan de un Hamiltoniano $H(\vec{r}, \vec{p})$, que representa la energía total del sistema:

$$H(\vec{r}, \vec{p}) = \sum_{i=1}^N \frac{\|\vec{p}_i\|^2}{2m_i} + V(\vec{r}_1, \dots, \vec{r}_N),$$

y cuya evolución temporal se determina por:

$$\begin{cases} \frac{d\vec{r}_i}{dt} = \frac{\partial H}{\partial \vec{p}_i} = \frac{\vec{p}_i}{m_i} = \vec{v}_i, \\ \frac{d\vec{p}_i}{dt} = -\frac{\partial H}{\partial \vec{r}_i} = \vec{F}_i, \end{cases}$$

donde:

- $H(\vec{r}, \vec{p})$ es la energía total del sistema,
- \vec{r} representa las coordenadas generalizadas²,
- $\vec{p} = m \cdot \vec{v}$ son los momentos conjugados.

²En este trabajo, por simplicidad, se considera únicamente el movimiento traslacional de las partículas, es decir, \vec{r}_i corresponde a las coordenadas espaciales del centro de masa de la partícula i . Sin embargo, en dinámica molecular también pueden intervenir grados de libertad rotacionales y vibracionales, especialmente relevantes en moléculas poliatómicas o sistemas rígidos.

Una propiedad fundamental de estas ecuaciones es su naturaleza simpléctica, lo que implica que las trayectorias del sistema conservan el volumen en el espacio de fases, según el teorema de Liouville [25]. Es decir, aunque las configuraciones del sistema pueden evolucionar deformando la forma de un volumen en el espacio de fases, el volumen total ocupado permanece constante [24]. Esta característica tiene importantes consecuencias en el ámbito numérico. Así, un algoritmo de integración de las ecuaciones de Hamilton debe cumplir:

- Aunque se produzcan pequeñas oscilaciones en la energía o el momento, los errores no se acumulen sistemáticamente, permitiendo conservar estos invariantes a largo plazo.
- Garantizar la estabilidad estructural del sistema en simulaciones prolongadas.

Los integradores simplécticos están diseñados para respetar estas propiedades geométricas, imitando el comportamiento del flujo hamiltoniano exacto, haciéndolos especialmente adecuados para simular sistemas conservativos y estudiar propiedades de equilibrio. De hecho, se ha demostrado que estos métodos pueden interpretarse como soluciones exactas de un sistema hamiltoniano modificado, ligeramente perturbado, cuya forma depende del paso de integración utilizado [26]. Esta característica explica su capacidad para conservar invariantes del sistema, como la energía o el volumen de fase, a lo largo de millones de pasos.

Para ilustrar las diferencias entre integradores simplécticos y no simplécticos, a continuación se comparan los resultados obtenidos al simular un oscilador armónico utilizando los métodos de Euler [27] (ver Figura 8) y Verlet (ver Figura 9). En todos los casos, se considera una partícula sometida a una fuerza lineal del tipo $\vec{F} = -k\vec{r}$, donde k es la constante de elasticidad.

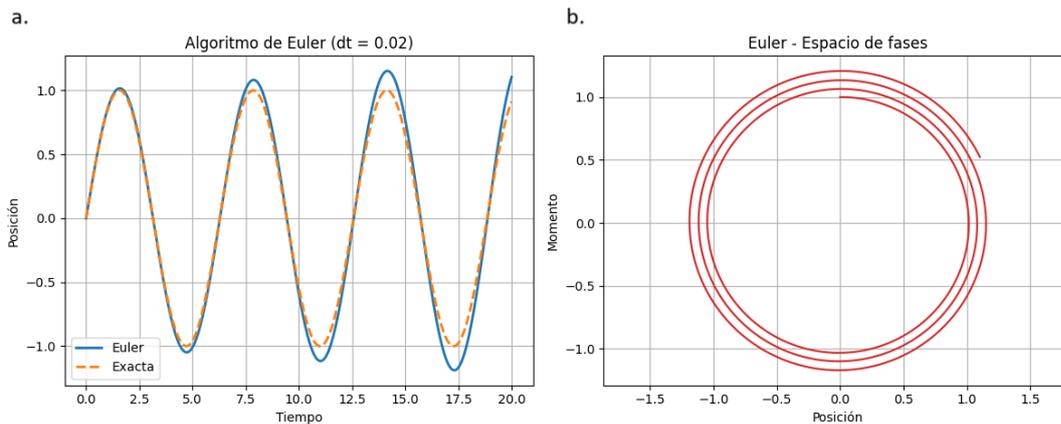


Figura 8.: Simulación con el método de Euler (no simpléctico). (a) Evolución temporal de la posición; (b) Trayectoria en el espacio de fases. Se observa una trayectoria espiral divergente, que indica una ganancia artificial de energía.

4. Algoritmos en Dinámica Molecular

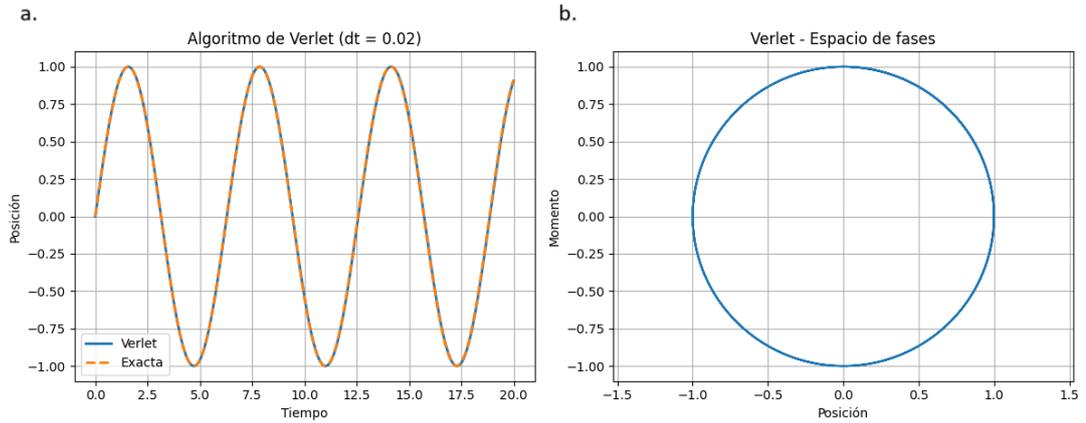


Figura 9.: Simulación con el método de Verlet (simpléctico). (a) Evolución temporal de la posición; (b) Trayectoria en el espacio de fases. La energía se conserva a largo plazo y el sistema describe una órbita cerrada.

Sin embargo, a pesar de sus ventajas, los integradores simplécticos no son incondicionalmente estables, pues, como se ha comentado anteriormente, existe un límite superior para el tamaño del paso temporal Δt más allá del cual las simulaciones pueden volverse inestables o generar trayectorias inconsistentes con la física del sistema. Esto puede observarse claramente en la [Figura 10](#), al analizar el comportamiento del algoritmo de Verlet frente a diferentes valores de Δt . Para pasos suficientemente pequeños, el método conserva su estabilidad y mantiene las propiedades geométricas del sistema. Sin embargo, al superar el umbral crítico, el integrador pierde estabilidad y deja de reflejar adecuadamente la dinámica del sistema.

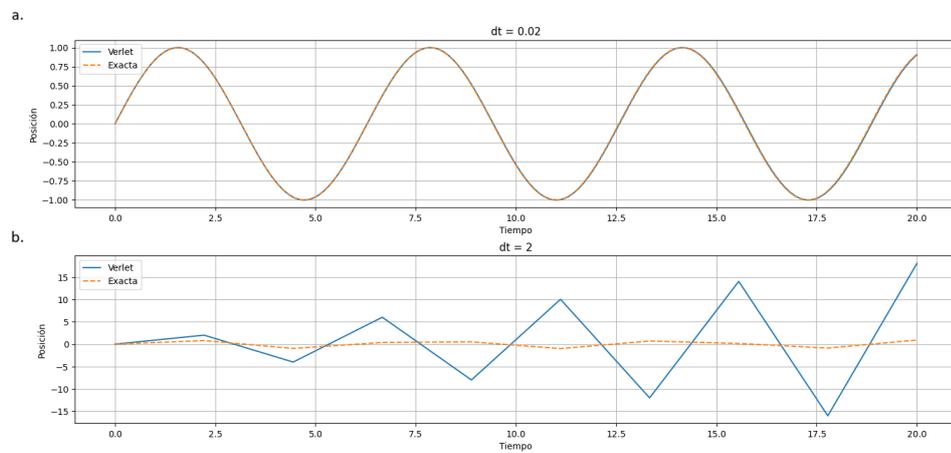


Figura 10.: Evolución temporal de la posición con el algoritmo de Verlet para dos tamaños de paso temporal. (a) $\Delta t = 0.02$, donde el algoritmo es estable; (b) $\Delta t = 2$, donde es inestable.

Por tanto, aunque los integradores simplécticos tienen excelentes propiedades de conservación de invariantes geométricos, es fundamental elegir adecuadamente el paso de tiempo

para garantizar su estabilidad numérica.

Por otro lado, esta propiedad está estrechamente relacionada con la reversibilidad temporal, pues un integrador simpléctico reversible es capaz de reproducir la trayectoria inversa exacta al invertir los momentos (en ausencia de errores de redondeo). Este comportamiento refleja fielmente la simetría del flujo hamiltoniano, y contribuye a la fiabilidad del método [26].

A continuación se presentan algunos de los algoritmos simplécticos reversibles más utilizados en simulaciones de MD, debido a su equilibrio entre eficiencia, precisión y simplicidad.

4.1.1. Algoritmo de Verlet

El algoritmo de Verlet constituye la base de muchos de los métodos más utilizados en dinámica molecular, destacando por su simplicidad y por sus excelentes propiedades de conservación de la energía [15]. Su formulación matemática se apoya en un desarrollo en serie de Taylor, mediante la cual se aproxima la posición de una partícula en los instantes $t + \Delta t$ y $t - \Delta t$, a partir de su valor en t . Esta simetría temporal lo convierte en un método reversible en el tiempo [26].

En primer lugar, se desarrolla $\vec{r}_i(t + \Delta t)$ en serie de Taylor alrededor de t :

$$\vec{r}_i(t + \Delta t) = \sum_{n=0}^{\infty} \frac{\vec{r}_i^{(n)}(t)}{n!} (\Delta t)^n = \vec{r}_i(t) + \vec{v}_i(t)\Delta t + \frac{1}{2}\vec{a}_i(t)\Delta t^2 + \frac{1}{6}\dot{\vec{a}}_i(t)\Delta t^3 + \mathcal{O}(\Delta t^4). \quad (5)$$

Análogamente, se obtiene el desarrollo hacia atrás:

$$\vec{r}_i(t - \Delta t) = \sum_{n=0}^{\infty} \frac{\vec{r}_i^{(n)}(t)}{n!} (-\Delta t)^n = \vec{r}_i(t) - \vec{v}_i(t)\Delta t + \frac{1}{2}\vec{a}_i(t)\Delta t^2 - \frac{1}{6}\dot{\vec{a}}_i(t)\Delta t^3 + \mathcal{O}(\Delta t^4). \quad (6)$$

Finalmente, combinando (5) y (6), se cancelan los términos impares y se obtiene la expresión del algoritmo de Verlet: s

$$\boxed{\vec{r}_i(t + \Delta t) = 2\vec{r}_i(t) - \vec{r}_i(t - \Delta t) + \frac{\Delta t^2}{m_i} F_i(t).} \quad (7)$$

La aproximación obtenida presenta un error local de orden $\mathcal{O}(\Delta t^4)$, luego, al tratarse de una ecuación diferencial de segundo orden, se tiene un error global de orden $\mathcal{O}(\Delta t^2)$. Por tanto, el algoritmo de Verlet se convierte en una herramienta muy precisa.

Además, el algoritmo de Verlet presenta una excelente estabilidad numérica. Al ser un integrador simpléctico, no conserva exactamente la energía total en cada paso, pero las fluctuaciones que introduce permanecen acotadas y no se acumulan de forma sistemática. Esta propiedad resulta especialmente importante en simulaciones realizadas bajo condiciones del conjunto microcanónico (NVE), donde la conservación de la energía es un criterio fundamental [28].

4. Algoritmos en Dinámica Molecular

Una de las limitaciones del algoritmo de Verlet es que no actualiza las velocidades de las partículas de forma explícita. No obstante, si se desea, estas se pueden estimar a partir de las posiciones en distintos pasos de tiempo, aprovechando la simetría del método. Para ello, restando las ecuaciones (5) y (6), se obtiene:

$$\vec{r}_i(t + \Delta t) - \vec{r}_i(t - \Delta t) = 2\vec{v}_i(t)\Delta t + \mathcal{O}(\Delta t^3).$$

Despejando la velocidad:

$$\vec{v}_i(t) = \frac{\vec{r}_i(t + \Delta t) - \vec{r}_i(t - \Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2). \quad (8)$$

Proporcionando así una estimación de las velocidades con un error local de orden $\mathcal{O}(\Delta t^2)$, lo cual resulta suficientemente preciso para la mayoría de aplicaciones prácticas, como el cálculo de energía cinética o temperatura en simulaciones bajo condiciones del colectivo microcanónico.

4.1.2. Velocity-Verlet

El algoritmo Velocity-Verlet es una de las variantes más populares del método de Verlet, ya que calcula de forma explícita, además de la posición, la velocidad de la partícula en cada paso de tiempo, lo que resulta muy útil para calcular propiedades como la energía cinética o la temperatura [15, 29].

Al igual que el algoritmo de Verlet clásico, este algoritmo también se basa en un desarrollo en serie de Taylor de las posiciones y velocidades, convirtiéndose así en un método reversible en el tiempo [26].

El desarrollo en serie de Taylor de la velocidad hacia adelante nos da:

$$\vec{v}_i(t + \Delta t) = \vec{v}_i(t) + \vec{a}_i(t)\Delta t + \mathcal{O}(\Delta t^2).$$

Sin embargo, dicha expresión solo utiliza la aceleración actual, ignorando la que la partícula alcanzará durante el desplazamiento. Por tanto, en busca de obtener una mayor precisión y simetría temporal, se calcula la velocidad en dos pasos: una mitad antes de actualizar la posición, y otra mitad después de calcular la nueva aceleración [30].

1. Primero se actualiza la velocidad a mitad de paso:

$$\vec{v}_i\left(t + \frac{\Delta t}{2}\right) = \vec{v}_i(t) + \frac{1}{2}\vec{a}_i(t)\Delta t + \mathcal{O}(\Delta t^2). \quad (9)$$

2. Seguidamente, tomando (9), se actualiza la posición completa:

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \vec{v}_i(t)\Delta t + \frac{1}{2}\vec{a}_i(t)\Delta t^2 + \mathcal{O}(\Delta t^3) = \vec{r}_i(t) + \vec{v}_i\left(t + \frac{\Delta t}{2}\right)\Delta t + \mathcal{O}(\Delta t^3).$$

3. Tras actualizar la posición, se recalcula la aceleración en el nuevo tiempo $t + \Delta t$ a partir de (1):

$$\vec{a}_i(t + \Delta t) = \frac{1}{m_i} \vec{F}_i(t + \Delta t).$$

4. Finalmente, se completa la otra mitad de la velocidad:

$$\vec{v}_i(t + \Delta t) = \vec{v}_i\left(t + \frac{\Delta t}{2}\right) + \frac{1}{2} \vec{a}_i(t + \Delta t) \Delta t + \mathcal{O}(\Delta t^3).$$

Obteniendo así las expresiones del algoritmo Velocity-Verlet:

$$\begin{aligned} \vec{v}_i\left(t + \frac{\Delta t}{2}\right) &= \vec{v}_i(t) + \frac{1}{2} \vec{a}_i(t) \Delta t, \\ \vec{r}_i(t + \Delta t) &= \vec{r}_i(t) + \vec{v}_i\left(t + \frac{\Delta t}{2}\right) \Delta t, \\ \vec{v}_i(t + \Delta t) &= \vec{v}_i\left(t + \frac{\Delta t}{2}\right) + \frac{1}{2} \vec{a}_i(t + \Delta t) \Delta t. \end{aligned}$$

Se obtiene un error local de orden $\mathcal{O}(\Delta t^3)$ tanto en el cálculo de la posición como de velocidad, luego, al tratarse de ecuaciones diferenciales de primer orden, presenta un error global de orden $\mathcal{O}(\Delta t^2)$ en ambos cálculos.

Además, al igual que el algoritmo de Verlet, el algoritmo Velocity-Verlet muestra una excelente estabilidad numérica para sistemas conservativos.

Por tanto, el algoritmo Velocity-Verlet representa un perfecto equilibrio entre precisión numérica, fidelidad física y eficiencia computacional, haciéndolo así uno de los algoritmos más empleados en simulaciones de dinámica molecular, destacando sobre todo en simulaciones biomoleculares [30].

4.1.3. Leap-Frog

El método Leap-Frog, o "salto de rana", debe su nombre al hecho de que las posiciones y velocidades se actualizan en pasos de tiempo intercalados. Aunque esta característica introduce cierta dificultad a la hora de calcular magnitudes como la energía cinética o la temperatura en tiempos enteros, su simpleza y eficiencia lo convierten en una opción adecuada para muchas simulaciones, especialmente en combinación con algunos controladores de temperatura [15, 29].

Para obtener la expresión de la velocidad, se parte del desarrollo en serie de Taylor centrada en t para los instantes $t \pm \frac{\Delta t}{2}$, convirtiéndose así en un método reversible en el tiempo [26]:

$$\begin{aligned} \vec{v}_i\left(t + \frac{\Delta t}{2}\right) &= \vec{v}_i(t) + \frac{1}{2} \vec{a}_i(t) \Delta t + \frac{1}{8} \dot{\vec{a}}_i(t) \Delta t^2 + \mathcal{O}(\Delta t^3), \\ \vec{v}_i\left(t - \frac{\Delta t}{2}\right) &= \vec{v}_i(t) - \frac{1}{2} \vec{a}_i(t) \Delta t + \frac{1}{8} \dot{\vec{a}}_i(t) \Delta t^2 + \mathcal{O}(\Delta t^3). \end{aligned}$$

4. Algoritmos en Dinámica Molecular

Restando ambas expresiones y despejando, se obtiene:

$$\vec{v}_i \left(t + \frac{\Delta t}{2} \right) = \vec{v}_i \left(t - \frac{\Delta t}{2} \right) + \vec{a}_i(t) \Delta t. \quad (10)$$

Una vez conocida la velocidad en el instante $t + \frac{\Delta t}{2}$, se actualiza la posición con:

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \vec{v}_i \left(t + \frac{\Delta t}{2} \right) \Delta t. \quad (11)$$

Las ecuaciones (10) y (11) constituyen el núcleo del algoritmo Leap-Frog, con un error local de orden $\mathcal{O}(\Delta t^3)$ tanto en posición como en velocidad.

