



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

FACULTAD DE CIENCIAS

DEPARTAMENTO DE FÍSICA

**Desarrollo del código de simulación SIMAT
(SIMulaciones ATomísticas) en GPUS para estudiar
sistemas biológicos a gran escala.**

Presenta

M. EN C. LUIS ROLANDO GUARNEROS NOLASCO

Para obtener el grado de

DOCTOR EN CIENCIAS (FÍSICA)

Bajo la Dirección del

**Dr. Roberto López Rendón
Dr. Felipe de Jesús Rodríguez Romero
Dr. Jorge Mulia Rodríguez**

Toluca de Lerdo, Edo. de México

Diciembre de 2018



INDICE

Resumen.....	3
1. Introducción	4
2. Antecedentes	5
2.1 Algoritmos de integración a través del operador de Liouville.....	5
2.2. Generación de ensambles de equilibrio vía el operador de Liouville	5
2.3 Algoritmos simplécticos	8
2.4 Ensamble microcanónico (NVE)	8
2.5 Ensamble canónico (NVT).....	9
2.6 Ensamble Isotérmico-Isobárico (NPT)	11
3. Hipótesis.....	13
4. Objetivos	14
4.1 Objetivo General.....	14
4.2 Objetivo Específico.....	14
5.1 Metodología	15
5.1 Dinámica Molecular.....	15
5.2 Condiciones Iniciales	16
5.4 Distribución inicial de velocidades	17
5.5 Condiciones de frontera periódica	18
5.4 Potencial de interacción	19
5.4.1 Potencial intramolecular	19
5.4.2 Potencial intermolecular	21
5.5 CUDA: Una plataforma para cómputo heterogéneo.....	22
6. Publicaciones del autor	24
6.1 Artículo Aceptado	24
6.2 Artículo en revisión.....	24
6.4 Confirmación de artículo enviado.....	31
7.1 Discusión general	45
8.1 Conclusiones	50
Referencias.....	51
Anexos	55
Anexo 1. Pseudocódigo y diagrama de flujo del software HIMD	55
Anexo 2. Infraestructura utilizada.....	55



Resumen

La Biofísica Computacional se ha convertido en una de las ramas más dinámicas de la biología contemporánea, ocupándose principalmente del estudio de los procesos celulares a nivel molecular. La unidad fundamental de investigación son las macromoléculas, las proteínas sobre todo. Sin embargo, el estudio del movimiento de una proteína muestra una notable dificultad. Una de las alternativas más prometedoras para estudiar los sistemas biológicos son sin duda los métodos de simulación molecular tales como el de Dinámica Molecular (DM). La DM se ha convertido en una de las herramientas numéricas más poderosas para estudiar propiedades termodinámicas de macromoléculas como son las proteínas y otras biomoléculas.

Por lo anterior, en este trabajo nos enfocamos en realizar la implementación de los algoritmos en los ensambles NVE y NVT de dinámica molecular basada en los operadores de Liouville [22-23]. En el ensamble NVT se usan las denominadas cadenas de termostatos Nose-Hoover [21]. Estos algoritmos se implementaron haciendo uso de la tecnología basada en Unidades de Procesamiento Gráficos (GPU) con lo cual nos permitió reducir el tiempo de ejecución de estos algoritmos para el desarrollo de las simulaciones.

Se publicó el artículo titulado “In silico Analysis of the Structural Properties of PSMA and its Energetic Relationship with Zn as Cofactor” en la revista *Journal of Nuclear Physics, Material Sciences, Radiation and Applications Journal* con DOI: 10.15415/jnp.2018.61020 con una simulación de dinámica molecular del PSMA con el propósito de caracterizarlo de forma energética y estructural en donde se dilucidaron las diferencias de PSMA con sus dos iones Zn^{2+} como cofactores y sin ellos en el perfil de energía libre, y en cuatro parámetros estructurales: desviaciones cuadráticas medias de la raíz y fluctuaciones cuadradas medias de la raíz por átomo y residuo de aminoácido, radio de giro y SASA.

Del mismo modo, un segundo artículo pre aceptado titulado “High-throughput of measure-preserving integrators for constant temperature molecular dynamics simulations on GPUs” con doi:10.20944/preprints201811.0250.v1 en la revista *High-Throughput*, en el cual se presenta el desempeño de los algoritmos NVE y NVT con las cadenas de Nose-Hoover implementadas en GPUs para desarrollar simulaciones de dinámica molecular.



CAPÍTULO 1

1. Introducción

Una de las alternativas más prometedoras para estudiar los sistemas biológicos son sin duda los métodos de simulación molecular tales como el de Dinámica Molecular (DM). La DM se ha convertido en una de las herramientas numéricas más poderosas para estudiar propiedades termodinámicas de macromoléculas como son las proteínas y otras biomoléculas. Sin embargo, la DM presenta dos grandes limitantes cuando se pretende estudiar sistemas de gran tamaño como son las proteínas a nivel molecular, estas son: el tamaño del sistema (el número de átomos a estudiar) y el tiempo de integración usado para evaluar las ecuaciones de movimiento. El uso convencional del tiempo de integración es del orden de femtosegundos, lo cual garantiza la conservación de energía principalmente dando así confiabilidad a los resultados obtenidos. Sin embargo, los procesos dinámicos de mayor interés que suceden en las proteínas son del orden de microsegundos o más allá. Por lo tanto, se requiere contar con programas de simulación cuya tecnología de programación logre superar estas limitantes en cuanto al tiempo de integración, y poder así realizar simulaciones a grandes escalas de tiempo. La otra limitante es referente al número de átomos a estudiar en una simulación con DM [1-5].

En general, los sistemas moleculares son complejos y consisten de un gran número de átomos, por lo cual sería imposible determinar sus propiedades de forma analítica. Para usar la DM en el estudio de sistemas biológicos se necesita contar con una infraestructura computacional robusta que supere estas dificultades. Actualmente, la tecnología empleada para la fabricación de tarjetas gráficas o GPUs (Graphics Processing Unit) han evolucionado de una manera muy significativa, de tal forma que dejaron de ser procesadores exclusivos para el procesamiento de gráficos y se convirtieron en sofisticados co-procesadores de bajo precio y alto rendimiento, que permitieron aumentar la capacidad de una computadora personal de escritorio o portátil con las mejoras en su arquitectura y un modelo flexible de programación para el manejo masivo de datos en paralelo, lo que los convierte en una atractiva alternativa de cómputo de alto rendimiento (High Performance Computing en inglés HPC). Por lo anterior, se pretende desarrollar el código de dinámica molecular HIMD (Simulaciones Atomísticas) implementado en las tecnologías de cómputo de alto rendimiento como son los GPUs.