En caso de que se desee obtener la velocidad en un paso de tiempo completo t , se puede aproximar mediante el promedio de las velocidades:

$$\vec{v}_i(t) = \frac{1}{2} \left[\vec{v}_i \left(t - \frac{\Delta t}{2} \right) + \vec{v}_i \left(t + \frac{\Delta t}{2} \right) \right].$$

Al igual que el algoritmo Velocity-Verlet, Leap-Frog presenta un error local de orden $\mathcal{O}(\Delta t^3)$ tanto en el cálculo de la posición como de velocidad, luego, al tratarse de ecuaciones diferenciales de primer orden, presenta un error global de orden $\mathcal{O}(\Delta t^2)$ en ambos cálculos.

Por otro lado, el algoritmo Leap-Frog, al ser simpléctico, también presenta una excelente estabilidad numérica para sistemas conservativos.

Finalmente, una ventaja destacable del algoritmo Leap-Frog es que proporciona una forma eficiente y estable de integrar las velocidades en pasos intermedios, lo cual puede resultar útil en combinación con ciertos termostatos que requieren acceso frecuente a las velocidades del sistema [28].

Conclusión

Los algoritmos de Verlet y sus variantes son ampliamente utilizados en MD debido a:

- Su derivación simple (por desarrollo de Taylor).
- Su reversibilidad temporal.
- Su estructura simpléctica, que, para un valor adecuado de Δt , garantiza estabilidad y conservación del volumen en el espacio de fases.
- Su bajo coste computacional, ya que solo requieren una evaluación de fuerzas por paso.

En la [Tabla 4.1](#) se puede ver un resumen de las propiedades de los algoritmos introducidos.

Algoritmo numérico	Magnitud	Error local de truncamiento	Error global de truncamiento	Reversible	Simpléctico
Verlet	\vec{r} \vec{v}	Δt^4 Δt^2	Δt^2 Δt^2	Sí	Sí
Velocity-Verlet	\vec{r} \vec{v}	Δt^3 Δt^3	Δt^2 Δt^2	Sí	Sí
Leap-Frog	\vec{r} \vec{v}	Δt^3 Δt^3	Δt^2 Δt^2	Sí	Sí

Tabla 4.1.: Propiedades del algoritmo de Verlet y de sus variantes.

4.2. Control de temperatura y presión

Las simulaciones de dinámica molecular no siempre se realizan bajo las condiciones del colectivo microcanónico (NVE), es decir, en condiciones de energía constante. En la práctica, con el fin de representar de forma realista las condiciones experimentales, suele ser necesario trabajar bajo condiciones del colectivo canónico (NVT), manteniendo constante la temperatura, o del isóbaro-isoterma (NPT), manteniendo constante tanto la presión como la temperatura. Para ello, se introducen mecanismos de control externos como termostatos y barostatos, que permiten acoplar el sistema a un baño térmico o barométrico, respectivamente [10, 15].

Desde un punto de vista físico-matemático, estos mecanismos modifican las ecuaciones de movimiento del sistema, ya sea mediante la introducción de términos adicionales o mediante un reescalado dinámico de las variables, en busca de establecer las condiciones deseadas.

La incorporación de termostatos y barostatos es necesaria porque, sin mecanismos de control, la temperatura y la presión del sistema pueden alejarse considerablemente de los valores deseados, ya sea por fluctuaciones estadísticas o por el propio proceso de inicialización. Por ejemplo, aplicando el teorema de equipartición de la energía para un sistema tridimensional con N partículas se tiene que:

$$\frac{1}{N} \sum_{i=1}^N \frac{1}{2} m_i \|\vec{v}_i\|^2 = \frac{3}{2} k_B T \Rightarrow T = \frac{1}{3Nk_B} \sum_{i=1}^N m_i \|\vec{v}_i\|^2 \quad (\text{K}), \quad (12)$$

donde:

- m_i es la masa de la partícula i ,
- \vec{v}_i es la velocidad de la partícula i ,
- k_B es la constante de Boltzmann ($\approx 1.38065 \times 10^{-23}$ J/K),
- T es la temperatura instantánea del sistema.

Por tanto, sin termostato, la temperatura del sistema puede derivar progresivamente debido a errores numéricos acumulados, especialmente en simulaciones largas. Esto puede provocar resultados físicamente incorrectos o inconsistentes con condiciones experimentales. Por ejemplo, si se desea modelar una proteína a 300 K pero no se regula la temperatura, la simulación puede desviarse de este valor, afectando a la estructura y la dinámica del sistema [12, 31].

4. Algoritmos en Dinámica Molecular

Análogamente, la presión instantánea del sistema se puede estimar mediante la ecuación virial de Clausius [32]:

$$P = \frac{Nk_B T}{V} + \frac{1}{3V} \sum_{i=1}^N \vec{r}_i \cdot \vec{F}_i \quad (\text{Pa}),$$

donde:

- V es el volumen de la celda de simulación,
- \vec{r}_i es la posición de la partícula i ,
- \vec{F}_i es la fuerza que actúa sobre ella.

Luego, sin un barostato que ajuste dinámicamente el volumen, la presión puede desviarse considerablemente del valor deseado, lo cual afecta a propiedades dependientes del volumen, como la densidad, estructura o energía libre del sistema. En muchos sistemas biológicos o de materiales blandos, las propiedades estructurales dependen de la presión ambiental. Si esta no se controla, se pierde la posibilidad de reproducir comportamientos físicos realistas [10].

4.2.1. Termostatos

Los termostatos permiten regular la temperatura del sistema durante la simulación, reproduciendo su interacción con un entorno térmico. Su uso es esencial en simulaciones bajo condiciones del colectivo canónico (NVT), donde la temperatura se mantiene constante, con el fin de asegurar un muestreo adecuado del espacio de fases [15, 24].

En este contexto, resulta deseable que las velocidades de las partículas muestren una distribución estadística coherente con la distribución de Maxwell-Boltzmann, la cual surge del tratamiento clásico del equilibrio térmico [15]. Esta distribución adopta la forma:

$$f(v) = 4\pi \left(\frac{m}{2\pi k_B T} \right)^{3/2} v^2 \exp \left(-\frac{mv^2}{2k_B T} \right),$$

donde $v = \|\vec{v}\|$. Esta expresión describe la probabilidad de encontrar una partícula con velocidad v en un sistema en equilibrio térmico.

Sin embargo, la aplicabilidad de esta distribución no es universal. Su validez se restringe a sistemas clásicos, en los que los efectos cuánticos pueden despreciarse, y que se encuentran en equilibrio térmico. Además, presupone que las velocidades de las partículas no están fuertemente correlacionadas entre sí, lo que se cumple típicamente en gases ideales o fluidos diluidos. En sistemas densos como líquidos o sólidos, donde las interacciones entre partículas son significativas y duraderas, pueden aparecer desviaciones locales respecto a la distribución de Maxwell-Boltzmann. No obstante, en muchas simulaciones se observa que, en equilibrio termodinámico, la distribución global de velocidades se aproxima razonablemente a la forma predicha [33, 24].

Por tanto, un termostato eficaz no debe limitarse a mantener constante la energía cinética media del sistema. En condiciones ideales, debe inducir una dinámica compatible con la ergo-

dicidad³ del mismo, permitiendo que, tras un número suficiente de pasos de simulación, las velocidades de las partículas sigan estadísticamente la distribución de Maxwell-Boltzmann. Esta propiedad es clave para asegurar que el sistema explore correctamente el espacio de fases del colectivo estadístico correspondiente y, en consecuencia, que las propiedades termodinámicas obtenidas sean físicamente representativas [15, 33].

En la práctica, sin embargo, no se alcanza un equilibrio térmico exacto, sino una aproximación al mismo. Debido a la finitud del sistema, a los errores numéricos asociados al método de integración y a la duración limitada de las simulaciones, lo que se observa es que el sistema tiende hacia un estado estacionario en el que la distribución global de velocidades se aproxima razonablemente a la forma predicha por Maxwell-Boltzmann [24, 33].

Comprobar que la distribución de velocidades generada en la simulación coincide con la de Maxwell-Boltzmann constituye, por tanto, un criterio empírico útil para evaluar si el sistema ha alcanzado un estado compatible con el equilibrio térmico. Si se observan desviaciones significativas respecto a dicha distribución, esto puede indicar problemas de acoplamiento térmico, un tiempo de relajación insuficiente o errores sistemáticos en la integración de las ecuaciones de movimiento [24]. Por tanto, el uso adecuado de termostatos, junto con la validación estadística de los resultados obtenidos, es fundamental para garantizar la fiabilidad de las simulaciones.

Existen distintos algoritmos:

Termostato de Berendsen

El termostato de Berendsen realiza un reescalado suave de las velocidades con el fin de ajustar la temperatura del sistema hacia un valor objetivo [31]. Matemáticamente, las velocidades se actualizan de la siguiente forma, asegurando que se cumple el teorema de equipartición:

$$\vec{v}_i \rightarrow \lambda \vec{v}_i, \quad \lambda = \sqrt{1 + \frac{\Delta t}{\tau_T} \left(\frac{T_0}{T} - 1 \right)},$$

donde:

- T representa la temperatura instantánea del sistema,
- T_0 es la temperatura objetivo,
- τ_T es la constante de relajación, tiempo que tarda el sistema en alcanzar el nuevo estado de equilibrio térmico.

Este método es sencillo, computacionalmente eficiente y estable [15]. Sin embargo, no reproduce con exactitud la distribución de velocidades de Maxwell-Boltzmann, por lo que no garantiza un correcto muestreo del colectivo canónico (NVT) [29]. Por ello, su uso se limita habitualmente a etapas de equilibrado inicial (ver [Tabla 5.2](#)), y no a fases de producción (ver [Tabla 5.2](#)) donde se requiera rigurosidad estadística.

³En este contexto, la ergodicidad implica que el promedio temporal de una magnitud a lo largo de una trayectoria del sistema coincide con su promedio en el colectivo estadístico, lo que permite obtener propiedades termodinámicas mediante simulaciones.

4. Algoritmos en Dinámica Molecular

Termostato de Andersen

El termostato de Andersen simula el acoplamiento del sistema con un baño térmico mediante colisiones aleatorias con partículas ficticias del entorno [34]. Para ello, en cada paso de simulación, una partícula es seleccionada al azar con una probabilidad p y su velocidad se reasigna a partir de una distribución de Maxwell-Boltzmann correspondiente a la temperatura objetivo.

Este procedimiento se basa en la siguiente distribución de probabilidad para la velocidad en una dimensión:

$$f(v) = \sqrt{\frac{m}{2\pi k_B T_0}} \exp\left(-\frac{m\|\vec{v}\|^2}{2k_B T_0}\right),$$

donde:

- \vec{v} es la velocidad instantánea de la partícula,
- m es la masa de la partícula,
- k_B es la constante de Boltzmann,
- T_0 es la temperatura objetivo.

Luego la velocidad en tres dimensiones se genera como un vector cuyas componentes se sortean de manera independiente según esta distribución.

Este método permite una correcta generación del colectivo canónico (NVT), ya que reproduce fielmente la distribución de velocidades térmicas [15, 29]. Sin embargo, al introducir aleatoriedad, puede provocar una ruptura de la conservación del momento lineal total del sistema, lo que puede ser un inconveniente en ciertos contextos físicos [28].

Termostato de Nosé-Hoover

El termostato de Nosé-Hoover extiende las ecuaciones de movimiento mediante la introducción de una variable dinámica adicional ζ que actúa como un "termostato virtual", permitiendo el intercambio de energía entre el sistema y un baño térmico ficticio [11, 12].

La evolución temporal de las partículas en este caso viene dada por:

$$\dot{\vec{v}}_i = \frac{1}{m_i} \vec{F}_i - \zeta \vec{v}_i, \quad \dot{\zeta} = \frac{1}{Q} \left(\sum_i m_i \|\vec{v}_i\|^2 - 3Nk_B T_0 \right),$$

donde:

- \vec{v}_i es la velocidad de la partícula i ,
- \vec{F}_i es la fuerza que actúa sobre dicha partícula,
- Q es un parámetro que representa el "momento de inercia térmico" del baño,
- ζ es la variable de fricción generalizada que evoluciona con el sistema,
- T_0 es la temperatura objetivo,

- k_B es la constante de Boltzmann,
- N es el número de partículas.

Despejando en (12), se observa que, en equilibrio térmico, se cumple la siguiente relación:

$$\sum_i m_i \|\vec{v}_i\|^2 = 3Nk_B T_0 \Rightarrow \dot{\xi} = 0.$$

Por tanto, esta formulación permite generar correctamente el colectivo canónico (NVT), asegurando tanto la reversibilidad temporal como el determinismo del sistema, a diferencia de los métodos vistos anteriormente [15]. Además, mantiene la continuidad de las trayectorias en el espacio de fases, lo que lo hace adecuado para el estudio de propiedades dinámicas [33].

No obstante, la elección del parámetro Q es crítica. Si se escoge un valor demasiado grande, el sistema responde lentamente al termostato y tarda mucho en alcanzar la temperatura deseada. Si, por el contrario, Q es muy pequeño, el sistema puede sufrir oscilaciones no físicas o inestabilidades numéricas. Por ello, su correcta parametrización suele ajustarse empíricamente o mediante técnicas de análisis de estabilidad [28].

4.2.2. Barostatos

Los barostatos modifican el volumen del sistema, ajustando dinámicamente el volumen de la celda de simulación, permitiendo así simular un entorno con presión constante. Al igual que con la temperatura, existen distintos métodos que abordan este objetivo:

Barostato de Berendsen

El barostato de Berendsen regula la presión del sistema mediante un reescalado suave y continuo del volumen hacia un valor objetivo, de forma análoga al termostato de Berendsen [31].

Este método se basa en una ecuación diferencial que modifica el volumen de la celda de simulación según la diferencia entre la presión actual y la deseada:

$$\frac{dV}{dt} = \frac{1}{\tau_P} (P_0 - P)V,$$

donde:

- V representa el volumen del sistema,
- P es la presión instantánea del sistema,
- P_0 representa la presión objetivo,
- τ_P es la constante de relajación, tiempo que tarda el sistema en alcanzar el nuevo estado de equilibrio.

Una vez actualizado el volumen del sistema, es necesario reescalar las posiciones de todas las partículas para que se ajusten coherentemente al nuevo volumen. Para ello, se define un factor de escala de longitud μ como:

4. Algoritmos en Dinámica Molecular

$$\mu = \left(\frac{V(t + \Delta t)}{V(t)} \right)^{1/3}.$$

Entonces, las posiciones se reescalan mediante:

$$\vec{r}_i \rightarrow \mu \vec{r}_i.$$

Este reescalado preserva la estructura relativa del sistema al modificar las distancias interatómicas en proporción al cambio de volumen. En simulaciones con condiciones periódicas de contorno, también se debe aplicar dicho reescalado a las dimensiones de la celda periódica.

En algunas implementaciones, también puede aplicarse un reescalado a las velocidades para mantener la coherencia dinámica, aunque en el barostato de Berendsen original esto no es estrictamente necesario.

Esta técnica es computacionalmente eficiente y estable, y permite controlar la presión media del sistema durante la simulación [15].

Sin embargo, al igual que ocurre con el termostato de Berendsen, este barostato no reproduce correctamente las fluctuaciones de volumen esperadas en un colectivo isóbaro-isotermo (NPT), lo que limita su rigor estadístico [29]. Por ello, aunque es útil en fases iniciales de equilibrio, no se recomienda su uso en la etapa de producción si se desean obtener propiedades termodinámicas con precisión, como el volumen promedio o la densidad del sistema.

Barostato de Parrinello-Rahman

El barostato de Parrinello-Rahman extiende la idea de Nosé-Hoover para permitir que no solo el volumen, sino también la forma de la celda de simulación, pueda cambiar de manera dinámica. Para ello, introduce como variable dinámica la matriz de celda h , cuyas columnas definen los vectores base del sistema simulado. Así, cada vector de la celda unitaria puede evolucionar de forma independiente, lo que permite deformaciones tanto isotrópicas como anisotrópicas [35]. Este enfoque resulta especialmente útil para simular materiales sólidos, fases cristalinas o sistemas sometidos a tensiones externas, donde no basta con escalar uniformemente el volumen, sino que también es necesario permitir cambios de forma.

Este método se basa en una formulación lagrangiana extendida, en la que se añaden grados de libertad adicionales asociados a la dinámica de la celda simulada. El sistema evoluciona bajo una dinámica acoplada entre las partículas y la matriz h , permitiendo así que la celda cambie de tamaño y forma en respuesta a la presión y al tensor de esfuerzos internos [15].

Aunque su implementación es más compleja y computacionalmente costosa que la de otros métodos, el barostato de Parrinello-Rahman permite una descripción precisa del colectivo isóbaro-isotermo (NPT), reproduciendo adecuadamente las fluctuaciones de volumen, por lo que es uno de los métodos más utilizados para simulaciones precisas en sistemas cristalinos o materiales bajo deformación [28].

Consideraciones sobre los tiempos de relajación

La elección de los tiempos de relajación, tanto térmica (τ_T) como barométrica (τ_P), es un aspecto crítico, ya que controlan la rapidez con la que el sistema se ajusta a la temperatura o presión objetivo [15].

Por tanto, si τ es demasiado pequeño, el sistema se fuerza de forma excesiva hacia el valor deseado, suprimiendo las variaciones naturales del sistema. Esto puede conducir a una evolución artificial del sistema y a una mala representación estadística, lo que resulta inadecuado para etapas de producción. Por otro lado, si τ es demasiado grande, la relajación será demasiado lenta, alargando innecesariamente la fase de equilibrado y pudiendo impedir alcanzar las condiciones deseadas en el tiempo de simulación disponible [15, 31].

En busca de programas más eficientes, se suele usar valores pequeños de τ durante las primeras etapas de equilibrado, en sistemas que parten de configuraciones lejos del equilibrio o en simulaciones en la que la precisión estadística no es una prioridad. Por el contrario, se suele usar valores grandes de τ durante las etapas finales de equilibrado, reduciendo gradualmente las perturbaciones externas, en sistemas pequeños o altamente sensibles, donde las oscilaciones provocadas por cambios bruscos pueden afectar negativamente a la simulación o en simulaciones más largas donde se busca una transición progresiva hacia la fase de producción.

Relajaciones sucesivas

En la práctica es habitual emplear una estrategia escalonada con varias etapas de relajación. Por ejemplo, en simulaciones bajo condiciones del colectivo NPT, se suele adoptar la siguiente secuencia:

1. En primer lugar, se aplica un termostato de Berendsen con un valor pequeño de τ_T para alcanzar rápidamente la temperatura objetivo.
2. A continuación, se introduce un barostato (también de Berendsen) con un valor reducido de τ_P para ajustar la presión del sistema.
3. Una vez estabilizado el sistema, se incrementan gradualmente los valores de τ_T y τ_P , o bien se cambia a métodos más rigurosos, como Nosé-Hoover o Parrinello-Rahman, para iniciar la fase de producción [29].

Esta estrategia permite un control eficiente y progresivo de las condiciones termodinámicas, evitando inestabilidades numéricas y asegurando un muestreo estadístico más fiable durante la etapa de análisis.

4.3. Algoritmos de optimización

Antes de iniciar una simulación de dinámica molecular, tras establecer la configuración inicial del sistema, es habitual llevar a cabo una etapa de minimización de energía, cuyo objetivo es encontrar una configuración estable del sistema que sirva como punto de partida para la integración de las ecuaciones de movimiento. En esta etapa se busca eliminar tensiones

4. Algoritmos en Dinámica Molecular

artificiales o solapamientos entre átomos que pueden haber sido introducidos durante la construcción inicial del sistema [29].