CAPÍTULO 2

2. Antecedentes

2.1 Algoritmos de integración a través del operador de Liouville

En los últimos años se ha hecho común los métodos de dinámica molecular a sistemas de gran tamaño permitiendo desarrollar metodologías aplicables a diferentes condiciones termodinámicas donde se ha uso del concepto de ensamble colectivo. Un ensamble es un ente constituido por un conjunto de sistemas que se encuentran en el mismo estado termodinámico, pero en diferentes estados microscópicos. De esta manera, un ensamble queda definido por una serie de variables naturales que se suponen fijas y cuyas propiedades se obtienen a partir del potencial termodinámico asociado a dichas variables.

Los ensambles se clasifican en función de las magnitudes termodinámicas que permanecen constantes:

- **NVE. Ensamble microcanónico.** Son aquellos en los que la única condición es mantener la energía constante dado un volumen y un número de moléculas fijo. Corresponde a un sistema aislado.
- **NVT. Ensamble canónico.** Las variables termodinámicas que se mantienen constantes son el número de moléculas, el volumen y la temperatura. Corresponde a un sistema cerrado.
- **NPT. Ensamble isotérmico-isobárico.** La presión como la temperatura y el número de moléculas permanecen fijos. Es de suma importancia en la validación de campos de fuerza.

Para poder simular cualquiera de estos ensambles, es necesario contar con un buen integrador numérico, esto es, un algoritmo que nos permita determinar las posiciones y momentos de un sistema en cualquier instante de tiempo dadas las condiciones iniciales.

2.2. Generación de ensambles de equilibrio vía el operador de Liouville

La metodología está basada en los esquemas propuestos por Tuckerman para la generación de ensambles de equilibrio vía el operador de Liouville [22-23]. Para más detalles vea Roberto López Rendón y colaboradores[43].



La discusión se basa en los sistemas Hamiltonianos puesto que son de gran importancia en simulaciones de DM ya que poseen características especiales, incluyendo la estructura simpléctica, conservan el volumen del espacio fase y además cumplen con la propiedad de reversibilidad en el tiempo. Cualquier integrador numérico debe cumplir con estas propiedades. Se mostrará un formalismo general para derivar integradores reversibles y simplécticos partiendo de una formulación basada en el operador de Liouville introduciendo algunos conceptos básicos. En mecánica clásica, todo sistema conservativo viene caracterizado por su Hamiltoniano, función que define la energía del sistema para cada estado termodinámico en términos de sus posiciones y momentos. Por ello examinamos un sistema simple de una partícula moviéndose en una dimensión con un Hamiltoniano dado por

$$H = \frac{p^2}{2m} + U(x) \quad (2.1)$$

Las ecuaciones de movimiento para este sistema están dadas por

$$\dot{x} = \frac{p}{m} \quad \dot{p} = -\frac{dU}{dx} = F(x) \quad (2.2)$$

Ahora, supongamos que tenemos una función $A(x, p)$ y cuya evolución en el tiempo es de acuerdo a

$$\frac{dA}{dt} = \{A, H\} \quad (2.3)$$

Donde H , es el Hamiltoniano clásico dado por 2.1 y los corchetes representan el paréntesis de Poisson definido según

$$\{A, H\} = \frac{\partial A}{\partial x} \frac{\partial H}{\partial p} - \frac{\partial A}{\partial p} \frac{\partial H}{\partial x} \quad (2.4)$$

La evolución de A nos dará como resultado las ecuaciones de Hamilton. Para verificar este punto podemos elegir $A(x, p) = x$ y sustituirla en la ecuación 2.3 para evaluar 2.4, teniendo como resultado

$$\begin{aligned} \frac{dx}{dt} &= \{A, H\} \\ \{x, H\} &= \frac{\partial x}{\partial x} \frac{\partial H}{\partial p} - \frac{\partial x}{\partial p} \frac{\partial H}{\partial x} \\ &= \frac{p}{m} \end{aligned} \quad (2.5)$$



Lo cual es equivalente a la primera relación que aparece en 2.2 puesto que $\partial x/\partial p = 0$, y por lo tanto la posición evoluciona según $\dot{x} = p/m$ como es de esperarse. De igual manera, haciendo $A(x, p) = p$ y sustituyendo este valor en la ecuación 2.3 para evaluar 2.4, tenemos que

$$\begin{aligned} \frac{dp}{dt} &= \{p, H\} \\ \{p, H\} &= \frac{\partial p}{\partial x} \frac{p}{m} - \frac{\partial p}{\partial p} \frac{\partial U}{\partial x} \\ &= - \frac{dU}{dx} = F(x) \end{aligned} \tag{2.6}$$

Donde hemos obtenido la segunda relación de 2.2. De esta manera vemos que la evolución de los momentos nos da la fuerza. Además, usando los paréntesis de Poisson podemos obtener las ecuaciones de movimiento para cualquier sistema dado un Hamiltoniano. Ahora, si definimos un vector bidimensional dado por $\Gamma = (x, p)$ y escribimos las ecuaciones de Hamilton en términos de Γ según la ecuación 2.3 de tal forma que

$$\frac{d\Gamma}{dt} = \{\Gamma, H\} \tag{2.7}$$

Y definimos un operador actuando sobre Γ que sea equivalente a los paréntesis de Poisson

$$iL\Gamma = \{\Gamma, H\} \tag{2.8}$$

donde el operador iL es conocido como el operador de Liouville. Siempre y cuando el Hamiltoniano no dependa explícitamente del tiempo, las ecuaciones de movimiento de acuerdo con este operador pueden escribirse como

$$\begin{aligned} \dot{\Gamma} &= iL\Gamma \\ \frac{d\Gamma}{dt} &= iL\Gamma \end{aligned} \tag{2.9}$$

integrando ésta última relación desde $\Gamma(0)$ hasta $\Gamma(t)$ el lado izquierdo y desde $t_0 = 0$ hasta t el lado derecho, encontramos que la solución para 2.9 queda formalmente determinada por

$$\Gamma(t) = e^{iLt} \Gamma(0) \tag{2.10}$$

Donde el operador $\exp(iLt)$ es conocido como el propagador clásico, $\Gamma(0)$ son las condiciones iniciales, es decir, posiciones y momentos a $t = 0$, $r_i(0)$, $p_i(0)$; t representa el tiempo total de



integración y $\Gamma(t)$ el estado final del sistema al tiempo $t = \Delta t$ con las nuevas posiciones y momentos $r_i(t), p_i(t)$. Aunque la ecuación 2.10 es solamente una solución formal a la ecuación 2.9, es el punto de partida para la derivación de esquemas de integración numéricos.

2.3 Algoritmos simplécticos

En la construcción de esquemas de integración, es importante tener en cuenta dos propiedades características de los sistemas Hamiltonianos.

La primera es que deben cumplir con el Teorema de Liouville, el cual establece la conservación del elemento de volumen del espacio fase. Por otro lado, la segunda propiedad se refiere a la reversibilidad en el tiempo de las ecuaciones de Hamilton. Esta propiedad implica que si una condición inicial x_0 evoluciona a un tiempo t , el sistema regresará a su estado inicial en otro intervalo de tiempo de longitud t . Cualquier esquema de integración aplicado a las ecuaciones de Hamilton deben respetar estas dos propiedades. Los integradores que cumplen con estas dos propiedades son llamados integradores o algoritmos simplécticos.

2.4 Ensemble microcanónico (NVE)

Un algoritmo que nos permita determinar las posiciones y momentos de un sistema en cualquier instante de tiempo usando las condiciones iniciales se le llama un integrador numérico. Existen buenos algoritmos de movimiento, es decir que cumplan con la conservación del volumen del espacio fase y que cumplan con las propiedades de reversibilidad.

A continuación, se muestra el algoritmo para energía constante implementado en el código HIMD aplicando la tecnología de GPU. Para mayor detalle de las ecuaciones y resultados véase de Roberto López Rendón y colaboradores [43].



```

for i = 1 to N       $\mathbf{p}_i\left(\frac{\Delta t}{2}\right) \leftarrow \mathbf{p}_i(0) + \frac{\Delta t}{2}\mathbf{F}_i(0)$ 
for i = 1 to N       $\mathbf{r}_i(\Delta t) \leftarrow \mathbf{r}_i(0) + \Delta t\left(\frac{\mathbf{p}_i(\Delta t/2)}{m_i}\right)$ 
Calcular las fuerzas  $\mathbf{F}_i(\Delta t)$ 
for i = 1 to N       $\mathbf{p}_i(\Delta t) \leftarrow \mathbf{p}_i\left(\frac{\Delta t}{2}\right) + \frac{\Delta t}{2}\mathbf{F}_i(\Delta t)$ 

```

Fig. 1 Algoritmo Verlet de velocidades al utilizar la metodología de los propagadores.

Donde \mathbf{F}_i es la fuerza, \mathbf{P}_i es el momento de la partícula i , respectivamente, r_i es la posición, m_i es la masa y Δt es el tiempo de integración de la dinámica. Este pseudocódigo representa el algoritmo llamado Verlet de velocidades que se obtiene de forma natural al usar la metodología de los propagadores. La descripción es como sigue. Para avanzar los átomos Δt se necesita conocer la posición y el momento a un tiempo inicial t_0 . Primero se calcula el momento a la mitad del tiempo $\Delta t/2$, con ese momento se calculan las nuevas posiciones al tiempo Δt , con esas posiciones se calcula la fuerza al tiempo Δt y finalmente con la fuerza al tiempo Δt y el momento al tiempo $\Delta t/2$ se calcula el nuevo momento al tiempo Δt . Esto sería para un ciclo de dinámica molecular y para cada partícula o átomo.

2.5 Ensamble canónico (NVT)

Una manera eficiente de mantener la temperatura constante en una dinámica molecular es con la metodología del llamado espacio fase extendido propuesta por Andersen [17]. En esta metodología las posiciones y momentos de las partículas están complementadas por variables adicionales en el espacio fase que controlan las fluctuaciones de la temperatura. Los métodos ampliamente utilizados para realizar la simulación de un ensamble NVT que se basan en el esquema de sistemas extendidos son el Termostato de Nosé-Hoover (NH) [18-19] y el de Cadenas de Termostatos de Nosé-Hoover (NHC).

La metodología de NHC es una generalidad del método de NH y fue propuesta por Martyna, Klein y Tuckerman (MTK) [21]. La idea es acoplar el termostato del sistema físico real a una cadena de termostatos, e incorporar el Hamiltoniano extendido estos grados de libertad. Solo el primer termostato ya definido por NH interactúa con el sistema real y los demás están acoplados entre ellos. El espacio fase que generan las NHC está formado por un conjunto adicional de variables extendidas



que evolucionan tanto sus posiciones como los momentos. De esta manera un punto en el espacio fase está complementado por dichas variables extendidas de la siguiente manera

$$\Gamma(t) = (\mathbf{p}_i, \mathbf{r}_i, \{p_\eta\}, \{\eta\}) \quad (2.11)$$

Donde p_η es el momento y η es la posición por molécula de las variables extendidas. Las ecuaciones de movimiento propuestas por MKT para simular un sistema en el ensamble canónico son:

$$\begin{aligned} \dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i} \\ \dot{\mathbf{p}}_i &= \mathbf{F}_i - \frac{p_{\eta_1}}{Q_1} \mathbf{p}_i \\ \dot{\eta}_k &= \frac{p_{\eta_k}}{Q_k} \quad k = 1, \dots, M \\ p_{\eta_k} &= G_k - \frac{p_{\eta_{k+1}}}{Q_{k+1}} p_{\eta_k} \quad k = 1, \dots, M-1 \\ \dot{p}_{\eta_M} &= G_M \end{aligned} \quad (2.12)$$

Donde Q_k es la masa de los termostatos, las G 's son las fuerzas del baño

$$\begin{aligned} G_1 &= \sum_{i=1}^N \frac{\mathbf{p}_i^2}{m_i} - 3Nk_bT \\ G_k &= \frac{p_{\eta_{k-1}}^2}{Q_{k-1}} - k_bT \end{aligned} \quad (2.13)$$

La temperatura está controlada mediante estas fuerzas que a su vez influyen en los momentos de los termostatos en la cadena. Si hacemos $M = 1$ es equivalente a tener el termostato original de NH y eliminando los termostatos nos lleva al ensamble NVE. La física incorporada en las ecuaciones 2.12 está basada en el hecho de que el término $-(p_{\eta_1}/Q_1)\mathbf{p}_i$ actúa como un tipo de fuerza de fricción dinámica. Esta fuerza de fricción regula la energía cinética de modo que su promedio es el valor canónico correcto. De una manera similar, la variable del baño ($k+1$) sirve para modular las fluctuaciones en la k -ésima variable de modo que cada variable del baño es conducida a tener un promedio canónico propio.

$$H' = H(\mathbf{p}, \mathbf{r}) + \sum_{k=1}^M \frac{p_{\eta_k}^2}{2Q_k} + k_bT \left[dN\eta_1 + \sum_{k=2}^M \eta_k \right]$$

Donde $H(\mathbf{p}, \mathbf{r})$ es el Hamiltoniano del sistema físico.