Por otro lado, los métodos de Monte Carlo ofrecen una alternativa interesante a la dinámica molecular clásica. Mientras que la MD resuelve ecuaciones diferenciales para obtener la evolución temporal del sistema, el enfoque de Monte Carlo introduce aleatoriedad y se centra en el muestreo de configuraciones según una determinada distribución de probabilidad. Este enfoque resulta especialmente útil en estudios termodinámicos o en simulaciones donde la evolución temporal explícita no es relevante [15].

4.3.1. Minimización de energía

La etapa de minimización de energía, como se ha mencionado anteriormente, constituye un paso previo habitual en muchas simulaciones de dinámica molecular. Su objetivo es encontrar una configuración estable del sistema, es decir, una disposición de las partículas que minimice la energía potencial total. Esta configuración corresponde, en general, a un mínimo local de la superficie de energía potencial, ya que el mínimo global suele ser inaccesible computacionalmente para sistemas con muchos grados de libertad [15, 29].

Además de garantizar la estabilidad mecánica local, esta etapa es crucial para evitar que fuerzas no físicas generen aceleraciones extremas o inestabilidades numéricas al comienzo de la simulación [15]. En particular, iniciar la integración desde una configuración alejada del equilibrio puede dar lugar a explosiones numéricas o a errores de integración acumulados.

En ciertos contextos, como el estudio del plegamiento de proteínas o el diseño de materiales, se recurre a técnicas de optimización global, como las metaheurísticas, que permiten escapar de mínimos locales y explorar regiones más amplias del paisaje energético [15, 36].

Desde un punto de vista matemático, esto equivale a resolver el siguiente problema de optimización:

$$\min_{\vec{r}_1, \dots, \vec{r}_N} V(\vec{r}_1, \dots, \vec{r}_N),$$

donde $V(\vec{r}_1, \dots, \vec{r}_N)$ representa la energía potencial total del sistema en función de las posiciones de los átomos.

Para alcanzar dicho mínimo local, muchos de los métodos más utilizados se basan en el gradiente del potencial, ya que los mínimos locales satisfacen las siguientes condiciones de primer y segundo orden:

$$\nabla_{\vec{r}_i} V = 0, \quad \text{Hess}V(\vec{r}_i) > 0, \quad (13)$$

donde la primera condición indica que la fuerza neta sobre cada partícula es nula, y la segunda que la matriz Hessiana es definida positiva en ese punto [37].

Recordemos que la matriz Hessiana de V viene dada por:

$$\text{Hess}V(\vec{r}) = \begin{bmatrix} \frac{\partial^2 V}{\partial r_{11}^2} & \frac{\partial^2 V}{\partial r_{11} \partial r_{12}} & \dots & \frac{\partial^2 V}{\partial r_{11} \partial r_{N3}} \\ \frac{\partial^2 V}{\partial r_{12} \partial r_{11}} & \frac{\partial^2 V}{\partial r_{12}^2} & \dots & \frac{\partial^2 V}{\partial r_{12} \partial r_{N3}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 V}{\partial r_{N3} \partial r_{11}} & \frac{\partial^2 V}{\partial r_{N3} \partial r_{12}} & \dots & \frac{\partial^2 V}{\partial r_{N3}^2} \end{bmatrix}, \quad (14)$$

donde $\vec{r} = (r_{11}, r_{12}, r_{13}, \dots, r_{N1}, r_{N2}, r_{N3})^T \in \mathbb{R}^{3N}$, para un sistema de N partículas, cada una con posición tridimensional $\vec{r}_i = (r_{i1}, r_{i2}, r_{i3})$. Lo habitual en el contexto de dinámica molecular es que esta matriz sea simétrica, ya que, en general, se trabaja con funciones suaves [26].

En la práctica, se considera que el sistema ha alcanzado un equilibrio mecánico local cuando la fuerza neta sobre cada átomo es inferior a un umbral prefijado.

A continuación se presentan algunos de los métodos de optimización más utilizados en simulaciones de dinámica molecular:

Descenso por gradiente

Este método, estudiado en la asignatura Aprendizaje Automático, consiste en actualizar iterativamente las posiciones de las partículas en la dirección del gradiente negativo del potencial hasta que la fuerza neta sobre cada átomo quede por debajo del umbral prefijado [15, 37]:

Algoritmo 1 Descenso por gradiente

```

 $\vec{r} \leftarrow \vec{r}_0$ 
while  $\|\nabla V(\vec{r})\| > \varepsilon$  do
     $\vec{r} \leftarrow \vec{r} - \alpha \nabla V(\vec{r})$ 
end while
return  $\vec{r}$ 

```

donde \vec{r}_0 representa la configuración inicial del sistema, α es el paso de aprendizaje o tasa de actualización, y ε el umbral prefijado.

Este método es especialmente útil cuando la configuración inicial se encuentra lejos del mínimo local. Sin embargo, el método puede volverse ineficiente al aproximarse al mínimo, ya que si el valor de α es demasiado grande, se producirán oscilaciones, mientras que si el valor de α es muy pequeño, la convergencia será muy lenta (ver Figura 11).

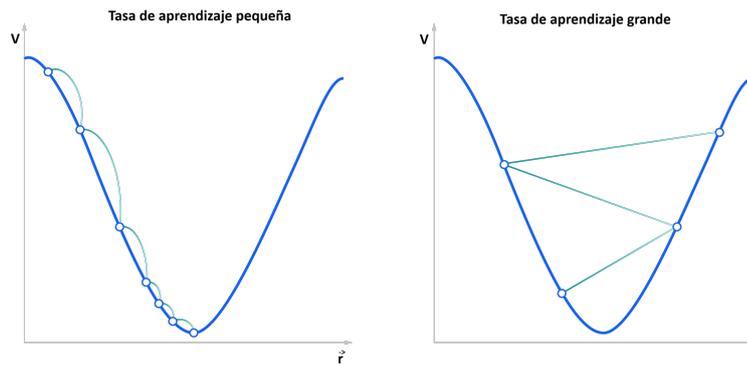


Figura 11.: Comparación del descenso por gradiente según la tasa de aprendizaje [38].

4. Algoritmos en Dinámica Molecular

Gradiente conjugado

El método del gradiente conjugado mejora la eficiencia del descenso por gradiente al generar, en cada iteración, una dirección de búsqueda conjugada respecto a la matriz Hessiana del sistema [37]. Esto permite una convergencia más rápida, especialmente cerca del mínimo.

Desde un punto de vista matemático, desarrollando $V(\vec{r})$ alrededor de la configuración inicial \vec{r}_0 :

$$V(\vec{r}) \approx V(\vec{r}_0) + \nabla V(\vec{r}_0)^T(\vec{r} - \vec{r}_0) + \frac{1}{2}(\vec{r} - \vec{r}_0)^T \text{Hess}V(\vec{r}_0)(\vec{r} - \vec{r}_0),$$

donde $\text{Hess}V(\vec{r}_0)$ es la matriz Hessiana (14) evaluada en \vec{r}_0 .

Ahora, definiendo $\vec{x} = \vec{r} - \vec{r}_0$ y observando que el término constante $V(\vec{r}_0)$ no afecta al cálculo del mínimo local (véase (13)), pues su derivada es nula, se escribe:

$$V(\vec{r}) \approx \frac{1}{2}\vec{x}^T \text{Hess}V(\vec{r}_0)\vec{x} + \nabla V(\vec{r}_0)^T \vec{x} = \frac{1}{2}\vec{x}^T A \vec{x} - \vec{x}^T b,$$

tomando $A = \text{Hess}V(\vec{r}_0)$ y $b = -\nabla V(\vec{r}_0)$. Por tanto, el método del gradiente conjugado busca minimizar la función cuadrática:

$$f(\vec{x}) = \frac{1}{2}\vec{x}^T A \vec{x} - \vec{x}^T b.$$

El algoritmo del gradiente conjugado viene dado por:

Algoritmo 2 Gradiente conjugado

```

 $\vec{x}_0 \leftarrow A\vec{r}_0 - \vec{b}$ 
 $\vec{p}_0 \leftarrow -\vec{x}_0$ 
 $k \leftarrow 0$ 
while  $\|\vec{x}_k\| > \varepsilon$  do
   $\vec{\alpha}_k \leftarrow \frac{\vec{x}_k^T \vec{x}_k}{\vec{p}_k^T A \vec{p}_k}$ 
   $\vec{r}_{k+1} \leftarrow \vec{r}_k + \vec{\alpha}_k \vec{p}_k$ 
   $\vec{x}_{k+1} \leftarrow \vec{x}_k + \vec{\alpha}_k A \vec{p}_k$ 
   $\vec{\beta}_k \leftarrow \frac{\vec{x}_{k+1}^T \vec{x}_{k+1}}{\vec{x}_k^T \vec{x}_k}$ 
   $\vec{p}_{k+1} \leftarrow -\vec{x}_{k+1} + \vec{\beta}_{k+1} \vec{p}_k$ 
   $k \leftarrow k + 1$ 
end while
return  $\vec{r}_k$ 

```

En simulaciones reales se suele emplear una combinación de ambos métodos, comenzando con descenso por gradiente para escapar rápidamente de regiones de alta energía, y continuando con gradiente conjugado cerca del mínimo local [15].

Metaheurísticas y optimización global

Si bien el descenso por gradiente y el gradiente conjugado son ampliamente utilizados por su eficacia y simplicidad, por lo que son preferibles en simulaciones rutinarias, tienen limitaciones importantes en escenarios donde la función objetivo posee múltiples mínimos locales, como ocurre en biomoléculas plegadas.

Para estos casos, han ganado relevancia las metaheurísticas, técnicas de optimización global que no dependen únicamente de información local del gradiente, sino que están diseñadas para explorar globalmente el espacio de búsqueda, permitiendo escapar de mínimos locales a través de mecanismos estocásticos o evolutivos. Entre las más utilizadas se encuentran los algoritmos genéticos y evolutivos [39], estudiados en la asignatura Metaheurísticas, el método de *Basin Hopping* [36] y el descenso estocástico del gradiente (*Stochastic Gradient Descent*, SGD) [40], estudiado en la asignatura Aprendizaje Automático. Estas técnicas, aunque no garantizan encontrar el mínimo global, aumentan significativamente la probabilidad de alcanzarlo.

Si bien exigen un mayor coste computacional en comparación con los métodos deterministas basados en gradientes, su uso está justificado en situaciones donde es prioritario encontrar mínimos globales, como en el estudio del plegamiento de proteínas o el diseño de materiales.

En resumen, la elección del método de optimización depende de la topología de la superficie de energía potencial, del conocimiento previo del sistema y de los recursos computacionales disponibles.

4.3.2. Métodos de Monte Carlo

El enfoque de Monte Carlo (MC), diseñado principalmente para el colectivo canónico (NVT), ofrece una alternativa eficaz y conceptualmente diferente a la dinámica molecular tradicional. Mientras que la dinámica molecular integra las ecuaciones de movimiento para obtener la evolución temporal de las partículas, los métodos de MC generan configuraciones aleatorias del sistema y deciden si se aceptan o no según un criterio probabilístico. El objetivo es muestrear configuraciones de acuerdo con una distribución de probabilidad del tipo Boltzmann:

$$P(\vec{r}) = \frac{1}{Z} \exp\left(-\frac{V(\vec{r})}{k_B T}\right), \quad (15)$$

donde Z es la constante de normalización o función de partición:

$$Z = \int \exp\left(-\frac{V(\vec{r})}{k_B T}\right) d\vec{r}. \quad (16)$$

A partir de esta expresión, las configuraciones con menor energía tienen mayor probabilidad de ser aceptadas. Sin embargo, también se permite aceptar configuraciones con mayor energía para evitar quedar atrapados en mínimos locales.

Los métodos de Monte Carlo tienen la ventaja de ser más fáciles de implementar y suelen ser más eficientes cuando se trata de muestrear grandes espacios de configuración, especialmente en sistemas con restricciones geométricas, como sólidos, redes cristalinas o polímeros [41].

4. Algoritmos en Dinámica Molecular

El algoritmo más conocido en este contexto es el algoritmo de Metropolis [42], desarrollado en 1953. Este método permite muestrear eficientemente el espacio de configuraciones de un sistema a temperatura constante, generando un conjunto de estados que siguen la distribución de Boltzmann (15). Su ventaja fundamental es que no requiere calcular la función de partición (16), una cantidad computacionalmente inalcanzable en sistemas con muchos grados de libertad, ya que implica una integral en un espacio de alta dimensionalidad [15, 43].

En su lugar, el algoritmo trabaja con el cociente de probabilidades entre estados consecutivos:

$$\frac{P(\vec{r}')}{P(\vec{r})} = \frac{\frac{1}{Z} \exp\left(-\frac{V(\vec{r}')}{k_B T}\right)}{\frac{1}{Z} \exp\left(-\frac{V(\vec{r})}{k_B T}\right)} = \exp\left(\frac{V(\vec{r}) - V(\vec{r}')}{k_B T}\right) = \exp\left(-\frac{\Delta V}{k_B T}\right) = {}^4 \exp\left(-\frac{\Delta E}{k_B T}\right).$$

Aunque este cociente no representa una probabilidad en sentido estricto, el algoritmo lo emplea como probabilidad de aceptación cuando $\Delta E > 0$. En tal caso, se define $p = \exp\left(-\frac{\Delta E}{k_B T}\right) \in (0, 1)$, y se compara con un número aleatorio $r \in [0, 1]$, lo que permite al sistema aceptar configuraciones energéticamente desfavorables con cierta probabilidad, facilitando así la salida de mínimos locales. En cambio, si $\Delta E < 0$, se acepta directamente la nueva configuración, asegurando que el sistema puede evolucionar hacia estados de menor energía.

Por tanto, su implementación viene dada por:

Algoritmo 3 Algoritmo de Metropolis

```
Inicializar configuración  $\vec{r}$  al azar
for paso = 1 hasta  $N$  do
  Elegir una partícula  $i$  al azar
  Calcular  $\vec{r}'$  a partir de  $\vec{r}$ , modificando la posición de la partícula  $i$ 
  Calcular  $\Delta E = V(\vec{r}') - V(\vec{r})$ 
  if  $\Delta E < 0$  then
    Aceptar la nueva configuración:  $\vec{r} \leftarrow \vec{r}'$ 
  else
    Calcular  $p = \exp\left(-\frac{\Delta E}{k_B T}\right)$ 
    Generar número aleatorio  $r \in [0, 1]$ 
    if  $r < p$  then
      Aceptar:  $\vec{r} \leftarrow \vec{r}'$ 
    else
      Rechazar: mantener configuración actual
    end if
  end if
end for
```

⁴En el contexto del algoritmo de Metropolis aplicado a simulaciones Monte Carlo bajo el colectivo canónico (NVT), los términos cinéticos no se consideran explícitamente, ya que la energía cinética no influye en la distribución de probabilidad configuracional. Por ello, se asume que el incremento de energía ΔE equivale al cambio en la energía potencial ΔV [15].

Este método garantiza que, tras un número suficiente de pasos, las configuraciones visitadas siguen la distribución de Boltzmann, lo cual es útil para el cálculo de propiedades termodinámicas en equilibrio.

Por tanto, aunque el método no permite estudiar trayectorias o dinámicas temporales, es especialmente útil para explorar estados de equilibrio, detectar configuraciones estables y estimar propiedades macroscópicas del sistema.

En resumen, mientras que la minimización de energía busca una configuración estable inicial resolviendo un problema de optimización determinista, los métodos de Monte Carlo exploran distintas configuraciones del sistema mediante una estrategia probabilística.

4.4. Simulaciones paralelas y aceleración

Las simulaciones de dinámica molecular requieren una elevada carga computacional debido a la complejidad de los cálculos implicados. A medida que el número de partículas crece o se desea simular procesos en escalas temporales más largas, el número de operaciones necesarias se incrementa de forma exponencial. En particular, el cálculo de fuerzas entre partículas y la integración de las ecuaciones de movimiento representan los principales cuellos de botella computacionales, siendo estos cálculos de complejidad $\mathcal{O}(N^2)$ en el caso más general [15, 29].

Para poder simular sistemas de interés biológico o materiales realistas con millones de átomos durante tiempos físicamente relevantes, es fundamental paralelizar el proceso de simulación. Este objetivo se logra mediante diferentes estrategias de paralelización, que permiten distribuir el trabajo entre múltiples procesadores o unidades de cómputo:

Descomposición espacial

Una de las técnicas más utilizadas es la descomposición espacial, donde el espacio simulado se divide en subdominios (cajas) y cada procesador es responsable de calcular las fuerzas e integrar las trayectorias de las partículas dentro de su región. Para mantener la coherencia global, los procesadores deben intercambiar información sobre partículas cercanas a los límites de sus dominios, denominadas partículas fantasma, lo cual requiere comunicación eficiente entre nodos [21].

Replicación de datos

En sistemas de menor tamaño o con baja comunicación entre procesos, se puede optar por replicar los datos en todos los procesadores, de manera que cada uno tenga una copia completa del sistema. Aunque esto incrementa el uso de memoria, permite minimizar la latencia de comunicación, a cambio de realizar cálculos redundantes [15].

Paralelización por fuerza o por átomo

Otra opción es paralelizar por tipo de cálculo, asignando por ejemplo a cada núcleo una fracción de las interacciones a calcular (paralelización por fuerzas) o el seguimiento de un conjunto de partículas específicas (paralelización por átomos). Aunque menos escalables,

4. Algoritmos en Dinámica Molecular

estas estrategias pueden ser útiles en arquitecturas específicas o en algoritmos especializados [26].

Uso de arquitecturas HPC y GPUs

El uso de arquitecturas de computación de alto rendimiento (*High-Performance Computing*, HPC) es fundamental para llevar a cabo simulaciones de dinámica molecular a gran escala. Sistemas como supercomputadores o clústeres permiten dividir el trabajo entre múltiples núcleos o nodos mediante bibliotecas paralelas como OpenMP [44] o MPI (*Message Passing Interface*) [45], acelerando así la ejecución de las simulaciones.

Por otro lado, las unidades de procesamiento gráfico (GPUs) han supuesto un avance significativo en este ámbito. A diferencia de las CPUs tradicionales, que están optimizadas para ejecutar unos pocos hilos de manera muy eficiente, las GPUs están diseñadas para manejar miles de hilos de manera simultánea. Esta arquitectura masivamente paralela es ideal para realizar operaciones vectoriales repetitivas, como el cálculo de fuerzas entre partículas, que constituye el núcleo computacional de la dinámica molecular [46].

Gracias a estas ventajas, muchos programas populares de simulación, como AMBER, GRO-MACS [47] o NAMD, han incorporado versiones optimizadas para su ejecución en GPU, permitiendo alcanzar aceleraciones de hasta un orden de magnitud (es decir, 10 veces más rápidas) con respecto a implementaciones basadas únicamente en CPU.

En conclusión, la paralelización y el uso de arquitecturas aceleradas son elementos fundamentales en la dinámica molecular actual. No solo permiten simular sistemas más grandes o durante más tiempo, sino que permiten llevar a cabo estudios que serían computacionalmente inviables con recursos secuenciales.

5. Implementación de una Simulación de Dinámica Molecular

Una vez establecidos los conceptos teóricos fundamentales de la dinámica molecular en los capítulos anteriores, una de las mejores formas de comprender una simulación es plantear cómo se estructuraría un programa simple bajo las siguientes condiciones:

Condiciones termodinámicas y parámetros característicos de la simulación

La simulación presentada en esta sección se lleva a cabo bajo las condiciones del colectivo microcanónico (NVE). En este conjunto estadístico se mantiene constante el número de partículas N , el volumen V y la energía total E (ver [Capítulo 2](#)).

Durante la evolución temporal, se calculan diversas propiedades termodinámicas del sistema, como la temperatura instantánea y la energía total por partícula.

Unidades reducidas

En simulaciones de dinámica molecular es habitual emplear un sistema de unidades reducidas con el fin de simplificar los cálculos, mejorar la estabilidad numérica y evitar errores derivados de trabajar con constantes físicas muy pequeñas (por ejemplo, $k_B \sim 10^{-23}$ J/K). Esta estrategia también permite escribir código más limpio y eficiente, eliminando la necesidad de introducir factores dimensionales en cada paso del algoritmo [15].

La idea fundamental consiste en escoger magnitudes características del sistema como escalas de referencia. En particular, cuando se emplea el potencial de Lennard-Jones, se fijan como unidades base las siguientes constantes:

- σ : longitud característica (diámetro efectivo de las partículas), se mide en metros [m].
- ε : energía de interacción (profundidad del pozo de potencial), se mide en julios [J].
- m : masa de las partículas, se mide en kilogramos [kg].
- k_B : constante de Boltzmann.

Esto permite expresar todas las demás magnitudes relevantes en forma adimensional, es decir, sin unidades físicas explícitas. A partir de estas definiciones base, se pueden reescalar el resto de variables del sistema, tal y como se resume en la [Tabla 5.1](#).

Por ejemplo, supongamos que en una simulación se emplean los valores $\sigma = \varepsilon = m = k_B = 1$ (elección habitual en unidades reducidas), y se obtiene una temperatura reducida $T^* = 1.2$. Entonces, empleando los valores físicos reales del argón, se obtiene la siguiente temperatura instantánea:

$$T = \frac{T^* \cdot \varepsilon}{k_B} \approx \frac{1.2 \cdot 1.65 \times 10^{-21}}{1.38 \times 10^{-23}} \cdot \frac{\text{J}}{\text{J/K}} \approx \frac{1.98 \times 10^{-21}}{1.38 \times 10^{-23}} \text{K} \approx 143.5 \text{K}.$$

5. Implementación de una Simulación de Dinámica Molecular

De igual forma, si se desea que, en valores físicos reales, el paso de tiempo empleado sea de $\Delta t = 10^{-15}$ s, entonces el paso de tiempo tomado en la simulación vendrá dado por:

$$t = t^* \cdot \sigma \sqrt{\frac{m}{\epsilon}} = t^* \cdot 3.405 \times 10^{-10} \sqrt{\frac{6.63 \times 10^{-26}}{1.65 \times 10^{-21}}} = 10^{-15} \Rightarrow t^* \approx 4.636 \times 10^{-4}.$$

Magnitud	Magnitud reducida	Reescalado a unidades físicas
t	$t^* = \frac{t}{\sigma \sqrt{m/\epsilon}}$	$t = t^* \cdot \sigma \sqrt{\frac{m}{\epsilon}}$ (s)
T	$T^* = \frac{k_B T}{\epsilon}$	$T = \frac{T^* \cdot \epsilon}{k_B}$ (K)
\vec{v}	$\vec{v}^* = \frac{\vec{v}}{\sqrt{\epsilon/m}}$	$\vec{v} = \vec{v}^* \cdot \sqrt{\frac{\epsilon}{m}}$ (m/s)
P	$P^* = \frac{P \cdot \sigma^3}{\epsilon}$	$P = P^* \cdot \frac{\epsilon}{\sigma^3}$ (Pa)
\vec{F}	$\vec{F}^* = \frac{\vec{F} \cdot \sigma}{\epsilon}$	$\vec{F} = \vec{F}^* \cdot \frac{\epsilon}{\sigma}$ (N)

Tabla 5.1.: Conversión entre magnitudes físicas y reducidas

Aunque las variables numéricas utilizadas en simulaciones con unidades reducidas son adimensionales y no poseen unidades físicas explícitas, sí representan de forma coherente las proporciones y relaciones entre las propiedades del sistema. Esto permite interpretar los resultados de forma cualitativa, comparar distintos sistemas moleculares y, si se desea, convertir los valores obtenidos nuevamente a unidades físicas mediante las expresiones de reescalado correspondientes [15].

Estructura del programa

En la [Tabla 5.2](#) se muestra la secuencia típica de etapas que componen una simulación de dinámica molecular.

Por tanto, el pseudocódigo desarrollado tendrá la siguiente estructura:

1. Lectura de los parámetros que especifican las condiciones iniciales de la simulación (número de partículas, temperatura inicial, paso de tiempo, etc.).
2. Inicialización del sistema: incluye la preparación, el calentamiento y el equilibrado del sistema.
3. Cálculo de las fuerzas que actúan sobre cada partícula.
4. Integración de las ecuaciones de movimiento de Newton.
5. Cálculo y visualización de propiedades termodinámicas del sistema.

Etapa	Descripción
Preparación del sistema	Definición de las coordenadas iniciales y asignación de velocidades, habitualmente mediante una distribución de Maxwell-Boltzmann.
Calentamiento	Escalado de las velocidades a la temperatura deseada.
Equilibrado	El sistema se lleva a una situación de equilibrio a partir de su configuración inicial.
Producción	Generación de las trayectorias del sistema, a partir de las cuales se calculan propiedades físicas y termodinámicas.

Tabla 5.2.: Etapas del desarrollo de una simulación de dinámica molecular. [48]

A partir de esta estructura, se obtiene el siguiente pseudocódigo, que implementa una simulación de dinámica molecular para un sistema atómico tridimensional sencillo, compuesto por N partículas:

Algoritmo 4 Esquema general de una simulación de dinámica molecular

```

Leer parámetros                                     ▷ Paso 1
Inicializar sistema                                 ▷ Paso 2
 $t \leftarrow 0$ 
while  $t < t_{\max}$  do
  Calcular fuerzas                                   ▷ Paso 3
  Integrar ecuaciones de movimiento                 ▷ Paso 4
  Calcular y mostrar propiedades                     ▷ Paso 5
   $t \leftarrow t + \Delta t$ 
end while

```

Las subrutinas inicializar sistema, calcular fuerzas e integrar ecuaciones de movimiento serán descritas en los algoritmos 5, 6 y 7, respectivamente.

5.1. Inicialización del sistema

En esta fase se llevan a cabo tres pasos esenciales: la preparación, el calentamiento y el equilibrado del sistema. En primer lugar, se definen las coordenadas iniciales de las partículas, que suelen colocarse sobre una red regular (como una red cúbica) para evitar solapamientos, y se asignan velocidades iniciales extraídas de una distribución aleatoria. Luego, se realiza un reescalado de las velocidades con el objetivo de que la energía cinética total del sistema corresponda a la temperatura deseada, lo que se conoce como calentamiento del sistema. Finalmente, se elimina la posible velocidad del centro de masas y se ajustan las condiciones para alcanzar un estado de equilibrio termodinámico.

Estas tareas pueden implementarse de forma conjunta dentro de una única rutina de inicialización (ver Algoritmo 5). A continuación se detallan los cálculos realizados en ella:

- **Preparación del sistema:** Las posiciones iniciales \vec{r}_i de las partículas se colocan sobre una red regular tridimensional (habitualmente cúbica), mediante una función `lattice_pos(i)`. Este paso evita solapamientos entre partículas que podrían producir fuerzas repulsivas extremadamente grandes en los primeros pasos de simulación [15].

5. Implementación de una Simulación de Dinámica Molecular

- **Asignación de velocidades aleatorias:** A cada partícula i se le asignan tres componentes de velocidad, una por cada dirección del espacio, utilizando números aleatorios uniformemente distribuidos en el intervalo $[-0.5, 0.5]$. Esta elección se emplea exclusivamente para garantizar una distribución inicial no sesgada. Posteriormente, tras el equilibrado, las velocidades seguirán una distribución de Maxwell-Boltzmann.
- **Eliminación del movimiento del centro de masas:** En una simulación de dinámica molecular, es importante que el sistema no adquiera un movimiento global no deseado, es decir, que el centro de masas permanezca en reposo. Si las velocidades iniciales se asignan aleatoriamente, es muy probable que la velocidad total del sistema no sea exactamente cero. Esto puede provocar un desplazamiento del sistema en su conjunto a lo largo del tiempo, lo cual no se desea [15].

Para corregir esto, se calcula la velocidad del centro de masas en cada dirección espacial, que, asumiendo $m = 1$, viene dada por la media de las velocidades individuales de todas las partículas:

$$\bar{v}_j = \frac{1}{N} \sum_{i=1}^N \vec{v}_{ij} \quad \text{para } j = 1, 2, 3,$$

donde \vec{v}_{ij} representa la componente j -ésima de la velocidad de la partícula i , y N es el número total de partículas. Esta media define el vector de velocidad del centro de masas del sistema:

$$\vec{v}_{\text{CM}} = (\bar{v}_1, \bar{v}_2, \bar{v}_3).$$

Una vez calculado este vector, se elimina el movimiento del centro de masas restándolo de la velocidad de cada partícula:

$$\vec{v}_i \leftarrow \vec{v}_i - \vec{v}_{\text{CM}} \quad \text{para } i = 1, \dots, N.$$

De este modo, se garantiza que el sistema tenga velocidad total nula.

- **Reescalado de las velocidades (calentamiento):** Para garantizar que la energía cinética media del sistema corresponde con la temperatura objetivo T , se utiliza el teorema de la equipartición (12), donde, suponiendo $m = 1$ y $k_B = 1$, se ajustan las velocidades mediante un factor de escala:

$$f_s = \sqrt{\frac{3T}{\sum_{i=1}^N \|\vec{v}_i\|^2 / N}}.$$

Por tanto, aplicando $\vec{v}_{ij} \leftarrow \vec{v}_{ij} \cdot f_s$ se consigue que la temperatura inicial del sistema sea exactamente T . Además, este procedimiento garantiza que las velocidades de las partículas sigan una distribución de Maxwell-Boltzmann [15].

- **Cálculo de las posiciones en $t - \Delta t$:** Finalmente, se calcula la posición de cada partícula en el paso anterior ($t - \Delta t$), necesaria para comenzar la integración mediante el algoritmo de Verlet (7). Para ello, se usan las velocidades asignadas suponiendo movimiento a velocidad constante, es decir, sin suponer ningún tipo de interacción ni fuerza externa.

Estos pasos completan la fase de inicialización del sistema, permitiendo comenzar la simulación con condiciones físicas coherentes, sin movimiento global del centro de masas, con una temperatura precisa y sin solapamientos significativos.

Algoritmo 5 Inicializar sistema

```

sumv1 ← 0, sumv2, sumv3 ← 0                                ▷ Suma de las velocidades
sumv2 ← 0                                                    ▷ Suma de las velocidades al cuadrado
for  $i = 1$  to  $N$  do
   $\vec{r}_i \leftarrow \text{lattice\_pos}(i)$                         ▷ Coloca la partícula  $i$  en una red regular
  for  $j = 1$  to 3 do
     $\vec{v}_{ij} \leftarrow \text{random}() - 0.5$                 ▷ Velocidad aleatoria en cada dirección
    sumv $j$  ← sumv $j$  +  $\vec{v}_{ij}$                         ▷ Suma para calcular velocidad del centro de masas
    sumv2 ← sumv2 +  $\vec{v}_{ij}^2$                             ▷ Actualización velocidades al cuadrado
  end for
end for
 $\vec{v}_{CM} \leftarrow (\text{sumv}_1/N, \text{sumv}_2/N, \text{sumv}_3/N)$       ▷ Velocidad del centro de masas
sumv2 ← sumv2/ $N$                                           ▷ Calcula la velocidad cuadrática media
 $f_s \leftarrow \sqrt{3 \cdot T_{\text{objetivo}}/\text{sumv2}}$         ▷ Escala para ajustar temperatura
for  $i = 1$  to  $N$  do
   $\vec{v}_i \leftarrow \vec{v}_i - \vec{v}_{CM}$                         ▷ Sistema en reposo global
   $\vec{v}_i \leftarrow \vec{v}_i \cdot f_s$                         ▷ Calentamiento
   $r\_prev_i \leftarrow \vec{r}_i - \vec{v}_i \cdot \Delta t$         ▷ Coordenadas de la partícula  $i$  en  $t - \Delta t$ 
end for

```

5.2. Cálculo de fuerzas

Una vez inicializado el sistema, en cada paso de la simulación es necesario calcular las fuerzas que actúan sobre cada partícula. Este cálculo se basa en el modelo de potencial que describe las interacciones entre partículas. En este caso, se emplea el potencial de Lennard-Jones (4), ampliamente utilizado para modelar interacciones de tipo van der Waals.

El algoritmo 6 realiza el cálculo de las fuerzas interatómicas y de la energía potencial total del sistema. Para mejorar la eficiencia computacional, se introduce un radio de corte r_c que evita calcular la interacción entre pares de partículas cuya distancia es suficientemente grande como para que su efecto sea despreciable (ver Subsección 3.2.4). Además, se aplican condiciones de contorno periódicas para simular un sistema infinito a partir de una celda finita (ver Figura 6).

A continuación, se detalla el proceso llevado a cabo:

- **Inicialización de la energía y fuerzas:** Se inicializa la energía potencial total del sistema con $E_{\text{pot}} \leftarrow 0$. Además, se pone a cero la fuerza total sobre cada partícula, es decir, $\vec{F}_i \leftarrow (0, 0, 0)$ para $i = 1, \dots, N$. Esto garantiza que no queden restos de cálculos previos que puedan alterar los resultados.
- **Bucle sobre pares de partículas:** Se recorren todos los pares únicos de partículas (i, j) con $i < j$, ya que la interacción de Lennard-Jones es simétrica (la fuerza de i sobre j es igual y opuesta a la de j sobre i). Esto permite ahorrar tiempo de cómputo.

5. Implementación de una Simulación de Dinámica Molecular

- **Cálculo de distancias con condiciones de contorno periódicas:** Para cada par de partículas (i, j) con $i < j$, se calcula el vector que une ambas posiciones:

$$\vec{r}_{ij} = \vec{r}_i - \vec{r}_j.$$

Para simular un sistema infinito con una celda finita, se aplican condiciones de contorno periódicas usando la mínima imagen, ya que puede ocurrir que la distancia más corta entre dos partículas no sea la directa, sino a través del borde del sistema ((ver [Figura 6](#)):

$$\vec{r}_{ij} \leftarrow \vec{r}_{ij} - L \cdot \text{round}(\vec{r}_{ij}/L),$$

donde round es una función que redondea al entero más cercano, y L representa el diámetro de la celda. Esto garantiza que cada interacción se calcula con la imagen más cercana de la otra partícula, asegurando que las distancias se mantengan dentro del intervalo $[-L/2, L/2]$ [15].

- **Filtrado por radio de corte:** Con el fin de evitar cálculos innecesarios de raíces cuadradas, que son computacionalmente costosos, se calcula la distancia al cuadrado $r^2 = \|\vec{r}_{ij}\|^2$, y solo se evalúa la interacción si $r^2 < r_c^2$; es decir, si la distancia entre las partículas es menor que el radio de corte r_c . Este criterio mejora el rendimiento computacional sin afectar significativamente a la precisión física.
- **Cálculo de la fuerza mediante el potencial de Lennard-Jones:** El módulo de la fuerza se obtiene a partir de la derivada del potencial de Lennard-Jones (4), donde, suponiendo $\epsilon = 1$, se tiene que:

$$\vec{F}_{ij} = \nabla V_{LJ}(r_{ij}) \cdot \frac{\vec{r}_{ij}}{r_{ij}} = 24 \cdot \left(\frac{2}{r_{ij}^{13}} - \frac{1}{r_{ij}^7} \right) \cdot \frac{\vec{r}_{ij}}{r_{ij}} = 24 \cdot \left(\frac{2}{(r_{ij}^2)^7} - \frac{1}{(r_{ij}^2)^4} \right) \cdot \vec{r}_{ij} = f \cdot \vec{r}_{ij},$$

donde r_{ij} representa la distancia entre las partículas i y j . De forma análoga, esta expresión se ha formulado usando r_{ij}^2 para evitar cálculos innecesarios de raíces cuadradas.

- **Actualización de las fuerzas:** La fuerza vectorial se obtiene multiplicando el módulo f por el vector dirección \vec{r}_{ij} :

$$\vec{F}_i \leftarrow \vec{F}_i + f \cdot \vec{r}_{ij}, \quad \vec{F}_j \leftarrow \vec{F}_j - f \cdot \vec{r}_{ij}.$$

Esto asegura que se cumpla la tercera ley de Newton (2).

- **Cálculo de la energía potencial:** La energía potencial entre i y j se suma al total:

$$E_{\text{pot}} \leftarrow E_{\text{pot}} + 4 \cdot \left(\frac{1}{(r_{ij}^2)^6} - \frac{1}{(r_{ij}^2)^3} \right) - E_{\text{cut}},$$

donde E_{cut} es una constante que compensa el truncamiento del potencial, asegurando la continuidad del mismo en el radio de corte [15]. Se puede calcular como el valor del potencial en $r = r_c$, luego:

$$E_{\text{cut}} \leftarrow 4 \left(\frac{1}{r_c^{12}} - \frac{1}{r_c^6} \right).$$

Algoritmo 6 Calcular fuerzas

```

 $E_{\text{pot}} \leftarrow 0$  ▷ Inicializa la energía potencial total
for  $i = 1$  to  $N$  do
   $\vec{F}_i \leftarrow (0, 0, 0)$  ▷ Inicializa la fuerza sobre la partícula  $i$ 
end for
for  $i = 1$  to  $N - 1$  do
  for  $j = i + 1$  to  $N$  do
     $\vec{r}_{ij} \leftarrow \vec{r}_i - \vec{r}_j$  ▷ Diferencia de posición
     $\vec{r}_{ij} \leftarrow \vec{r}_{ij} - L \cdot \text{round}(\vec{r}_{ij}/L)$  ▷ Condiciones de contorno periódicas
     $r2 \leftarrow \vec{r}_{ij}[0]^2 + \vec{r}_{ij}[1]^2 + \vec{r}_{ij}[2]^2$  ▷ Distancia al cuadrado
    if  $r2 < r_c^2$  then ▷ Verifica si está dentro del radio de corte
       $f \leftarrow 24 \cdot \left( \frac{2}{(r2)^7} - \frac{1}{(r2)^4} \right)$  ▷ Potencial de Lennard-Jones dividido por  $\|\vec{r}_{ij}\|$ 
       $\vec{F}_i \leftarrow \vec{F}_i + f \cdot \vec{r}_{ij}$  ▷ Actualiza fuerza sobre la partícula  $i$ 
       $\vec{F}_j \leftarrow \vec{F}_j - f \cdot \vec{r}_{ij}$  ▷ Actualiza fuerza sobre la partícula  $j$ 
       $E_{\text{pot}} \leftarrow E_{\text{pot}} + 4 \cdot \left( \frac{1}{(r2)^6} - \frac{1}{(r2)^3} \right) - E_{\text{cut}}$  ▷ Actualiza la energía potencial total
    end if
  end for
end for

```

5.3. Integración de las ecuaciones de movimiento

Tras calcular las fuerzas que actúan sobre cada partícula, se procede a integrar las ecuaciones de movimiento para obtener las nuevas posiciones y velocidades (ver algoritmo 7). En este caso, se utiliza el algoritmo de Verlet.