De acuerdo al trabajo de Roberto López Rendón [43], aplicando matemáticas y álgebra se puede llegar al siguiente esquema:

$$\begin{array}{ll}
 \text{Thermo - Update} \left(\frac{\Delta t}{2}, \{\eta\}, \{p_\eta\}, \mathbf{P}_i \right) & \dot{\mathbf{r}}_i = \frac{\mathbf{P}_i}{m_i} \\
 \text{for } i = 1 \text{ to } N & \mathbf{p}_i \left(\frac{\Delta t}{2} \right) \leftarrow \mathbf{p}_i(0) + \frac{\Delta t}{2} \mathbf{F}_i(0) & \dot{\mathbf{p}}_i = \mathbf{F}_i - \frac{p_{\eta_1}}{Q_1} \mathbf{p}_i \\
 \text{for } i = 1 \text{ to } N & \mathbf{r}_i(\Delta t) \leftarrow \mathbf{r}_i(0) + \frac{\Delta t}{m_i} \mathbf{p}_i \left(\frac{\Delta t}{2} \right) & \dot{\eta}_k = \frac{p_{\eta_k}}{Q_k} \quad k = 1, \dots, M \\
 \text{Calcular las fuerzas } \mathbf{F}_i(\Delta t) & & \dot{p}_{\eta_k} = G_k - \frac{p_{\eta_{k+1}}}{Q_{k+1}} p_{\eta_k} \\
 \text{for } i = 1 \text{ to } N & \mathbf{p}_i(\Delta t) \leftarrow \mathbf{p}_i \left(\frac{\Delta t}{2} \right) + \frac{\Delta t}{2} \mathbf{F}_i(\Delta t) & \dot{p}_{\eta_M} = G_M \\
 \text{Thermo - Update} \left(\frac{\Delta t}{2}, \{\eta\}, \{p_\eta\}, \mathbf{P}_i \right) & &
 \end{array}$$

Fig. 2 Algoritmo NVT incorporando los termostatos de NHC.

El ensamble NVE se encuentra en la parte central.

En estas ecuaciones se puede ver la implementación del algoritmo NVT en un código de Dinámica Molecular. La parte central corresponde al ensamble NVE. Thermo - update es la rutina donde se incorporan los termostatos de NHC y se aplica al inicio y al final de la dinámica. Las siguientes ecuaciones que se programan en la rutina Thermo - update escritas en términos de velocidades, corresponde a escalar las velocidades de las variables extendidas del primer termostato que esta acoplado a las partículas o átomos del sistema.

2.6 Ensamble Isotérmico-Isobárico (NPT)

En estos algoritmos, el volumen es introducido como una variable dinámica adicional junto con una variable correspondiente al momento con el fin de mantener las condiciones isobáricas del sistema. Para entender la metodología que genera un algoritmo de movimiento que conserva el espacio fase, ver López Rendón [43], en el cual se implementa un algoritmo en un código de Dinámica Molecular para realizar simulaciones NPT.

$$\begin{array}{l}
 \dot{\mathbf{r}}_i = \frac{\mathbf{P}_i}{m_i} + \frac{p_\epsilon}{W} \mathbf{r}_i \\
 \dot{\mathbf{p}}_i = \mathbf{F}_i - \left(1 + \frac{1}{N} \right) \frac{p_\epsilon}{W} \mathbf{p}_i - \frac{p_{\eta_1}}{Q_1} \mathbf{p}_i \\
 \text{Volumen} \\
 \dot{V} = \frac{dV}{W} p_\epsilon \\
 \dot{p}_\epsilon = dV(P_{\text{int}} - P_{\text{ext}}) + \frac{1}{N} \sum_{i=1}^N \frac{\mathbf{P}_i^2}{m_i} - \frac{p_{\xi_1}}{Q'_1} p_\epsilon
 \end{array}$$



Termostato

$$\dot{\eta}_k = \frac{p_{\eta_k}}{Q_k} \quad k = 1, \dots, M$$

$$\dot{p}_{\eta_k} = G_k - \frac{p_{\eta_{k+1}}}{Q_{k+1}} p_{\eta_k}$$

$$\dot{p}_{\eta_M} = G_M$$

Barostato

$$\dot{\xi}_k = \frac{p_{\xi_k}}{Q'_k} \quad k = 1, \dots, M$$

$$\dot{p}_{\xi_k} = G'_k - \frac{p_{\xi_{k+1}}}{Q'_{k+1}} p_{\xi_k}$$

$$\dot{p}_{\xi_M} = G'_M$$

Fig. 3 Algoritmo NPT.



CAPÍTULO 3

3. Hipótesis

Derivado de la estructura de los algoritmos utilizados para el termostato de las cadenas de Nose-Hoover, se intuye que su implementación en arquitecturas unificadas para cómputo en paralelo es de manera natural, por lo que su eficiencia será maximizada al ejecutarse en unidades de procesamiento gráfico (GPU).



CAPÍTULO 4

4. Objetivos

Los objetivos del presente proyecto son:

4.1 *Objetivo General*

El objetivo general de este proyecto de tesis conlleva dos componentes principales. Una va dirigida a la implementación de novedosos algoritmos para realizar simulaciones de sistemas biológicos a gran escala en diversos ensambles estadísticos como son a temperatura constante. La segunda directriz va encaminada en el desarrollo e innovación de tecnología de software de simulación molecular con el desarrollo del código HIMD (Simulaciones Atomísticas).

4.2 *Objetivo Específico*

La manera de garantizar llevar a buen término las metas propuestas de este proyecto es mediante la ejecución de los siguientes objetivos específicos:

- 1.- Revisar y documentar la derivación de algoritmos de dinámica molecular a partir del operador de Liouville [22-23].
- 2.- Revisar y documentar la derivación de ensambles estadísticos a temperatura y presión constantes (Canónico e Isotérmico-Isobárico) bajo el contexto de sistemas extendido como las Cadenas de Nose-Hoover [21].
- 3.- Implementar los algoritmos desarrollados por Roberto López-Rendón y colaboradores en el código de simulación molecular HIMD en tecnologías de GPUS [43].
- 4.- Validar los resultados obtenidos del código escrito en GPU contra los resultados reportados por la literatura.
- 5.- Realizar pruebas de desempeño del código escrito en GPU en las diferentes tarjetas GTX de Nvidia instaladas en la supercomputadora Olinka de la Facultad de Ciencias.