Por tanto, se estiman las posiciones de las partículas en cada paso de tiempo a partir de (7) y, seguidamente, se estimarán las velocidades a partir de (8).

A continuación, se detalla el proceso llevado a cabo:

- **Inicialización de la energía cinética:** Se define la variable `sumv2`, que acumulará el valor de las velocidades al cuadrado de todas las partículas, necesaria para estimar la temperatura y la energía cinética.
- **Cálculo de la nueva posición (Verlet):** Para cada partícula i , se calcula la nueva posición mediante la fórmula (7).
- **Estimación de la velocidad:** Para cada partícula i , se calcula la nueva velocidad mediante la fórmula (8).
- **Actualización de posiciones:** La posición anterior se actualiza como la actual, y la actual se reemplaza por la nueva estimada. Así, en el siguiente paso, se puede reutilizar el mismo esquema de Verlet.
- **Cálculo de la velocidad del centro de masas:** Se calcula la velocidad del centro de masas para verificar que la simulación se ha realizado correctamente.

5. Implementación de una Simulación de Dinámica Molecular

- **Cálculo de la temperatura:** La temperatura se calcula a partir de (12), luego, asumiendo $k_B = 1$ y $m = 1$, se tiene:

$$T = \frac{1}{3N} \sum_{i=1}^N \|\vec{v}_i\|^2.$$

- **Energía total por partícula:** Finalmente, se calcula la energía total promedio por partícula sumando la energía potencial (obtenida en el algoritmo anterior) y la energía cinética, luego, asumiendo $m = 1$, se tiene:

$$E_{\text{tot}} = \frac{E_{\text{pot}} + \frac{1}{2} \sum \|\vec{v}_i\|^2}{N}.$$

Algoritmo 7 Integrar ecuaciones de movimiento

```

sumv1 ← 0, sumv2, sumv3 ← 0           ▷ Suma de las velocidades
sumv2 ← 0                               ▷ Inicializa la suma de las velocidades al cuadrado
for  $i = 1$  to  $N$  do                   ▷ Bucle principal
     $r_{\text{next}_i} \leftarrow 2\vec{r}_i - r_{\text{prev}_i} + \Delta t^2 \cdot \frac{1}{m_i} \vec{F}_i$            ▷ Nueva posición (Verlet)
     $\vec{v}_i \leftarrow \frac{r_{\text{next}_i} - r_{\text{prev}_i}}{2\Delta t}$                                ▷ Estimación de velocidad
    for  $j = 1$  to  $3$  do
        sumv $j$  ← sumv $j$  +  $\vec{v}_{ij}$            ▷ Suma para calcular velocidad del centro de masas
    end for
    sumv2 ← sumv2 +  $\vec{v}_{i1}^2 + \vec{v}_{i2}^2 + \vec{v}_{i3}^2$            ▷ Actualización velocidades al cuadrado
     $r_{\text{prev}_i} \leftarrow \vec{r}_i$            ▷ Actualización posición anterior
     $\vec{r}_i \leftarrow r_{\text{next}_i}$            ▷ Actualización posición actual
end for
 $\vec{v}_{CM} \leftarrow (\text{sumv}_1/N, \text{sumv}_2/N, \text{sumv}_3/N)$            ▷ Velocidad del centro de masas
Temp ←  $\frac{\text{sumv2}}{3N}$            ▷ Temperatura instantánea
 $E_{\text{tot}} \leftarrow \frac{E_{\text{pot}} + 0.5 \cdot \text{sumv2}}{N}$            ▷ Energía total por partícula

```

La energía total por partícula E_{tot} debería permanecer aproximadamente constante a lo largo de la simulación. Del mismo modo, la velocidad del centro de masas \vec{v}_{CM} debería mantenerse cercana a cero en cada una de sus tres componentes. Cualquier desviación significativa en estos valores podría ser señal de errores en la implementación del algoritmo [15].

5.4. Programas de dinámica molecular

A lo largo de las últimas décadas, se han desarrollado numerosos programas de simulación de dinámica molecular, cada uno con características específicas orientadas a distintos tipos de sistemas, escalas y recursos de cálculo. Estos programas permiten ejecutar simulaciones, analizar trayectorias, visualizar estructuras y aplicar diversos campos de fuerza.

A continuación, se describen algunos de los campos de fuerza más conocidos:

- **AMBER** (*Assisted Model Building with Energy Refinement*): Desarrollado inicialmente para simular ácidos nucleicos y proteínas. Utiliza una formulación bastante precisa de los términos torsionales y electrostáticos. Está especialmente optimizado para sistemas biológicos y se usa ampliamente en bioquímica computacional [49].
- **CHARMM** (*Chemistry at HARvard Macromolecular Mechanics*): Campo de fuerza ampliamente utilizado en simulaciones de proteínas, lípidos y ácidos nucleicos. Está diseñado para trabajar conjuntamente con el programa CHARMM, pero también es compatible con otros paquetes como NAMD o GROMACS. Incluye parámetros específicos para simular membranas biológicas y otros entornos complejos [50].
- **GROMOS** (*GRoningen MOlecular Simulation*): Enfocado inicialmente al modelado de líquidos y biomoléculas, se ha usado tradicionalmente con el paquete de simulación GROMOS. A diferencia de AMBER o CHARMM, GROMOS emplea funciones de potencial ligeramente diferentes, como un término de Van der Waals basado en una forma exponencial en lugar del clásico (4) [51].
- **OPLS** (*Optimized Potentials for Liquid Simulations*): Diseñado para reproducir propiedades termodinámicas de líquidos orgánicos y biomoléculas. Se basa en una parametrización rigurosa contra datos experimentales y simulaciones de química cuántica. Existe una versión OPLS-AA (*All-Atom*) que incluye todos los átomos explícitamente, lo cual mejora la precisión en muchos contextos [52].

Estos modelos se basan en una combinación de términos empíricos (como enlaces, ángulos, torsiones e interacciones de Van der Waals y electrostáticas) y están calibrados mediante datos experimentales y cálculos de química cuántica sobre moléculas pequeñas. Cada uno se desarrolla con un objetivo específico y está calibrado para ciertos tipos de moléculas, por lo que su elección depende del sistema a estudiar y del tipo de simulación que se desee realizar.

Por otro lado, un programa de simulación es un software que implementa los algoritmos necesarios para llevar a cabo la integración temporal, el cálculo de fuerzas y el análisis de trayectorias. Muchos de estos programas permiten la utilización de distintos campos de fuerza externos, lo que les proporciona una gran versatilidad y adaptabilidad a diversos tipos de sistemas moleculares.

Entre los programas de simulación más relevantes se encuentran:

- **AMBER**: Hace referencia tanto a un conjunto de programas como a una familia de campos de fuerza, ya descritos anteriormente. Aunque está especialmente optimizado para sistemas biológicos, sus campos de fuerza pueden exportarse a otros programas como GROMACS.
- **GROMACS**: Código libre y altamente optimizado para realizar simulaciones de MD, especialmente en sistemas grandes como proteínas en membranas. Admite múltiples campos de fuerza (incluyendo AMBER, CHARMM y OPLS), y destaca por su velocidad y eficiencia en arquitecturas paralelas y GPU [47].
- **NAMD**: Diseñado para simular sistemas biomoleculares de gran tamaño, aprovechando recursos computacionales distribuidos (HPC). Compatible con los campos de fuerza CHARMM y AMBER, es ampliamente utilizado en simulaciones a escala masiva [21].

5. Implementación de una Simulación de Dinámica Molecular

- **LAMMPS:** Código versátil orientado a la simulación de materiales y polímeros, pero también capaz de modelar sistemas biológicos. Soporta campos de fuerza clásicos y avanzados, lo que le permite simular fenómenos complejos como la formación y ruptura de enlaces o efectos electrónicos. Además, admite simulaciones a nivel atómico y mesoscópico [53].
- **OpenMM:** Librería orientada a la programación de simulaciones en Python con soporte nativo para GPU. Permite construir simulaciones flexibles, usar campos de fuerza estándar y desarrollar algoritmos personalizados de integración o energía [54].

6. Aplicaciones de la Dinámica Molecular

La dinámica molecular se ha convertido en una herramienta fundamental en numerosos campos científicos y tecnológicos. Su aplicabilidad se extiende a múltiples disciplinas, desde la biología estructural hasta la ciencia de materiales, lo que la convierte en una técnica con gran relevancia en la ciencia computacional moderna [15, 55].

Antes de entrar en las distintas áreas de aplicación, es importante distinguir entre dos grandes tipos de simulaciones: aquellas centradas en la evolución temporal del sistema (como las que estudian transporte, difusión o reacciones dinámicas), y aquellas cuyo objetivo es obtener propiedades de equilibrio, lo que se denomina un estudio termodinámico⁵. En este segundo caso, no interesa cómo cambia el sistema con el tiempo, sino cómo se comporta en promedio, en condiciones de equilibrio. Sin embargo, incluso en estos casos se emplean algoritmos de dinámica molecular, como el integrador de Verlet, con el fin de obtener trayectorias representativas del colectivo estadístico considerado.

6.1. Bioquímica y biología estructural

En esta sección se presentan las principales macromoléculas biológicas y se analiza cómo la dinámica molecular permite estudiar sus propiedades, interacciones y funciones, proporcionando una aproximación computacional que complementa las técnicas experimentales clásicas de la biología estructural.

Para facilitar la comprensión, se parte de una breve descripción de los componentes clave, antes de abordar sus aplicaciones concretas en simulación computacional.

6.1.1. Moléculas biológicas: conceptos básicos

En el contexto biológico, los protagonistas de las simulaciones suelen ser macromoléculas como las siguientes:

Proteínas

Las proteínas son cadenas de aminoácidos que se pliegan en formas tridimensionales específicas. Este plegamiento determina su función biológica, que puede ir desde actuar como enzimas que catalizan reacciones hasta transportar oxígeno o servir como receptores en membranas celulares. Estudiar su dinámica es clave para entender cómo se pliegan, cómo interactúan con otras moléculas, o cómo ciertas mutaciones afectan a su comportamiento [55, 56].

⁵Un estudio termodinámico se centra en calcular propiedades macroscópicas de equilibrio del sistema, sin considerar su evolución temporal.

Ácidos nucleicos (ADN y ARN)

Los ácidos nucleicos son las moléculas encargadas de almacenar y transmitir la información genética. El ADN es conocido por su estructura de doble hélice, pero su conformación puede variar en función del entorno o de su interacción con proteínas. El ARN, además de su papel en la síntesis de proteínas, puede adoptar estructuras tridimensionales complejas con funciones catalíticas o regulatorias [57].

Membranas celulares

Las membranas celulares son estructuras flexibles que rodean y delimitan las células, separando su interior del entorno exterior. Están compuestas principalmente por una doble capa de moléculas similares a grasas, llamadas lípidos, que actúan como una barrera semipermeable. Esta barrera no solo protege la célula, sino que también regula qué sustancias pueden entrar o salir. Mediante simulaciones de dinámica molecular, es posible modelar el comportamiento de estas membranas a nivel atómico y estudiar procesos esenciales como la difusión de pequeñas moléculas, el transporte controlado de sustancias, o cómo ciertas proteínas se insertan y funcionan dentro de la membrana [58].

Complejos biomoleculares

Los complejos biomoleculares son conjuntos de varias moléculas (proteínas, ADN, lípidos) que se ensamblan de forma funcional. Por ejemplo, un ribosoma o un canal iónico. Simular su comportamiento permite comprender mecanismos dinámicos que a menudo no son observables directamente mediante técnicas experimentales [57].

6.1.2. Aplicaciones en simulación biomolecular

Gracias a la MD, es posible investigar numerosos procesos relevantes en biología y bioquímica, que tienen un papel central en el diseño de fármacos. A continuación, se presentan algunas de las aplicaciones más relevantes de la dinámica molecular en este contexto:

Plegamiento de proteínas

El plegamiento es el proceso mediante el cual una cadena lineal de aminoácidos adopta su conformación tridimensional funcional. Las simulaciones se centran en la evolución temporal del sistema, permitiendo explorar este fenómeno paso a paso, identificar intermedios estructurales e incluso estudiar plegamientos erróneos asociados a enfermedades neurodegenerativas como el Alzheimer o el Parkinson [56].

Dinámica estructural

Las proteínas y los ácidos nucleicos no son estructuras rígidas, sino que sufren cambios térmicos y reordenamientos que pueden activar o inhibir sus funciones. La dinámica molecular se centra en la evolución temporal del sistema, permitiendo estudiar estos cambios en función del tiempo y en distintos contextos fisiológicos.

Interacción ligando-receptor

Uno de los usos más relevantes de la dinámica molecular en farmacología es el estudio del acoplamiento molecular, que describe cómo una molécula pequeña (ligando) se une a una proteína diana (receptor). Este análisis se centra en la evolución temporal del sistema, permitiendo predecir afinidades de unión, identificar sitios activos y guiar el diseño racional de fármacos [59].

Estos mecanismos fundamentales son aprovechados conjuntamente en estrategias de descubrimiento de fármacos:

Diseño de fármacos y descubrimiento molecular

Uno de los usos más avanzados de la dinámica molecular en biomedicina es el diseño racional de fármacos. Esta estrategia, conocida como *drug discovery in silico*, consiste en predecir mediante simulaciones qué compuestos podrían unirse eficazmente a una proteína implicada en una enfermedad, la cual actúa como diana biológica. A diferencia de los métodos tradicionales de cribado experimental, que requieren analizar físicamente miles de moléculas, la simulación permite filtrar virtualmente grandes bibliotecas de ligandos, evaluando su interacción con dianas biológicas en función de propiedades estructurales, energéticas y dinámicas [56, 59].

Una técnica ampliamente utilizada en el diseño de fármacos es el *docking*, que consiste en predecir computacionalmente cómo se une un ligando a un receptor. El objetivo es determinar la orientación y posición más probable del ligando dentro del sitio activo del receptor, evaluando su ajuste espacial y la energía asociada a la interacción. Sin embargo, este procedimiento suele asumir modelos rígidos, en los que tanto el ligando como el receptor permanecen estáticos durante el proceso de acoplamiento. Esta simplificación limita la capacidad del *docking* para capturar la flexibilidad estructural de las moléculas y la complejidad real del entorno molecular.

Al combinar *docking* con simulaciones de dinámica molecular, es posible refinar las predicciones iniciales y estudiar cómo evoluciona la interacción a lo largo del tiempo. Este enfoque combinado permite detectar inestabilidades en la unión, explorar conformaciones alternativas del complejo y validar si la interacción predicha se mantiene estable en un entorno solvado y dinámico [57]. Esto es especialmente útil para reducir falsos positivos, es decir, aquellos casos en los que una predicción inicial sugiere una buena afinidad entre un ligando y una proteína, pero que en realidad no resulta estable cuando se simula con más realismo.

Una vez identificadas moléculas con actividad prometedora frente a una diana biológica, estas se consideran compuestos líderes, los cuales representan puntos de partida clave en el proceso de desarrollo de fármacos.

Las simulaciones de dinámica molecular permiten analizar en detalle cómo interactúa un compuesto líder con su diana, observando los contactos atómicos, los ajustes estructurales y la estabilidad de la unión. A partir de esta información, es posible diseñar modificaciones estructurales (por ejemplo, añadir o sustituir grupos químicos específicos) para mejorar su eficacia, la selectividad (es decir, que actúe únicamente sobre la diana deseada) y otras

propiedades farmacológicas como la solubilidad o la estabilidad en el organismo. Este proceso, conocido como optimización del compuesto líder, es esencial para avanzar desde un candidato preliminar hasta un fármaco potencial.

Finalmente, esta estrategia ayuda a seleccionar candidatos farmacológicos, es decir, moléculas que, tras un filtrado computacional riguroso, presentan mayor probabilidad de éxito en etapas posteriores como los ensayos preclínicos o clínicos. Este enfoque ha sido aplicado con éxito al desarrollo de terapias frente a enfermedades como el cáncer, el VIH, trastornos neurodegenerativos o infecciones virales como el SARS-CoV-2 [60].

En los últimos años, la combinación de MD con modelos de inteligencia artificial ha impulsado aún más el diseño computacional de fármacos. Modelos basados en redes neuronales, como ANI [61] o DeepMD [62], permiten representar funciones de energía aprendidas a partir de cálculos cuánticos, alcanzando una precisión similar pero con un coste computacional mucho menor [63]. Estas herramientas permiten generar nuevos compuestos, predecir afinidades de unión o identificar patrones relevantes directamente a partir de grandes bases de datos moleculares.

6.2. Materiales y nanotecnología

La dinámica molecular ha adquirido un papel fundamental en el campo de la ciencia de materiales, al permitir simular el comportamiento de materiales a nivel atómico. En lugar de describir los materiales mediante propiedades medias o macroscópicas, la MD permite observar directamente cómo se mueven los átomos, cómo se rompen enlaces o cómo se forman defectos internos bajo diferentes condiciones. Esto es especialmente útil en contextos donde los métodos tradicionales (como modelos continuos o ensayos experimentales) no permiten acceder a escalas nanométricas o tiempos extremadamente cortos. Esta técnica proporciona una herramienta eficaz para investigar la estructura interna de sólidos cristalinos, materiales amorfos, polímeros o aleaciones metálicas [26].

Simulación de propiedades de materiales

Mediante simulaciones de dinámica molecular es posible determinar muchas de las propiedades físicas clave de un material. En este contexto, el objetivo principal no es estudiar la evolución temporal del sistema, sino realizar un análisis termodinámico que permita calcular magnitudes a partir del comportamiento promedio del sistema en equilibrio. Por ejemplo, al simular una red cristalina, como la del cobre o el silicio, es posible aplicar una deformación virtual y observar cómo responden los átomos que la componen. A partir de esta respuesta, se pueden calcular magnitudes relevantes como el módulo de elasticidad, que mide la rigidez del material y cuantifica cuánto se deforma ante una tensión aplicada; la energía de cohesión, que representa la energía necesaria para separar los átomos del sólido y refleja la estabilidad interna de la estructura; y el límite elástico, que señala el punto a partir del cual el material deja de comportarse de forma elástica y comienza a sufrir deformaciones permanentes. Estas propiedades permiten caracterizar el comportamiento mecánico del material a nivel atómico, algo fundamental para el diseño y la optimización de nuevos compuestos.

Además, se pueden estudiar defectos como dislocaciones (fallos lineales en la estructura), vacantes (átomos que faltan) o intersticiales (átomos extra entre posiciones regulares), y ana-

lizar cómo estos afectan a la resistencia mecánica, la difusión de átomos o la conductividad térmica [64].