CAPÍTULO 5

5.1 Metodología

La metodología utilizada en la realización de este proyecto se centra en los métodos de simulación molecular. Específicamente, el método de dinámica molecular el cual se basa en resolver numéricamente las ecuaciones clásicas de movimiento. El estudio teórico - computacional de cualquier sistema a nivel molecular involucra hacer uso de un modelo de potencial de interacción, tal modelo debe estar conformado por un conjunto de ecuaciones analíticas o empíricas que describan las interacciones moleculares. La calidad de los resultados obtenidos con esta metodología depende básicamente del potencial de interacción o campo de fuerzas usado para describir cada uno de los grados de libertad del sistema. Para lograr este desafío, es necesario contar con una infraestructura metodológica capaz de atacar estos problemas. Se han desarrollado algoritmos más eficientes para estudiar la dinámica de sistemas complejos bajo distintas condiciones termodinámicas, estos algoritmos diseñados bajo en contexto de operadores de Liouville son capaces de conservar el espacio fase.

5.1 Dinámica Molecular

La dinámica molecular [26] es una técnica de simulación computacional que estudia el comportamiento de un sistema de muchas partículas calculando la evolución en el tiempo y promediando una cantidad de interés sobre un tiempo suficientemente largo. Para esto es necesario integrar numéricamente las ecuaciones de movimiento de un sistema de N partículas a través de la segunda ley de Newton:

$$\mathbf{F}_i = m_i \frac{d^2 \mathbf{r}_i}{dt^2} \quad i = 1, \dots, N \quad (5.1)$$

Donde \mathbf{F}_i son las fuerzas que actúan sobre cada partícula debidas a un potencial por pares $U(r_{ij})$, esto es:

$$\mathbf{F}_i = \sum_{i \neq j}^N \mathbf{F}_{ij} \quad (5.2)$$

\mathbf{F}_{ij} es la fuerza entre las partículas i y j que está expresada en términos del potencial:

$$\sum_{i \neq j}^N \mathbf{F}_{ij} = - \sum_{i \neq j}^N \nabla U(r_{ij}) \quad (5.3)$$

Por lo tanto, tenemos:

$$\sum_{i \neq j}^N \mathbf{F}_{ij} = - \sum_{i \neq j}^N \frac{dU(r_{ij})}{dr_{ij}} \frac{\mathbf{r}_{ij}}{r_{ij}} \quad (5.4)$$

Donde $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ es la distancia relativa entre los centros de las 2 partículas.

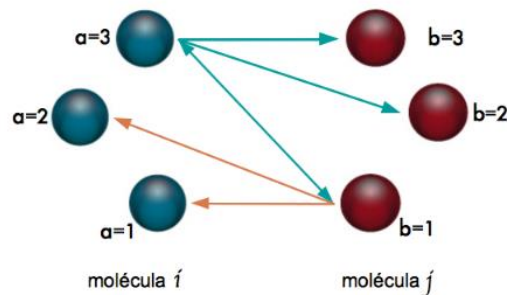


Fig. 4 Interacciones en el modelo átomo-átomo entre moléculas distintas donde el átomo $a = 1$ de la molécula i interactúa con los átomos de la molécula j y así sucesivamente con los demás átomos.

El corazón de una simulación con dinámica molecular depende de una descripción adecuada del sistema en términos del potencial de interacción.

5.2 Condiciones Iniciales

El primer paso en la simulación de cualquier sistema mediante DM es especificar las posiciones iniciales de las partículas que lo constituyen. Consideremos un sistema compuesto por N partículas. La forma más sencilla de establecer la estructura inicial de un fluido, cuya densidad numérica de partículas sea ρ , consiste en asignar posiciones aleatorias a las N partículas dentro de un volumen N/ρ . Sin embargo, este procedimiento plantea serios inconvenientes prácticos, ya que algunas partículas pueden quedar muy próximas entre sí, ocasionando que la energía de interacción entre ellas sea extremadamente alta, una situación muy improbable en la realidad y que, además, dificulta la integración de las ecuaciones del movimiento del sistema. Por ello resulta conveniente situar a las



partículas inicialmente en las posiciones de una red cristalina, lo que evita traslapes fortuitos entre ellas. Esta estructura cristalina se fundirá al iniciar la propagación del sistema, equilibrándose para llegar a las temperaturas y densidades típicas de los fluidos simulados. En principio puede elegirse cualquier tipo de red cristalina. En la práctica, se comprueba que los resultados de la simulación son independientes de cuál sea la red cristalina inicial. Por ello, se utiliza la más sencilla de todas, la red cúbica centrada en las caras (FCC, face centered cubic) [25]. El número de átomos contenidos en una red será $N = 4n^3$, siendo n un número entero positivo. Ello implica que la red solo puede construirse con unos determinados números de átomos: 32, 108, 256, 500, 864 y así sucesivamente.

Una vez asignadas las posiciones de todos los átomos del líquido, deben especificarse también sus velocidades iniciales. Lo usual es elegir estas velocidades de forma aleatoria dentro de cierto intervalo, distribuidas de manera uniforme o mediante una distribución gaussiana.

5.4 Distribución inicial de velocidades

La distribución inicial de velocidades parte de una distribución de Maxwell-Boltzmann [54].

$$P(v_{i,\alpha}) = \left[\frac{m}{2\pi k_B T} \right]^{\frac{1}{2}} e^{-\frac{mv_{i,\alpha}^2}{2k_B T}} \quad (5.5)$$

donde $v_{i,\alpha}$ es el componente α ($= x, y, z$) de la velocidad del átomo i . La distribución puede ser utilizada para definir la temperatura instantánea $T(t)$ utilizando el teorema de equipartición

$$\left\langle \frac{mv_\alpha^2}{2} \right\rangle = \frac{1}{2} k_B T \quad (5.6)$$

que relaciona la energía cinética media con la temperatura ($\langle \dots \rangle$ - promedio de conjunto). Ec. (5.6) se puede obtener directamente de la Ec. (5.5). Porque la media del conjunto corresponde a la media de todas las velocidades de los átomos, la temperatura instantánea $T(t)$ es

$$k_B T(t) = \frac{1}{N_f} \sum_{i,\alpha} mv_{i,\alpha}^2 \quad (5.7)$$



donde N_f es el número de grados de libertad. Por lo tanto, la Ec. (5.7) nos permite calcular la temperatura instantánea a partir de la distribución de velocidades. También está claro que, para una dada la realización de la distribución de velocidad, $T(t)$ no es estrictamente igual a T . Aunque las velocidades se generan utilizando la distribución en la ec. (5.5) a la temperatura T , la el sistema molecular contiene solo un número finito de átomos y el (instantáneo) la temperatura $T(t)$ se desviará de T . Para mantener la temperatura constante, se puede volver a escalar las velocidades de acuerdo a

$$v_{i,\alpha} = \sqrt{\frac{T}{T(t)}} v_{i,\alpha} \quad (5.8)$$

Es sencillo mostrar que la temperatura instantánea después de volver a escalar $T'(t) \equiv T$. Si no se realiza el reescalamiento, las fluctuaciones relativas de temperatura en el sistema de N átomos están dados por (para el sistema de aproximadamente 1000 átomos las fluctuaciones en $T(t)$ son $\sim 3\%$).

$$\frac{\Delta T(t)}{\langle T(t) \rangle} = \frac{(\langle T^2(t) \rangle - \langle T(t) \rangle^2)^{\frac{1}{2}}}{\langle T(t) \rangle} \sim N^{-\frac{1}{2}} \quad (5.9)$$

5.5 Condiciones de frontera periódica

Una característica especial cuando se realiza una simulación de DM es principalmente el número de partículas que integran el sistema a modelar. Principalmente cuando se realizan simulaciones que implican a cientos o miles de átomos. El tiempo que realizan de uso de cómputo en los programas de DM crece exponencialmente con el número de átomos que se desean modelar, esto se debe por la evaluación que realizan las fuerzas entre los átomos, por lo que es necesario mantener un número tan reducido como sea posible. Sin embargo, el problema es que en un sistema de tamaño tan reducido (comparado con el número de partículas de una mol, del orden de 10^{23}) no es representativo del seno de un líquido, ya que el sistema está dominado por los efectos de superficie.

Debido a lo anterior, esto se logra resolver cuando se aplican las denominadas condiciones de frontera periódica [54]. Con esta técnica el cubo en el cual se encuentra el sistema, la celda primaria, es rodeado por réplicas exactas en todas las direcciones, las cuales son llamadas celdas imágenes, con la cual se forma una red infinita. Con esto las celdas imágenes almacenan los mismos átomos que la celda primaria y, durante una simulación, cada uno de los átomos de las celdas imágenes se mueve

de la misma forma que los átomos de la celda primaria. Así, si un átomo de la celda primaria la abandona por una de sus caras, su imagen de la cara opuesta entra en la celda primaria.

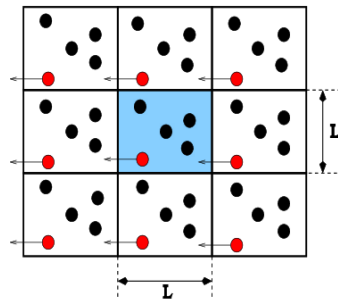


Fig. 5 Condiciones de frontera periódicas en un sistema periódico bidimensional.

La celda sombreada corresponde a la celda central.

5.4 Potencial de interacción

El éxito de una simulación con DM dependerá del empleo de un modelo de potencial adecuado que contendrá la física esencial del sistema. De la fidelidad con que éste represente las interacciones reales entre las partículas dependerá de la calidad de los resultados. El campo de fuerza de CHARMM [55] es un campo de fuerza que va encaminado al modelado de grandes biomoléculas tales como proteínas o polímeros. Se divide en 2 contribuciones:

5.4.1 Potencial intramolecular

Se refiere a las contribuciones dentro de una molécula que se toman en cuenta para realizar una dinámica [55], como son:

Distancia de enlace (armónica o rígida)

Se pueden modelar mediante una función de tipo armónico.

$$U(r_{ij}) = \frac{k_r}{2} (r_{ij} - r_0)^2 \quad (5.10)$$

Donde:

r_{ij} es la distancia relativa entre los átomos i y j

r_0 es su distancia de equilibrio

k_r constante de enlace

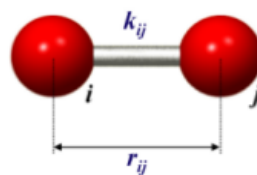


Fig. 6 Distancia de enlace.

Ángulo de enlace (se forma con 3 ángulos adyacentes)

Están relacionados con la estructura de la molécula que a su vez están relacionados con la disposición de los pares de electrones.

$$U(\theta_{ijk}) = \frac{k_{\theta}}{2} (\theta_{ijk} - \theta_0)^2 \quad (5.11)$$

Donde:

θ_{ijk} es el ángulo de enlace entre los átomos i , j y k

θ_0 es el ángulo de equilibrio

k_{θ} es la constante de enlace

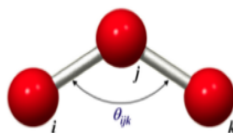


Fig. 7 Ángulo de enlace.

Ángulo de torsión (se forma entre el plano de 4 átomos consecutivos)

Se incluyen con el propósito de conservar la geometría molecular en conformaciones predeterminadas.

$$U(\phi_{ijkl}) = \sum \frac{V_n}{2} [1 + \cos(n\phi - \phi_0)] \quad (5.12)$$

Donde:

n indica en número de mínimos que presenta la superficie de energía potencial de 0° a 360°

V_n es la constante asociada a la barrera de rotación

ϕ es el ángulo de torsión relativo que conforman los dos planos formados por las partículas i, j, k en un caso y j, k, l en el otro caso

ϕ_0 es el ángulo de torsión donde la energía potencial pasa a través de su valor mínimo.

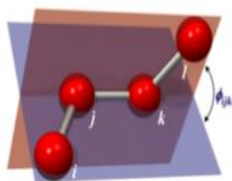


Fig. 8 Ángulo de torsión.

5.4.2 Potencial intermolecular

Potencial de Lennard-Jones.

Describe las interacciones entre átomos de líquidos y las interacciones tipo Van der Waals entre átomos de moléculas diferentes [55]. Para el sistema analizado en el presente estudio la interacción entre el átomo a en la molécula i y el átomo b en la molécula j es representado por la siguiente expresión.

$$U_{LJ} = \sum_{i=1}^{N-1} \sum_{j>i}^N \sum_{a=1}^{N_i} \sum_{b=1}^{N_j} u_{LJ}(r_{iajb}) = \sum_{i=1}^{N-1} \sum_{j>i}^N \sum_{a=1}^{N_i} \sum_{b=1}^{N_j} 4\epsilon_{ab} \left[\left(\frac{\sigma_{ab}}{r_{iajb}} \right)^{12} - \left(\frac{\sigma_{ab}}{r_{iajb}} \right)^6 \right] \quad (5.13)$$

El término $\left(\frac{\sigma_{ab}}{r_{iajb}} \right)^{12}$ describe la repulsión y el término $\left(\frac{\sigma_{ab}}{r_{iajb}} \right)^6$ describe la atracción entre los átomos, σ_{ab} es la medida del diámetro de los átomos que interactúan y ϵ_{ab} es el parámetro que representa la medida de atracción entre los sitios en moléculas distintas, r_{iajb} es la distancia entre el sitio a en la molécula i y el sitio b en la molécula j, Figura 9.