Otra aplicación importante es el estudio de procesos de fractura y fatiga. Simulando un material con grietas o sometido a tensiones cíclicas, se puede observar cómo se propagan fallos internos, lo cual es muy útil para diseñar materiales más resistentes o duraderos.

Modelado de superficies e interfaces

Muchas aplicaciones industriales y tecnológicas implican materiales con superficies o interfaces (por ejemplo, un recubrimiento sobre una pieza metálica). La MD permite simular qué ocurre en esas regiones de frontera, es decir, cómo se adsorben moléculas sobre una superficie, cómo se produce el crecimiento de capas de material (crecimiento epitaxial) o cómo se comportan los materiales en contacto cuando hay diferencias de estructura o composición [65].

Este tipo de modelado es especialmente importante en áreas como la electrónica, donde se usan capas finas de materiales con propiedades específicas; en la nanofabricación, donde se necesita entender procesos de deposición y litografía a escala atómica; o en fenómenos como la corrosión y la adhesión, en los que se simula cómo interactúan los materiales con el entorno o con otros sólidos.

Nanotecnología y materiales avanzados

La nanotecnología se basa en diseñar y manipular materiales a escala nanométrica (1 nanómetro = 10^{-9} metros). En este contexto, la dinámica molecular permite modelar estructuras novedosas llamadas nanomateriales, que pueden tener propiedades muy diferentes a sus contrapartes macroscópicas.

Algunos ejemplos importantes son:

- **Nanotubos de carbono (CNTs):** estructuras cilíndricas formadas por átomos de carbono, extremadamente resistentes y ligeras. Con MD se puede estudiar su flexibilidad, resistencia mecánica o capacidad de conducción térmica.
- **Fullerenos y nanocápsulas:** esferas de carbono que pueden encapsular otras moléculas (como fármacos), y cuya estabilidad y reactividad se pueden analizar por simulación.
- **Grafeno y materiales 2D:** láminas de un átomo de grosor con propiedades electrónicas y mecánicas excepcionales. La MD permite estudiar cómo vibran sus átomos (fonones), cómo se comporta térmicamente o cómo se deforma bajo tensión [66].

Además, la dinámica molecular permite modelar fenómenos como el autoensamblaje de nanopartículas, es decir, cómo ciertas moléculas se organizan espontáneamente formando estructuras útiles. Estos fenómenos son clave en el diseño de nanodispositivos aplicados a la medicina personalizada.

La interacción entre nanopartículas y sistemas biológicos (por ejemplo, nanopartículas diseñadas para transportar fármacos) también ha sido ampliamente estudiada mediante dinámica molecular, permitiendo predecir aspectos clave como la toxicidad, la absorción celular

6. Aplicaciones de la Dinámica Molecular

o la compatibilidad de los materiales utilizados en nanomedicina. Gracias a su capacidad para incorporar entornos realistas (como agua, membranas o iones), la dinámica molecular permite analizar estas interacciones con un nivel de detalle que complementa y profundiza la información obtenida experimentalmente. [67].

En resumen, la dinámica molecular aporta una herramienta versátil para explorar y diseñar materiales desde su estructura atómica, anticipando su comportamiento real incluso antes de fabricarlos físicamente. Esta capacidad resulta especialmente útil en contextos de investigación aplicada, innovación tecnológica y desarrollo de nuevos materiales con propiedades a medida.

6.3. Química y física de fluidos

La dinámica molecular también se emplea para estudiar el comportamiento de fluidos a escala molecular, es decir, considerando directamente la interacción entre las moléculas individuales que componen el líquido o gas. En este tipo de simulaciones, el objetivo principal es caracterizar el sistema en equilibrio, por lo que se centran en el estudio termodinámico del sistema, permitiendo calcular propiedades macroscópicas como la viscosidad (resistencia al flujo), el coeficiente de difusión (cómo se dispersan las moléculas en el medio), la tensión superficial (energía que se requiere para aumentar la superficie de un líquido) o la conductividad térmica (capacidad del fluido para transferir calor) [15].

Esta forma de simular resulta especialmente útil cuando los modelos tradicionales, que tratan los fluidos como medios continuos, no son suficientes para describir correctamente el sistema. Esto ocurre, por ejemplo, cuando se trabaja a escalas muy pequeñas (nanométricas), donde las propiedades del fluido dependen directamente del comportamiento individual de las moléculas. En estos casos, tener en cuenta la naturaleza discreta de la materia y las interacciones entre partículas es esencial para obtener resultados realistas [68].

Un ejemplo claro es el caso de los líquidos complejos, las soluciones iónicas (como la sal disuelta en agua), los fluidos confinados en canales muy estrechos o las mezclas con muchos tipos de moléculas. En todos ellos, las interacciones específicas entre moléculas pueden dar lugar a efectos que los modelos continuos no predicen, como patrones de organización local, separación de fases o transporte anómalo.

Por otro lado, la dinámica molecular también permite estudiar reacciones químicas en medios líquidos utilizando enfoques híbridos, como el método QM/MM (*Quantum Mechanics/Molecular Mechanics*). En este tipo de simulaciones, una parte del sistema (normalmente donde tiene lugar la reacción) se modela con métodos de química cuántica, mientras que el entorno restante se simula con dinámica clásica (ver [Subsección 7.2.1](#)). Este tipo de técnica ha sido utilizado con éxito en el estudio de mecanismos enzimáticos, reacciones ácido-base en disolución o procesos de transferencia electrónica [33, 56].

6.4. Estudios de energía y catalizadores

La dinámica molecular también se utiliza ampliamente en el estudio de tecnologías relacionadas con la producción, almacenamiento y conversión de energía. En particular, se ha aplicado

al modelado de dispositivos como baterías de ion-litio, supercondensadores y celdas de combustible [29]. Estos sistemas dependen críticamente de cómo se mueven los iones (átomos cargados) dentro del material, de cómo interactúan con los electrodos y de los fenómenos que ocurren en las interfaces entre diferentes fases (por ejemplo, entre un sólido y un líquido).

Mediante simulaciones de dinámica molecular es posible observar estos procesos a nivel atómico, proporcionando información sobre la velocidad a la que se difunden los iones, cómo afectan las impurezas o defectos estructurales al transporte iónico, o cómo varía la estructura del material bajo diferentes condiciones de carga o temperatura. Esto es fundamental para diseñar baterías más eficientes y duraderas, optimizar la estabilidad térmica de los materiales activos o mejorar la conductividad de los electrolitos [64].

Otro campo muy importante en el que se aplica la dinámica molecular es la catálisis, es decir, el estudio de materiales que aceleran las reacciones químicas sin consumirse en el proceso. En catálisis heterogénea (donde el catalizador y los reactivos están en fases distintas, como un sólido y un gas), la MD permite estudiar cómo se adsorben los reactivos sobre la superficie del catalizador, cómo cambian de posición y reaccionan, y cómo se liberan los productos. Estas simulaciones ayudan a entender la estructura y dinámica de los llamados "sitios activos", que son las regiones del material donde ocurren las reacciones [69].

En el caso de la catálisis homogénea (donde el catalizador está disuelto en el mismo medio que los reactivos), la dinámica molecular también es útil para analizar el entorno solvatado⁶ del catalizador y cómo influye en su reactividad. Combinada con métodos cuánticos (como el enfoque QM/MM), se pueden simular directamente las transformaciones químicas que ocurren durante la catálisis [33, 56].

⁶Proceso por el cual las moléculas del disolvente (por ejemplo, agua) rodean a una especie química disuelta, formando una capa de solvatación. Estas interacciones afectan a la estructura, estabilidad y reactividad de dicha especie.

7. Desafíos y futuras direcciones

A pesar de su consolidación como herramienta fundamental en la investigación computacional, la dinámica molecular continúa enfrentando importantes desafíos, tanto desde el punto de vista metodológico como computacional. Aunque en las últimas décadas se han logrado avances significativos en técnicas numéricas robustas y en la capacidad computacional disponible, sigue teniendo limitaciones que restringen su aplicabilidad en escenarios más complejos o a escalas mayores. Estas limitaciones, sumadas al crecimiento exponencial de los recursos de hardware y a los recientes avances en algoritmos inteligentes, han abierto nuevas líneas de desarrollo con el objetivo de ampliar la aplicabilidad y mejorar la precisión de esta técnica. En este capítulo se presentan dichas restricciones, se identifican tendencias emergentes en el campo y se exponen diversas oportunidades de mejora que podrían marcar la evolución futura de la MD.

7.1. Limitaciones actuales

A continuación, se describen algunas de las principales limitaciones actuales, tanto desde el punto de vista computacional como metodológico:

7.1.1. Escalabilidad

Una de las principales limitaciones de la dinámica molecular clásica es su limitada escalabilidad computacional. Si bien los métodos de paralelización y el uso de arquitecturas GPU han permitido avances significativos en rendimiento, las simulaciones a gran escala siguen estando condicionadas por el elevado coste computacional asociado al cálculo de fuerzas y a la integración temporal. Esto se vuelve especialmente restrictivo cuando se desea simular sistemas que contienen millones de átomos o que requieren tiempos de simulación del orden de micro o milisegundos [15, 46].

Este cuello de botella impide que muchos fenómenos de interés, como el plegamiento de proteínas, puedan ser simulados con suficiente resolución. Además, incluso en arquitecturas de alto rendimiento, la comunicación entre procesos puede convertirse en un factor limitante, especialmente cuando se emplean esquemas de descomposición espacial con frecuentes intercambios de información entre nodos.

7.1.2. Precisión de los modelos de fuerza

Otra limitación crítica proviene de la precisión de los modelos de interacción utilizados, los conocidos campos de fuerza. Los campos de fuerza clásicos, como AMBER, CHARMM o OPLS, se basan en parametrizaciones empíricas que, si bien son eficientes, pueden carecer de precisión cuando se aplican a entornos no estándar, como interfaces complejas, metales o moléculas sintéticas, ya que en estos casos pueden darse tipos de enlace o interacciones que no están correctamente representados en el modelo original [29]. Esta aproximación limita

7. Desafíos y futuras direcciones

la transferibilidad de los modelos y su capacidad para predecir comportamientos fuera del dominio de entrenamiento.

Además, estos campos de fuerza no incorporan efectos cuánticos explícitos, lo que puede ser determinante en procesos donde se produce formación o ruptura de enlaces, transferencia de carga o reorganización electrónica [33]. Por tanto, la calidad de los resultados obtenidos depende en gran medida de la adecuación del modelo al problema específico.

7.1.3. Tiempo de simulación

Otro obstáculo importante es la restricción temporal de las simulaciones. Los algoritmos de integración requieren pasos de tiempo muy pequeños (del orden de los femtosegundos) para preservar la estabilidad numérica y resolver adecuadamente los modos vibracionales más rápidos del sistema (ver Figura 10). Como consecuencia, para alcanzar escalas del orden de microsegundos o más, se deben realizar millones de pasos, lo que incrementa notablemente el coste computacional [56].

Esta limitación impide simular procesos lentos como el plegamiento de proteínas, la difusión en materiales sólidos o los mecanismos de señalización celular. Muchas de estas interacciones clave ocurren en escalas temporales inalcanzables para la dinámica molecular convencional, lo que requiere el uso de técnicas complementarias como simulaciones multiescala, modelos acelerados o estrategias de muestreo mejorado [15].

7.2. Nuevas tendencias

Frente a estas limitaciones, se están explorando múltiples líneas de avance, entre las cuales destacan las siguientes:

7.2.1. Métodos híbridos clásico-cuánticos (QM/MM)

Una de las estrategias más eficaces para estudiar sistemas complejos donde intervienen tanto procesos mecánicos clásicos como fenómenos cuánticos es la combinación de dinámica molecular clásica con métodos de estructura electrónica. Esta aproximación, conocida como QM/MM (*Quantum Mechanics/Molecular Mechanics*), permite tratar una pequeña región del sistema, donde ocurren procesos esencialmente cuánticos como reacciones químicas, ruptura de enlaces o transferencia electrónica, mediante mecánica cuántica, mientras que el resto del entorno (por ejemplo, el disolvente o la proteína circundante) se modela utilizando potenciales clásicos menos costosos computacionalmente [33, 70, 71].

Sin embargo, conectar estas dos regiones no es trivial. Uno de los principales retos es lograr una transición suave en la zona de frontera entre la región cuántica y la clásica, de forma que no aparezcan efectos no realistas o inconsistencias en las fuerzas. Para conseguirlo, se han desarrollado varias técnicas. Por ejemplo, los llamados *átomos enlace* se utilizan cuando un enlace químico cruza la frontera QM/MM, simulando artificialmente una terminación para mantener la validez del cálculo cuántico. También existen métodos de *embebido electrostático*, en los que las cargas del entorno clásico afectan directamente al cálculo cuántico para simular la influencia del entorno.

Finalmente, en los enfoques de *embedding polarizable*, se permite que la región clásica también responda a los cambios en la densidad electrónica de la región cuántica, simulando un acoplamiento bidireccional más realista [71, 72].

Desde el punto de vista computacional, este método permite llevar a cabo simulaciones más realistas de sistemas complejos sin asumir el elevado coste computacional que implicaría un tratamiento cuántico completo. Actualmente, múltiples paquetes de simulación como NAMD o GROMACS soportan implementaciones híbridas, combinando de forma eficiente los cálculos de energía y fuerzas para ambas regiones [73, 74].

Por tanto, el enfoque QM/MM es una herramienta muy útil para estudiar procesos moleculares a diferentes escalas, tanto en sistemas biológicos como en materiales, proporcionando un equilibrio entre precisión y eficiencia computacional. Su uso en simulaciones modernas apunta a ser una de las formas más prometedoras de hacer que la dinámica molecular sea aplicable a situaciones más realistas.

7.2.2. Aprendizaje automático para generación de campos de fuerza

En los últimos años, la combinación de dinámica molecular con técnicas de inteligencia artificial y aprendizaje automático está revolucionando el campo [63]. Estas herramientas están permitiendo abordar muchas de las limitaciones tradicionales de la dinámica molecular, gracias a su capacidad para aprender patrones complejos a partir de grandes volúmenes de datos y generalizar a nuevos escenarios.

Entre sus principales aplicaciones destacan:

- La predicción eficiente de superficies de energía potencial, lo que permite evitar el cálculo explícito de fuerzas mediante métodos cuánticos costosos.
- La generación automática de campos de fuerza específicos mediante técnicas de aprendizaje supervisado o redes neuronales profundas.
- La aceleración del muestreo del espacio de fases y la detección de eventos poco frecuentes, como transiciones conformacionales o procesos de nucleación [60].

Una de las líneas más prometedoras en este ámbito son los *neural network potentials*, funciones de energía entrenadas sobre datos cuánticos de alta fidelidad, que permiten alcanzar precisiones cercanas a la química cuántica a un coste computacional mucho menor. Modelos avanzados como ANI o DeepMD, que se tratarán más adelante, han demostrado una notable capacidad para reproducir propiedades moleculares y dinámicas complejas, lo que los convierte en herramientas clave para simulaciones de gran precisión [63].

7.2.3. Simulaciones multiescala y a gran escala

Otra tendencia creciente en el campo de la dinámica molecular es la simulación multiescala, que busca combinar diferentes niveles de descripción para modelar sistemas complejos de manera eficiente. Este enfoque permite acoplar modelos atomísticos de alta resolución con representaciones más simplificadas o *coarse-grained*, manteniendo el detalle en las regiones de interés sin caer en un coste computacional prohibitivo.

7. Desafíos y futuras direcciones

El objetivo principal es conectar distintas escalas para poder simular sistemas de gran tamaño o procesos que ocurren en escalas de tiempo largas, como el plegamiento de proteínas, la difusión de macromoléculas o fenómenos interfaciales. Esta técnica está siendo ampliamente utilizada en áreas como el diseño de materiales, la biología estructural y el estudio de superficies y membranas [26, 75].

7.3. Oportunidades de mejora

La dinámica molecular está evolucionando a gran velocidad, gracias a mejoras en los métodos, la tecnología y los enfoques teóricos. Esta sección explora algunas de las direcciones más prometedoras para mejorar la precisión, eficiencia y aplicabilidad de las simulaciones en ciencia de materiales, biología computacional y diseño molecular.

7.3.1. Nuevos algoritmos e integradores simplécticos

Como respuesta al elevado coste computacional de simular sistemas complejos durante largos periodos de tiempo, se han desarrollado algoritmos de integración multiescala, entre los que destaca el método RESPA (*REference System Propagator Algorithm*) [76]. Este enfoque se basa en descomponer las fuerzas que actúan en el sistema según su escala temporal: las fuerzas que varían rápidamente (como las vibraciones internas de una molécula) se integran con pasos de tiempo pequeños, mientras que las fuerzas más lentas (como las interacciones de largo alcance) se actualizan con menor frecuencia, utilizando pasos más grandes.

Gracias a esta división jerárquica, RESPA permite reducir significativamente el número total de evaluaciones de fuerza sin comprometer la precisión ni la estabilidad numérica, lo que lo convierte en una herramienta especialmente eficaz en sistemas que presentan dinámicas en múltiples escalas temporales, como proteínas en disolución o materiales con estructuras complejas.

Además de estos esquemas jerárquicos, se están desarrollando algoritmos de integración adaptativos, que ajustan dinámicamente ciertos parámetros de la simulación, como el tamaño del paso de integración, en función del comportamiento local del sistema. Por ejemplo, si los átomos se mueven lentamente, el algoritmo puede aumentar el paso de integración para reducir el tiempo de simulación; si detecta interacciones rápidas o colisiones, lo reduce automáticamente para mantener la precisión y evitar errores numéricos. Gracias a este tipo de integración adaptativa, se puede optimizar el uso de recursos computacionales sin perder fiabilidad en los resultados.

Por otro lado, se están desarrollando algoritmos de integración de orden superior frente a los esquemas clásicos vistos anteriormente. Estos nuevos métodos utilizan más información (por ejemplo, derivadas de orden superior) para mejorar la precisión sin necesidad de usar pasos de tiempo tan pequeños. Aunque su coste computacional por paso es mayor, permiten emplear pasos de integración más grandes para alcanzar la misma precisión, lo que puede traducirse en una mayor eficiencia a largo plazo.

Ambos enfoques se benefician del uso de arquitecturas modernas de computación paralela, como GPUs o clústeres multinúcleo. Esto permite distribuir los cálculos más pesados y complejos de forma eficiente, reduciendo significativamente los tiempos de simulación. Gracias

a estas mejoras, la dinámica molecular puede usarse en sistemas más grandes y en procesos de mayor duración [15, 28].

7.3.2. Avances en el diseño de campos de fuerza

Tradicionalmente, como se presentó en el [Capítulo 5](#), los campos de fuerza más utilizados en bioquímica y ciencia de materiales han sido AMBER, CHARMM, OPLS o GROMOS. Estos modelos, aunque han demostrado ser eficaces en muchos contextos, presentan limitaciones cuando se aplican a entornos no estándar, como sistemas metálicos, materiales con estructuras poco convencionales, interfaces complejas o moléculas sintéticas. En estos casos, la precisión de las simulaciones puede disminuir notablemente debido a la falta de parámetros adecuados o a la incapacidad del modelo para capturar efectos cuánticos sutiles [29].