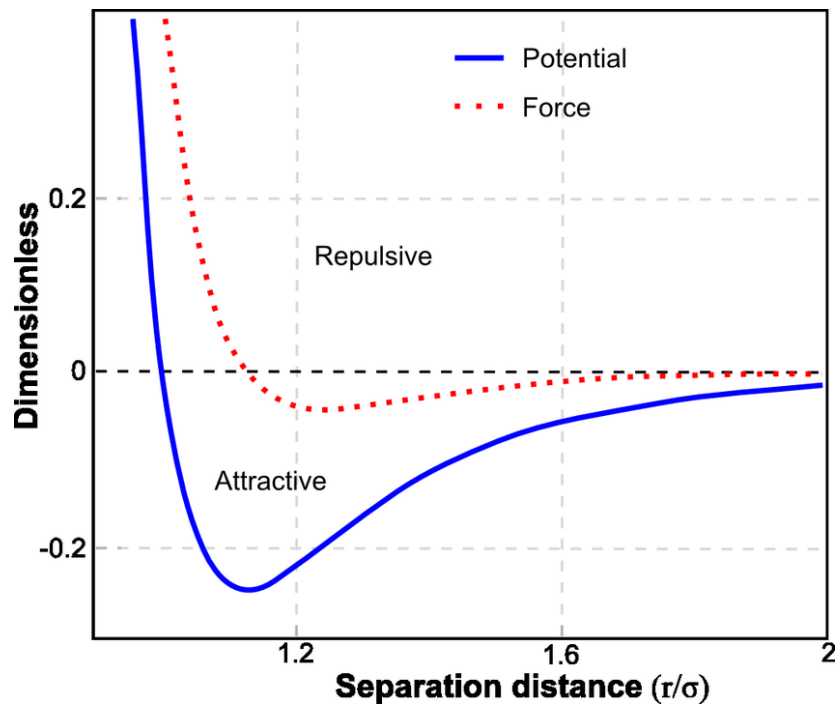


Figura 9. Potencial de Lennard-Jones. La línea azul representa lo atracción de los átomos y la línea roja su repulsión.

Potencial de Coulomb.

Teniendo en cuenta las cargas puntuales atómicas, este potencial permite modelar las interacciones electrostáticas producidas debido a la distribución de cargas dentro de la molécula.

$$U_c = \frac{1}{4\pi\epsilon_0} \frac{q_{ia}q_{jb}}{r_{iajb}} \quad (5.14)$$

Donde ϵ_0 es la permitividad en el vacío y r_{iajb} la distancia relativa entre la carga q_{ia} del átomo a en la molécula i , y q_{jb} la carga del átomo b en la molécula j .

5.5 CUDA: Una plataforma para cómputo heterogéneo

CUDA [51] es una de las interfaces de programación para cálculo paralelo desarrollado por la empresa NVIDIA la cual aprovecha la gran potencia de las GPU (unidad de procesamiento gráfico) en donde se obtiene un incremento considerable en la ejecución de operaciones de un sistema. Hoy en día, miles de personas dedicadas al desarrollo de software, científicos e investigadores han encontrado un gran aliado al desarrollar aplicaciones prácticas para esta tecnología. El diseño de la arquitectura de trabajo de CUDA permite la operación y comunicación de un GPU con un CPU a través de un bus PCI-Express.

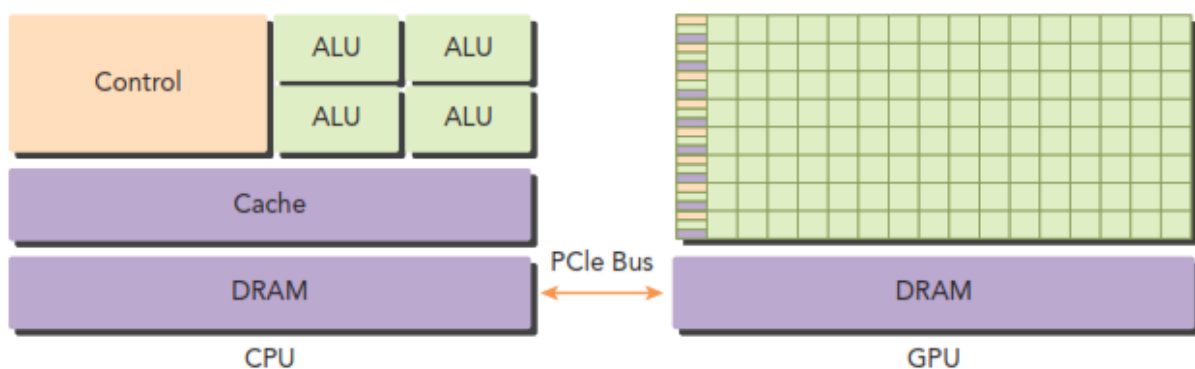


Fig. 10 Comparación arquitectura de CPU y GPU

Un típico procesamiento de un programa basado en CUDA sigue el siguiente flujo:

- Se copia datos de la memoria del CPU a la memoria del GPU

- Se invoca el kernel para ejecutar los datos almacenados en la memoria de la GPU
- Se copian los datos de la memoria del GPU al CPU

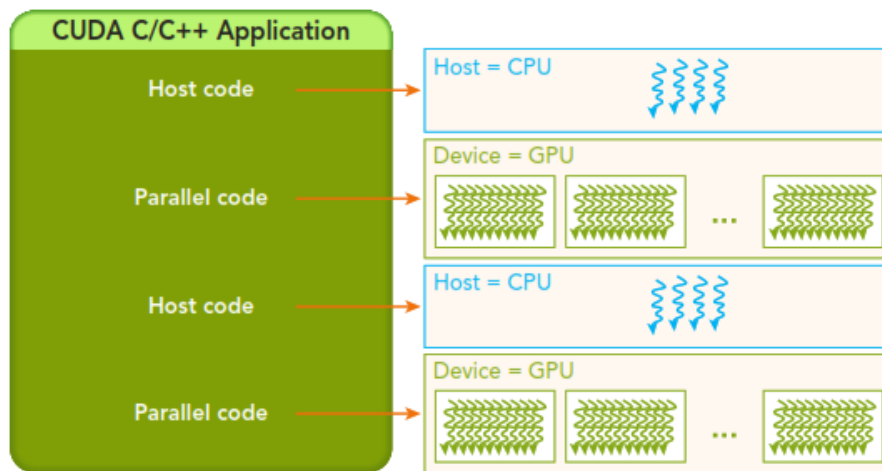


Fig. 11 Flujo de ejecución de un programa basado en CUDA

La plataforma de CUDA es accesible a través de bibliotecas basadas en C y C++ que cuenta con directivas de compilación y aplicaciones para interfaces de programación. Debido a su alto potencial de cálculo se han realizado extensiones a lenguajes estándar en la industria entre los que se pueden mencionar Fortran y Python, entre otros, con lo cual se puede utilizar en diversos campos de aplicación.