Para superar estas limitaciones, en los últimos años se han desarrollado campos de fuerza de nueva generación basados en datos obtenidos mediante química cuántica y técnicas de aprendizaje automático. Estos enfoques, conocidos como *machine-learned potentials*, buscan aproximar con precisión la superficie de energía potencial de un sistema sin depender de expresiones funcionales predefinidas, sino aprendiendo directamente de grandes conjuntos de datos cuánticos.

Dos ejemplos destacados de esta nueva generación de campos de fuerza son los modelos ANI (*Accurate Neural Network Potentials*) y DeepMD. Ambos combinan el aprendizaje automático con datos de alta precisión provenientes de cálculos de química cuántica para modelar con mayor fidelidad las interacciones atómicas.

El modelo ANI [61] utiliza redes neuronales entrenadas con miles de cálculos realizados mediante *Density Functional Theory* (DFT)⁷. Esta técnica proporciona energías y fuerzas de referencia con alta precisión, lo que permite a la red aprender una aproximación muy cercana a la verdadera superficie de energía potencial. Gracias a su arquitectura y a la forma en la que se representan las moléculas (usando descriptores rotacional y traslacionalmente invariantes), ANI puede generalizar a moléculas que no estaban en el conjunto de entrenamiento, manteniendo una buena precisión energética. Esto lo hace especialmente útil en simulaciones biomoleculares, farmacológicas o de materiales orgánicos.

Por otro lado, el modelo DeepMD [62] se centra en extender esta idea a materiales más diversos, incluidos sólidos, líquidos y sistemas complejos como aleaciones metálicas o superficies. A diferencia de ANI, DeepMD representa las configuraciones atómicas directamente como entradas a una red neuronal profunda que respeta las simetrías físicas del sistema (traslación, rotación y permutación). Esta red es capaz de reproducir con precisión las energías, fuerzas y tensores de estrés obtenidos de cálculos *ab initio*, permitiendo realizar simulaciones de dinámica molecular con calidad cuántica pero a un coste computacional similar al de los modelos clásicos.

Este tipo de enfoques híbridos, que combinan datos obtenidos mediante cálculos cuánticos con técnicas de aprendizaje automático, permiten aumentar notablemente la precisión y transferibilidad de los campos de fuerza, superando muchas de las limitaciones de los modelos clásicos cuando se enfrentan a entornos químicos no estándar [63].

⁷La teoría del funcional de la densidad (DFT) es un enfoque computacional de la mecánica cuántica que permite calcular la energía de un sistema de muchos electrones a partir únicamente de su densidad electrónica, en lugar de usar la función de onda completa, lo cual reduce considerablemente el coste computacional [77].

8. Conclusiones

Este trabajo se ha centrado en el estudio de la dinámica molecular, abarcando desde sus fundamentos teóricos hasta sus principales aplicaciones, profundizando en los distintos algoritmos empleados en las simulaciones.

En primer lugar, destacar la utilidad e importancia que tiene la dinámica molecular en el estudio de sistemas atómicos y moleculares con muchos grados de libertad. Su capacidad para describir la evolución temporal de sistemas complejos bajo distintas condiciones termodinámicas la convierte en una técnica clave para calcular propiedades estructurales, dinámicas y energéticas que son difíciles de obtener de forma experimental.

Del estudio teórico realizado se concluye que uno de los aspectos clave para obtener una simulación fiable es la correcta elección del integrador numérico, junto con un paso de integración adecuado, ya que ambos influyen directamente en la estabilidad del algoritmo. Los métodos simplécticos, como el algoritmo de Verlet y sus variantes, garantizan una buena conservación de la energía a largo plazo siempre que el paso de integración sea apropiado, lo que resulta fundamental para evitar inestabilidades numéricas. Del mismo modo, tanto el campo de fuerzas como la representación empleada para describir el sistema son factores críticos, ya que influyen significativamente en el equilibrio entre la precisión y el coste computacional de la simulación.

Además, en el trabajo se evidencia la importancia de hacer uso de mecanismos de control como termostatos y barostatos, ya que sin ellos no sería posible simular muchos de los escenarios que se dan en condiciones reales. No obstante, su aplicación debe realizarse con precaución, pues pueden alterar las propiedades estadísticas del sistema y comprometer la fiabilidad de la simulación. Asimismo, se ha destacado la importancia de llevar a cabo una adecuada etapa de minimización de energía antes de iniciar la simulación, la cual permite eliminar configuraciones artificiales no físicas y partir de un estado estable. En sistemas complejos, donde la superficie de energía presenta múltiples mínimos locales, se ha destacado el uso de metaheurísticas como complemento a los métodos deterministas tradicionales, especialmente cuando se busca aproximarse al mínimo global.

Luego, se ha expuesto la utilidad de emplear unidades reducidas en simulaciones de dinámica molecular, ya que permiten simplificar considerablemente las expresiones físicas y evitar errores derivados de conversiones entre sistemas de unidades. Además, mejoran la estabilidad numérica y la generalización del código, debido a que los resultados pueden reescalarse posteriormente a unidades físicas reales si se desea. El uso de unidades reducidas facilita también la comparación entre diferentes simulaciones y modelos. Por todo esto, representa una práctica habitual en simulaciones de dinámica molecular.

Por otro lado, se ha mostrado que la dinámica molecular tiene aplicaciones de gran relevancia en numerosas disciplinas científicas y tecnológicas. Entre ellas, destaca el diseño racional de fármacos, donde permite estudiar con detalle la interacción entre moléculas y

8. Conclusiones

sus posibles dianas terapéuticas. Esta capacidad resulta clave en la búsqueda de tratamientos para enfermedades complejas como el cáncer, el VIH o los trastornos neurodegenerativos, cuya cura representaría, sin duda, un hito histórico en la medicina. Además, la dinámica molecular también se aplica en campos como la ciencia de materiales, la nanotecnología o la biología estructural.

Finalmente, se ponen de manifiesto las limitaciones actuales de la dinámica molecular, entre las que destacan la limitada escalabilidad computacional, la falta de precisión de los modelos de campos de fuerza clásicos para describir fenómenos de naturaleza cuántica y los elevados tiempos de simulación requeridos por ciertos experimentos. Frente a estas barreras, se presentan líneas de investigación abiertas y prometedoras, como el uso de inteligencia artificial, la integración de métodos multiescala o el aprovechamiento de arquitecturas de alto rendimiento, que reflejan el gran potencial de la dinámica molecular para seguir evolucionando.

A. Código

Para apoyar la explicación de la simplecticidad y de la influencia del valor del paso de integración Δt en la estabilidad del integrador, se han implementado dos scripts:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # Parámetros del sistema
5 dt = 0.02          # paso de tiempo
6 tend = 20         # tiempo de finalización de la simulación
7 n = int(tend / dt) # números de iteraciones en la simulación
8 x0, v0 = 0.0, 1.0 # posición y velocidades iniciales
9 k, m = 1.0, 1.0   # constante de boltzmann y masa del cuerpo
10
11 # Vector de tiempos
12 tvec = np.linspace(0, tend, n)
13
14 # Algoritmo de Verlet
15 x_verlet = np.zeros(n) # vector de posición
16 v_verlet = np.zeros(n) # vector de velocidad
17 x_verlet[0] = x0       # posición inicial
18 v_verlet[0] = v0       # velocidad inicial
19 x_prev = x0 - v0 * dt + 0.5 * (-k * x0 / m) * dt**2 # x(-dt)
20
21 # Integración con Verlet (posición)
22 for i in range(0, n - 1):
23     force = -k * x_verlet[i]
24     x_next = 2 * x_verlet[i] - x_prev + (dt**2 / m) * force
25     x_prev = x_verlet[i]
26     x_verlet[i + 1] = x_next
27
28 # Cálculo de velocidades con Verlet
29 for i in range(1, n - 1):
30     v_verlet[i] = (x_verlet[i + 1] - x_verlet[i - 1]) / (2 * dt)
31
32 # Recorte para espacio de fases, ya que v[n] no se calcula
33 x_verlet_cut = x_verlet[:-1]
34 v_verlet_cut = v_verlet[:-1]
35
36 # Algoritmo de Euler
37 x_euler = np.zeros(n) # vector de posición
38 v_euler = np.zeros(n) # vector de velocidad
39 x_euler[0] = x0       # posición inicial
40 v_euler[0] = v0       # velocidad inicial
41
42 # Cálculo de posiciones y velocidades
43 for i in range(n - 1):
```

A. Código

```
44     x_euler[i + 1] = x_euler[i] + v_euler[i] * dt
45     v_euler[i + 1] = v_euler[i] - k * x_euler[i] * dt
46
47 # Solución exacta
48 x_exact = v0 * np.sin(tvec) + x0 * np.cos(tvec)
49
50 # Gráfica conjunta del algoritmo de Verlet
51 fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))
52
53 # Posición
54 ax1.plot(tvec, x_verlet, label='Verlet', linewidth=2)
55 ax1.plot(tvec, x_exact, '--', label='Exacta', linewidth=2)
56 ax1.set_title(f"Algoritmo de Verlet (dt = {dt})")
57 ax1.set_xlabel("Tiempo")
58 ax1.set_ylabel("Posición")
59 ax1.legend()
60 ax1.grid(True)
61
62 # Espacio de fases
63 ax2.plot(x_verlet_cut, v_verlet_cut, color='tab:blue')
64 ax2.set_xlabel("Posición")
65 ax2.set_ylabel("Momento")
66 ax2.set_title("Verlet - Espacio de fases")
67 ax2.axis("equal")
68 ax2.grid(True)
69
70 plt.tight_layout(rect=[0, 0.03, 1, 0.95])
71 plt.show()
72
73 # Gráfica conjunta del algoritmo de Euler
74 fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))
75
76 # Posición
77 ax1.plot(tvec, x_euler, label='Euler', linewidth=2)
78 ax1.plot(tvec, x_exact, '--', label='Exacta', linewidth=2)
79 ax1.set_title(f"Algoritmo de Euler (dt = {dt})")
80 ax1.set_xlabel("Tiempo")
81 ax1.set_ylabel("Posición")
82 ax1.legend()
83 ax1.grid(True)
84
85 # Espacio de fases
86 ax2.plot(x_euler, v_euler, color='tab:red')
87 ax2.set_xlabel("Posición")
88 ax2.set_ylabel("Momento")
89 ax2.set_title("Euler - Espacio de fases")
90 ax2.axis("equal")
91 ax2.grid(True)
92
93 plt.tight_layout(rect=[0, 0.03, 1, 0.95])
94 plt.show()
```

Código A.1: Script para reflejar la diferencia entre los algoritmos simplécticos y no simplécticos. Para ello, se ha simulado un oscilador armónico con el método de Euler (no simpléctico) y con el de Verlet (simpléctico). En ambas simulaciones se muestra la evolución temporal de la posición y la trayectoria en el espacio de fases obtenidas.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def verlet_simulation(ax, dt, t_end=20, k=1.0, m=1.0, x0=0.0, v0=1.0):
5     n = int(t_end / dt)          # números de iteraciones en la
6         simulación
7     x = np.zeros(n)             # vector de posición
8     v = np.zeros(n)             # vector de velocidad
9     tvec = np.linspace(0, t_end, n) # vector de tiempos
10
11     # Condiciones iniciales
12     x[0] = x0
13     x_prev = x0 - v0 * dt + 0.5 * (-k * x0 / m) * dt**2          # x(-dt)
14
15     # Integración con Verlet (posición)
16     for i in range(0, n - 1):
17         force = -k * x[i]
18         x_next = 2 * x[i] - x_prev + (dt**2 / m) * force
19         x_prev = x[i]
20         x[i + 1] = x_next
21
22     # Solución exacta
23     x_exact = x0 * np.cos(tvec) + v0 * np.sin(tvec)
24
25     # Gráfico
26     ax.plot(tvec, x, label="Verlet")
27     ax.plot(tvec, x_exact, '--', label="Exacta")
28     ax.set_xlabel("Tiempo")
29     ax.set_ylabel("Posición")
30     ax.set_title(f"dt = {dt}")
31     ax.legend()
32     ax.grid(True)
33
34 # Crear una figura con 2 subplots
35 fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8))
36
37 # Ejecutar para dt = 0.02 y dt = 2
38 verlet_simulation(ax1, dt=0.02)
39 verlet_simulation(ax2, dt=2)
40
41 plt.tight_layout(rect=[0, 0.03, 1, 0.95])
42 plt.show()

```

A. Código

Código A.2: Script para comprobar la estabilidad del algoritmo de Verlet para distintos valores del paso de integración Δt . Para ello, se ha simulado un oscilador armónico y se ha mostrado la evolución temporal obtenida de la posición para $\Delta t = 0.02$ y para $\Delta t = 2$.

El código mostrado se puede encontrar en el siguiente repositorio:

https://github.com/Joarpeoz/TFG_DinamicaMolecular.git

Glosario

Adhesión Capacidad de dos materiales diferentes para mantenerse unidos en una interfaz común, debido a interacciones físicas (como fuerzas de Van der Waals) o químicas (como enlaces covalentes o iónicos).

Aminoácido Molécula orgánica que contiene un grupo amino ($-\text{NH}_2$), un grupo carboxilo ($-\text{COOH}$) y una cadena lateral específica (grupo R). Los aminoácidos son las unidades básicas que forman las proteínas mediante enlaces peptídicos.

Átomo Es la unidad básica de la materia. Está compuesto por un núcleo, que contiene protones (con carga positiva) y neutrones (sin carga), y por electrones (con carga negativa) que giran alrededor del núcleo.

Baño térmico Entorno idealizado que intercambia energía térmica con el sistema simulado, manteniendo su temperatura constante.

Baño barométrico Entorno idealizado que permite el intercambio de volumen con el sistema, manteniendo la presión constante.

Barostato Mecanismo que regula la presión del sistema en una simulación de dinámica molecular, ajustando el volumen de la celda simulada para mantener la presión deseada.

Campo de fuerza (*force field*) Modelo matemático que describe las fuerzas internas y externas actuando sobre un sistema molecular. Incluye términos para enlaces, ángulos, torsiones, y fuerzas de Van der Waals y electrostáticas.

Catálisis Estudio de materiales que aceleran las reacciones químicas sin consumirse en el proceso.

Catalizador Sustancia que aumenta la velocidad de una reacción química sin consumirse en el proceso.

Centro de masas Punto que representa el movimiento del sistema como un todo. Por ejemplo, si no hay fuerza neta, entonces el centro de masas tiene un movimiento rectilíneo y uniforme.

Coarse-graining Técnica de simplificación utilizada en simulaciones en la que se agrupan varios átomos en una única partícula efectiva.

Colectivo En mecánica estadística, un colectivo (o ensamble) representa el conjunto de todas las configuraciones microscópicas posibles de un sistema que cumplen ciertas condiciones macroscópicas impuestas externamente, como número de partículas, volumen, energía o temperatura.

Colectivo canónico (NVT) Colectivo en el que se mantienen constantes el número de partículas (N), el volumen (V) y la temperatura (T) del sistema. Este tipo de simulación requiere el uso de un termostato.

- Colectivo isóbaro-isotermo (NPT)** Colectivo en el que se mantienen constantes el número de partículas (N), la presión (P) y la temperatura (T) del sistema. Este tipo de simulación requiere el uso tanto de un termostato como de un barostato.
- Colectivo microcanónico (NVE)** Colectivo en el que se mantienen constantes el número de partículas (N), el volumen (V) y la energía total (E) del sistema.
- Complejos biomoleculares** Conjuntos de macromoléculas biológicas, como proteínas, ácidos nucleicos o lípidos, que interactúan de manera específica y estable para llevar a cabo funciones biológicas concretas.
- Compuesto líder** Molécula con actividad biológica prometedora que actúa sobre una diana terapéutica concreta y que sirve como punto de partida para el desarrollo y optimización de nuevos fármacos.
- Configuración estable** Estado del sistema en el que las partículas se encuentran en un mínimo local de la superficie de energía potencial.
- Corrosión** Proceso químico o electroquímico mediante el cual un material, normalmente metálico, se degrada debido a su interacción con el entorno.
- CPU (Central Processing Unit)** Unidad central de procesamiento de un ordenador. Es el componente encargado de ejecutar instrucciones secuenciales y coordinar el funcionamiento del resto del sistema.
- Deformación anisotrópica** Cambio en la forma y/o volumen del sistema en el que las distintas direcciones espaciales pueden escalarse de manera diferente.
- Deformación isotrópica** Cambio en el volumen de un sistema en el que todas las dimensiones espaciales se escalan por igual, preservando la forma del sistema.
- Deposición** Técnica utilizada en ciencia de materiales y nanotecnología para depositar capas delgadas de material sobre una superficie.
- Diana biológica** Molécula del organismo (generalmente una proteína, como un receptor o una enzima) cuya modulación por parte de un compuesto químico produce un efecto terapéutico deseado.
- Docking** Técnica computacional utilizada para predecir la orientación y afinidad de unión entre una molécula pequeña (ligando) y una macromolécula (receptor o enzima).
- Enlace peptídico** Enlace covalente que une el grupo carboxilo ($-\text{COOH}$) de un aminoácido con el grupo amino ($-\text{NH}_2$) de otro, liberando una molécula de agua.
- Entorno solvatado** Sistema en el que una o más moléculas (solutos) están rodeadas por moléculas de disolvente, comúnmente agua.
- Enzima** Proteína especializada que actúa como catalizador biológico, acelerando reacciones químicas específicas sin consumirse en el proceso.
- Ergodicidad** Propiedad estadística según la cual el promedio temporal de una magnitud a lo largo de una trayectoria del sistema coincide con su promedio en el colectivo estadístico.

- Espacio de fases** Espacio matemático multidimensional en el que cada punto representa un estado completo del sistema, especificado por las coordenadas y los momentos de todas las partículas.
- Fase de equilibrio** Etapa de la simulación en la que se lleva al sistema a una situación de equilibrio a partir de su configuración inicial.
- Fase de producción** Etapa de la simulación en la que se generan las trayectorias del sistema, a partir de las cuales se calculan propiedades físicas y termodinámicas.
- Fluido** Sustancia que puede fluir y adaptarse a la forma del recipiente que la contiene. Incluye líquidos, gases y plasmas. Los fluidos carecen de forma fija y pueden sufrir deformaciones continuas bajo la acción de una fuerza, por pequeña que sea.
- GPU (*Graphics Processing Unit*)** Unidad de procesamiento gráfico especializada en operaciones de cálculo en paralelo.
- HPC (*High-Performance Computing*)** Conjunto de técnicas y recursos computacionales que permiten resolver problemas de elevada complejidad mediante el uso de arquitecturas paralelas, clústeres de ordenadores o supercomputadores.
- Ligando** Molécula que se une de manera específica y reversible a una macromolécula, como una proteína o un receptor.
- Lípido** Molécula orgánica que desempeña funciones estructurales y de almacenamiento energético.
- Litografía** Técnica de microfabricación utilizada para transferir patrones definidos sobre una superficie mediante el uso de radiación (habitualmente luz ultravioleta).
- Membranas celulares** Estructuras lipídicas que delimitan las células, regulando el transporte de sustancias y la comunicación entre compartimentos.
- Molécula** Agrupación de dos o más átomos unidos mediante enlaces químicos.
- Metaheurística** Estrategia de optimización general que guía y controla algoritmos de búsqueda para resolver problemas complejos donde los métodos deterministas resultan ineficaces. Las metaheurísticas no garantizan encontrar el óptimo global, pero son útiles para explorar y explotar adecuadamente el espacio de búsqueda.
- Paisaje energético** Representación conceptual de la energía potencial de un sistema en función de sus configuraciones microscópicas. Cada punto del paisaje corresponde a un estado del sistema, y la topología del mismo refleja la estabilidad relativa de las configuraciones, así como las trayectorias posibles de transición entre ellas.
- Paso de integración** Unidad de tiempo que separa dos estados consecutivos en una simulación numérica.
- Período** Tiempo que tarda una partícula en completar un ciclo completo de oscilación. En simulaciones de dinámica molecular, se usa como referencia para seleccionar un paso de integración adecuado, asegurando una representación precisa del movimiento atómico.

- PBC (Periodic Boundary Conditions)** Condiciones de contorno periódicas utilizadas en simulaciones para evitar efectos de borde artificiales. Consisten en replicar la celda de simulación en todas las direcciones del espacio, de modo que cuando una partícula sale por un lado, entra por el lado opuesto. Esto permite simular sistemas infinitos a partir de un número finito de partículas.
- Potencial de interacción** Función matemática que describe la energía potencial entre pares de partículas en función de su distancia o configuración relativa. Ejemplos comunes son el potencial de Lennard-Jones y el potencial de Coulomb.
- Proteína** Molécula biológica formada por cadenas de aminoácidos unidas mediante enlaces peptídicos. Su estructura tridimensional determina su función.
- Radio de corte** Distancia máxima a la que se consideran las interacciones entre partículas en una simulación. Más allá de este radio, las fuerzas (como las de Lennard-Jones o electrostáticas) se desprecian para reducir el coste computacional, asumiendo que su contribución es insignificante.
- Radio de vecindad** Distancia superior al radio de corte que se utiliza para construir listas de vecinos en simulaciones. Permite anticipar qué partículas podrían entrar en la zona de interacción en los siguientes pasos de la simulación, evitando recalcular interacciones en cada paso y mejorando la eficiencia computacional.
- Receptor** Proteína que reconoce y se une de manera específica a un ligando, desencadenando una respuesta bioquímica o celular.
- Reversibilidad temporal** Propiedad de las ecuaciones de movimiento de la mecánica clásica por la cual, si se invierte el sentido del tiempo y los momentos de las partículas, el sistema sigue una trayectoria compatible con las leyes físicas originales. Esta simetría implica que el sistema puede evolucionar hacia atrás en el tiempo siguiendo las mismas ecuaciones que rigen su evolución hacia adelante.
- Simplecticidad** Propiedad matemática de ciertos integradores que preservan la estructura geométrica del sistema hamiltoniano, evitando la acumulación sistemática de errores. En particular, conservan el volumen en el espacio de fases.
- Termostato** Mecanismo que regula la temperatura del sistema en una simulación de dinámica molecular, asegurando que permanezca próxima a un valor objetivo.
- Tiempo de relajación** Tiempo que tarda un sistema en volver a un estado de equilibrio o cuasi-equilibrio tras una perturbación.
- Trade-off** Compromiso inherente en el diseño de modelos o algoritmos, donde la mejora de una característica (como la precisión) suele implicar una pérdida en otra (como el coste computacional).

Bibliografía

- [1] Dirk P. Kroese, Thomas Taimre, and Zdravko I. Botev. *Handbook of Monte Carlo Methods*. Wiley Series in Probability and Statistics. John Wiley & Sons, 2011.
- [2] Norbert Attig, Kurt Binder, Helmut Grubmüller, and Kurt Kremer. *Computational Soft Matter: From Synthetic Polymers to Proteins*, volume 23 of *NIC Series*. John von Neumann Institute for Computing, Jülich, Germany, 2004.
- [3] B. J. Alder and T. E. Wainwright. Phase transition for a hard sphere system. *The Journal of Chemical Physics*, 27(5):1208–1209, 1957.
- [4] Aneesur Rahman. Correlations in the motion of atoms in liquid argon. *Physical Review*, 136(2A):A405–A411, 1964.
- [5] Loup Verlet. Computer experiments on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical Review*, 159(1):98–103, 1967.
- [6] Roberto Car and Michele Parrinello. Unified approach for molecular dynamics and density-functional theory. *Physical Review Letters*, 55(22):2471–2474, 1985.
- [7] Isaac Newton. *The Principia: Mathematical Principles of Natural Philosophy*. University of California Press, Berkeley, CA, 1999. Original work published in 1687.
- [8] Isaac Newton. *Philosophiæ Naturalis Principia Mathematica*. Royal Society, London, 1687.
- [9] Paul A. Tipler and Gene Mosca. *Physics for Scientists and Engineers*. W. H. Freeman and Company, New York, 6th edition, 2008. Chapter 8: Conservation of Energy.
- [10] M.P. Allen and D.J. Tildesley. *Computer Simulation of Liquids*. Oxford University Press, Oxford, 1989.
- [11] Shuichi Nosé. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of Chemical Physics*, 81(1):511–519, 1984.
- [12] William G. Hoover. Canonical dynamics: Equilibrium phase-space distributions. *Physical Review A*, 31(3):1695–1697, 1985.
- [13] Glenn J. Martyna, Douglas J. Tobias, and Michael L. Klein. Constant pressure molecular dynamics algorithms. *The Journal of Chemical Physics*, 101(5):4177–4189, 1994.
- [14] Michele Parrinello and Aneesur Rahman. Polymorphic transitions in single crystals: A new molecular dynamics method. *Journal of Applied Physics*, 52(12):7182–7190, 1981.
- [15] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation: From Algorithms to Applications*. Academic Press, 2nd edition, 2001.
- [16] Jacob N. Israelachvili. Intermolecular and surface forces. *Academic Press*, 3, 2011.
- [17] Tom Darden, Darrin York, and Lee Pedersen. Particle mesh ewald: An n-log(n) method for ewald sums in large systems. *The Journal of Chemical Physics*, 98(12):10089–10092, 1993.
- [18] William D. Cornell, Piotr Cieplak, Christopher I. Bayly, Kenneth M. Merz, David M. Ferguson, David C. Spellmeyer, Thomas Fox, James W. Caldwell, and Peter A. Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995.
- [19] Alexander D. MacKerell Jr., D. Bashford, M. Bellott, R. L. Dunbrack Jr., J. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, B. Roux, M. Schlenkrich, J. C.

- Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz-Kuczera, D. Yin, and M. Karplus. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The Journal of Physical Chemistry B*, 102(18):3586–3616, 1998.
- [20] Matthew W. Mahoney and William L. Jorgensen. A five-site model for liquid water and the reproduction of the density anomaly by rigid, nonpolarizable potential functions. *The Journal of Chemical Physics*, 112(20):8910–8922, 2000.
- [21] James C Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Eric Villa, Christophe Chipot, Robert D Skeel, Laxmikant Kale, and Klaus Schulten. Scalable molecular dynamics with namd. *Journal of Computational Chemistry*, 26(16):1781–1802, 2005.
- [22] David A. Case, Thomas E. Cheatham III, Thomas A. Darden, Robert E. Duke, Hideo Gohlke, Andreas W. Goetz, Neil Homeyer, Scott Izadi, Carlos Simmerling, Junmei Wang, Ray Luo, and Peter A. Kollman. *AMBER 2021 Reference Manual*. University of California, San Francisco, 2021.
- [23] William Humphrey, Andrew Dalke, and Klaus Schulten. Vmd: Visual molecular dynamics. *Journal of Molecular Graphics*, 14(1):33–38, 1996.
- [24] Saman Alavi. *Molecular Simulations: Fundamentals and Practice*. Wiley-VCH, 2020.
- [25] Herbert Goldstein, Charles Poole, and John Safko. *Classical Mechanics*. Addison Wesley, 3rd edition, 2002.
- [26] Michael Griebel, Stephan Knapek, and Gerhard Zumbusch. *Numerical Simulation in Molecular Dynamics: Numerics, Algorithms, Parallelization, Applications*. Springer, 2007.
- [27] Uri M. Ascher and Linda R. Petzold. *Numerical Methods for Differential Equations: Initial Value Problems and Boundary Value Problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1998.
- [28] Benedict Leimkuhler and Sebastian Reich. *Simulating Hamiltonian Dynamics*. Cambridge University Press, 2004.
- [29] Andrew R. Leach. *Molecular Modelling: Principles and Applications*. Pearson Education, 2nd edition, 2001.
- [30] William C. Swope, Hans C. Andersen, Peter H. Berens, and Kent R. Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics*, 76(1):637–649, 1982.
- [31] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak. Molecular dynamics with coupling to an external bath. *The Journal of Chemical Physics*, 81(8):3684–3690, 1984.
- [32] Ignacio García Quirce. Cálculo de propiedades macroscópicas mediante dinámica molecular, 2016.
- [33] Mark E. Tuckerman. *Statistical Mechanics: Theory and Molecular Simulation*. Oxford University Press, 2010.
- [34] Hans C. Andersen. Molecular dynamics simulations at constant pressure and/or temperature. *The Journal of Chemical Physics*, 72(4):2384–2393, 1980.
- [35] CEAD Group. Comportamiento isotrópico y anisotrópico de los materiales. <https://knowledge.ceadgroup.com/es/comportamiento-isotr%C3%B3pico-y-anisotr%C3%B3pico-de-los-materiales>.
- [36] D. J. Wales and J. P. K. Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.
- [37] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2nd edition, 2006. Capítulo 2: Unconstrained Optimization Problems.
- [38] IBM Corporation. ¿qué es el descenso de gradiente?, oct 2021.
- [39] A. E. Eiben and J. E. Smith. *Introduction to evolutionary computing*. Springer, 2003.

- [40] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [41] Alan Hinchliffe. *Molecular Modelling for Beginners*. John Wiley & Sons, 2 edition, 2008.
- [42] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- [43] M. E. J. Newman and G. T. Barkema. *Monte Carlo methods in statistical physics*. Oxford University Press, 1999.
- [44] Leonardo Dagum and Ramesh Menon. *OpenMP: An Industry-Standard API for Shared-Memory Programming*, volume 5. IEEE, 1998.
- [45] William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. MIT Press, 2nd edition, 1999.
- [46] John E Stone, David J Hardy, Ivan S Ufimtsev, and Klaus Schulten. Accelerating molecular modeling applications with graphics processors. *Journal of Computational Chemistry*, 31(16):2757–2767, 2010.
- [47] Mark J Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C Smith, Berk Hess, and Erik Lindahl. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1:19–25, 2015.
- [48] Juan A. Bueren-Calabuig. Dinámica molecular. In J.A. Bueren-Calabuig et al., editor, *Bases Físicas de la Bioinformática Estructural*, chapter 17. Universidad de Sevilla, 2023.
- [49] D.A. et al. Case. The amber biomolecular simulation programs. *Journal of Computational Chemistry*, 26(16):1668–1688, 2005.
- [50] B.R. et al. Brooks. Charmm: The biomolecular simulation program. *Journal of Computational Chemistry*, 30(10):1545–1614, 2009.
- [51] W. F. van Gunsteren and H. J. C. Berendsen. *GROMOS Manual*. BIOMOS, Groningen, The Netherlands, 1987.
- [52] W.L. Jorgensen and J. Tirado-Rives. The oplis potential functions for proteins. *Journal of the American Chemical Society*, 110:1657–1666, 1988.
- [53] S. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *Journal of Computational Physics*, 117:1–19, 1995.
- [54] P. et al. Eastman. Openmm 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology*, 13(7):e1005659, 2017.
- [55] Michael Levitt. *Computer Modelling of Biomolecular Processes*. World Scientific, 1996.
- [56] Martin Karplus and J. Andrew McCammon. Molecular dynamics simulations of biomolecules. *Nature Structural Biology*, 9(9):646–652, 2002.
- [57] Julia M. Goodfellow, editor. *Computer Modelling in Molecular Biology*. Wiley-VCH, 1995.
- [58] D. Peter Tieleman and Herman J. C. Berendsen. *Molecular dynamics simulations of a fully hydrated dipalmitoylphosphatidylcholine bilayer with different macroscopic boundary conditions and parameters*, volume 105. 2001.
- [59] J. Andrew McCammon and Bruce R. Harvey. *Dynamics of Proteins and Nucleic Acids*. Cambridge University Press, 1987.
- [60] Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine learning for molecular simulation. *Annual Review of Physical Chemistry*, 71:361–390, 2020.
- [61] Justin S Smith, Olexandr Isayev, and Adrian E Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical science*, 8(4):3192–3203, 2017.

Bibliografía

- [62] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics. *Physical Review Letters*, 120(14):143001, 2018.
- [63] O. T. Unke, S. Chmiela, M. Gastegger, K. T. Schütt, H. E. Sauceda, and K. R. Müller. Machine learning force fields. *Chemical Reviews*, 121(16):10142–10186, 2021.
- [64] Ellad B. Tadmor and Ronald E. Miller. *Modeling Materials: Continuum, Atomistic and Multiscale Techniques*. Cambridge University Press, 2011.
- [65] William Smith and Timothy R. Forester. *Introduction to Computational Chemistry*. Wiley, 2nd edition, 2007.
- [66] S. Plimpton, P. Crozier, and A. Thompson. Lammmps—large-scale atomic/molecular massively parallel simulator. *Sandia National Laboratories Technical Report*, 2007. <https://www.lammps.org>.
- [67] Cristina Torres Andón, Alberto Curto, and Francesca Balas. Nanomaterials and their interaction with biological systems: insights from molecular dynamics simulations. *Frontiers in Bioengineering and Biotechnology*, 9:629276, 2021.
- [68] Jean-Pierre Hansen and Ian R. McDonald. *Theory of Simple Liquids: With Applications to Soft Matter*. Academic Press, 2013.
- [69] D. C. Rapaport. *The Art of Molecular Dynamics Simulation*. Cambridge University Press, 2 edition, 2004.
- [70] Arieh Warshel and Michael Levitt. Theoretical studies of enzymic reactions: dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *Journal of Molecular Biology*, 103(2):227–249, 1976.
- [71] Hans Martin Senn and Walter Thiel. Qm/mm methods for biomolecular systems. *Angewandte Chemie International Edition*, 48(7):1198–1229, 2009.
- [72] Hu Lin and Donald G Truhlar. Qm/mm: what have we learned, where are we, and where do we go from here? *Theoretical Chemistry Accounts*, 117(2):185–199, 2007.
- [73] Martin J Field. Combining quantum mechanics and molecular mechanics. In *Reviews in Computational Chemistry*, pages 1–46. Wiley Online Library, 2000.
- [74] Cp2k: A quantum chemistry and solid-state physics software package. <https://www.cp2k.org>. Accessed June 2025.
- [75] Anton van der Ven and Kaushik Bhattacharya. Multiscale modeling in materials science: The road to molecular dynamics simulation of complex systems. *MRS Bulletin*, 36(11):876–884, 2011.
- [76] Mark E Tuckerman, Bruce J Berne, and Glenn J Martyna. Reversible multiple time scale molecular dynamics. *The Journal of chemical physics*, 97(3):1990–2001, 1992.
- [77] Robert G. Parr and Weitao Yang. *Density-Functional Theory of Atoms and Molecules*. Oxford University Press, 1989.
- [78] Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2 edition, 2006.
- [79] Justin Solomon. *Numerical Algorithms*. MIT, 2021.
- [80] Cleve Moler. *Numerical Computing with MATLAB*. SIAM, 2004. Capítulo 7 recomendado para introducción al análisis de estabilidad.
- [81] Benedict Leimkuhler and Charles Matthews. *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*, volume 39 of *Interdisciplinary Applied Mathematics*. Springer, 2015.
- [82] Tamar Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide*, volume 21 of *Interdisciplinary Applied Mathematics*. Springer, 2010.

- [83] Szilárd Páll, Mark J. Abraham, Carsten Kutzner, Berk Hess, and Erik Lindahl. Tackling exascale software challenges in molecular dynamics simulations with gromacs. *arXiv preprint arXiv:1506.00716*, 2015.
- [84] Mark J Abraham, Teemu Murtola, Roland Schulz, Szilárd Páll, Jeremy C Smith, Berk Hess, and Erik Lindahl. Heterogeneous parallelization and acceleration of molecular dynamics simulations in gromacs. *The Journal of Chemical Physics*, 153(13):134110, 2020.
- [85] Daniel F. Styer. *Statistical Mechanics*. Oberlin College, Oberlin, OH, 2023.
- [86] CompChems. Barostats in molecular dynamics, 2023.
- [87] Nadia Villalba Andrade. *Simulación computacional de propiedades de transporte usando dinámica molecular: difusión y viscosidad de líquidos simples*. PhD thesis, Universidad Nacional Autónoma de México, 2022.
- [88] Christophe Chipot and Andrew Pohorille. *Free Energy Calculations: Theory and Applications in Chemistry and Biology*. Springer, 2007.
- [89] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [90] David P. Landau and Kurt Binder. *A Guide to Monte Carlo Simulations in Statistical Physics*. Cambridge University Press, 2nd edition, 2005.
- [91] Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proceedings of the National Academy of Sciences*, 99(20):12562–12566, 2002.
- [92] Glenn M Torrie and John P Valleau. Nonphysical sampling distributions in monte carlo free-energy estimation: Umbrella sampling. *Journal of Computational Physics*, 23(2):187–199, 1977.
- [93] Sunhwan Jo, Taehoon Kim, and Wonpil Im. Charmm-gui: a web-based graphical user interface for charmm. *Journal of computational chemistry*, 29(11):1859–1865, 2008.
- [94] Massimiliano Bonomi, Davide Branduardi, Giovanni Bussi, Carlo Camilloni, Davide Provasi, Paolo Raiteri, Davide Donadio, Fabrizio Marinelli, Fabio Pietrucci, Ricardo A Broglia, et al. Plumed: A portable plugin for free-energy calculations with molecular dynamics. *Computer Physics Communications*, 180(10):1961–1972, 2009.
- [95] Oliver F Lange and David Baker. Integrative modeling of biomolecular complexes. *Annual review of biophysics*, 41:605–624, 2012.
- [96] Sandro Bottaro and Kresten Lindorff-Larsen. Integrative structural biology using molecular simulations. *Nature Structural & Molecular Biology*, 27(6):590–598, 2020.
- [97] Scott A Hollingsworth and Ron O Dror. Molecular dynamics simulation for all. *Neuron*, 99(6):1129–1143, 2018.
- [98] Adam Hospital, José R Goñi, Modesto Orozco, and Josep Lluís Gelpí. Molecular dynamics simulations: advances and applications. *Advances and applications in bioinformatics and chemistry: AABC*, 8:37, 2015.
- [99] William L. Jorgensen, Jayaraman Chandrasekhar, Jeffrey D. Madura, Roger W. Impey, and Michael L. Klein. Comparison of simple potential functions for simulating liquid water. *The Journal of Chemical Physics*, 79(2):926–935, 1983.