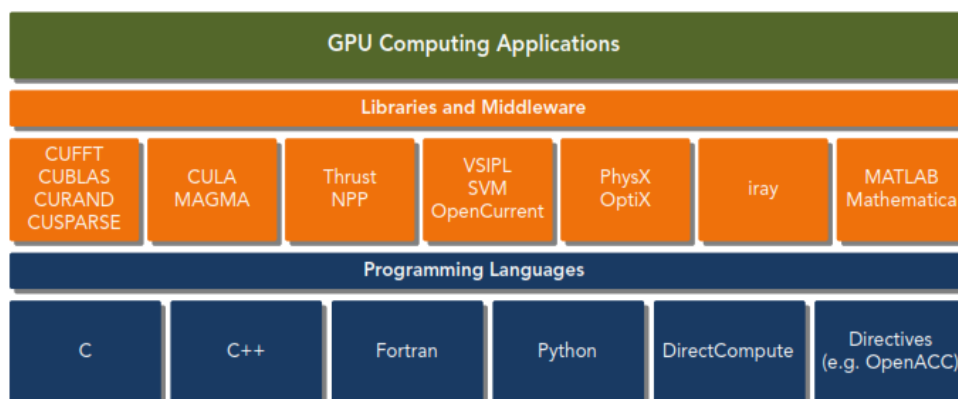


Figura 12. Librerías y lenguajes de programación en CUDA



CAPÍTULO 6

6. Publicaciones del autor

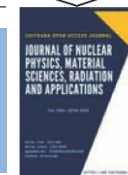
6.1 Artículo Aceptado

M.A. Fuentes, L. A. Mandujano, R. López, **L.R. Guarneros**, E. Azorín, D. Osorio-González. In silico Analysis of the Structural Properties of PSMA and its Energetic Relationship with Zn as Cofactor. *J. Nucl. Phys. Mat. Sci. Rad. A. Vol. 6, No.1, Aug. 2018*, pp.115–120. DOI: 10.15415/jnp.2018.61020

6.2 Artículo en revisión

Guarneros-Nolasco, L.; Terrón-Mejía, K.A.; Mulia-Rodríguez, J.; Osorio-Gonzalez, D.; López-Rendón, R.; Rodriguez-Romero, F. High-throughput of measure-preserving integrators for constant temperature molecular dynamics simulations on GPUs. *Preprints 2018*, 2018110250 (doi: 10.20944/preprints201811.0250.v1).

Se agrega contenido de los dos artículos.



***In silico* Analysis of the Structural Properties of PSMA and its Energetic Relationship with Zn as Cofactor**

M.A. Fuentes¹, L. A. Mandujano², R. L6pez³, L.R. Guarneros⁴, E. Azorin⁵, D. Osorio-Gonzalez^{1*}¹*Molecular Biophysics Laboratory of the Faculty of Sciences, Autonomous University of Mexico State, Mexico*²*Molecular Biophysical Modeling and Design Laboratory, Mexiquense University, S. C.*³*Multiscale Molecular Bioengineering Laboratory of the Faculty of Sciences, Autonomous University of Mexico State, Mexico**Email: dog@uaemex.mx

ARTICLE INFORMATION

Received: June 16, 2018

Revised: July 05, 2018

Accepted: July 20, 2018

Published online: August 6, 2018

Keywords:

PSMA structural analysis, PSMA with and without Zn as cofactor.

DOI: [10.15415/jnp.2018.61020](https://doi.org/10.15415/jnp.2018.61020)

ABSTRACT

The prostate-specific membrane antigen (PSMA) is a 100 kDa type II transmembrane glycoprotein with enzymatic activity similar to the family of zinc-dependent exopeptidases. This protein is of great medical and pharmacological interest as overexpression in prostate cells is related to the progression of prostate cancer; therefore, it represents an important target for the design of radiopharmaceuticals. The presence of two Zn²⁺ ions in the active site is crucial to the enzymatic activity and the design of high-affinity inhibitors. The amino acid residues coordinating these ions are highly conserved in PSMA orthologs from plants to mammals, and site-mutagenesis assays of these residues show a loss of enzymatic function or reduction of the kinetic parameters. In the present work, we performed molecular dynamics simulation of PSMA with the purpose of characterizing it energetically and structurally. We elucidated the differences of PSMA with its two Zn²⁺ ions as cofactors and without them in the free energy profile, and in four structural parameters: root mean square deviations and root mean square fluctuations by atom and amino acid residue, radius of gyration, and solvent accessible surface area.

1. Introduction

Prostate cancer is a global problem since it is the second type of cancer with the highest incidence and the fifth with the highest number of deaths among men, while in Mexico it is the cancer type with higher incidence and mortality rate [1]. The Prostate-specific Membrane Antigen (PSMA) is a type II transmembrane glycoprotein of 100 kDa composed of 750 amino acids with at least three functions: hydrolytic NAALADase activity, folate hydrolase and dipeptidyl peptidase IV activity [2-4]. This protein is overexpressed in poorly differentiated and metastatic cells; consequently, it is considered an important indicator of prostate cancer and a target for the development of many inhibitors [5-6]. Recently, small molecule inhibitors (SMI) targeting PSMA have been developed; these are zinc-binding compounds linked to glutamate or a glutamate isomer. Urea-based SMI (Glu-urea-R) have demonstrated to specifically bind to PSMA and inhibit its activity in the LNCaP cell line. In such compounds, Glu-urea is the binding terminal and the R-group is the coupling terminal to other chemical groups such as a linker and a chelator associated with radionuclides [7-10].

The theranostic agents are based on the use of a radionuclide with the same PSMA-targeting ligands for therapy and diagnosis; for this purpose ¹⁷⁷Lu, ²²⁵Ac, and ¹³¹I have been used. Particularly, ¹⁷⁷Lu associated with PSMA-617 has provided a safe and effective therapy in patients with metastatic castration-resistant prostate cancer [11-14]. PSMA-617 is a ligand conformed by a DOTA chelator (1,4,7,10-Tetraazacyclododecane-1,4,7,10-tetraacetic acid) conjugated with Glu-urea-Lys pharmacophore by a linker composed of two aromatic rings; it was designed for labeling with ¹⁷⁷Lu and ⁶⁸Ga to achieve high-quality image and efficient endotherapy [15-16].

Crystallographic structural studies of PSMA made possible to elucidate the interaction of the protein with the inhibitors. Structural information of PSMA is available only for the extracellular part of the protein (residues 44-750). It reveals that the protein exists as a symmetrical homodimer *in vivo*, each polypeptide monomer having three structural domains: a protease-like domain (residues 56-116 and 352-591), an apical domain -also called the protease-associated domain- (residues 117-351), and the helical domain -also called the C-terminal domain- (residues 592-750). [Figure. 1]. The active site of the protein contains a binuclear

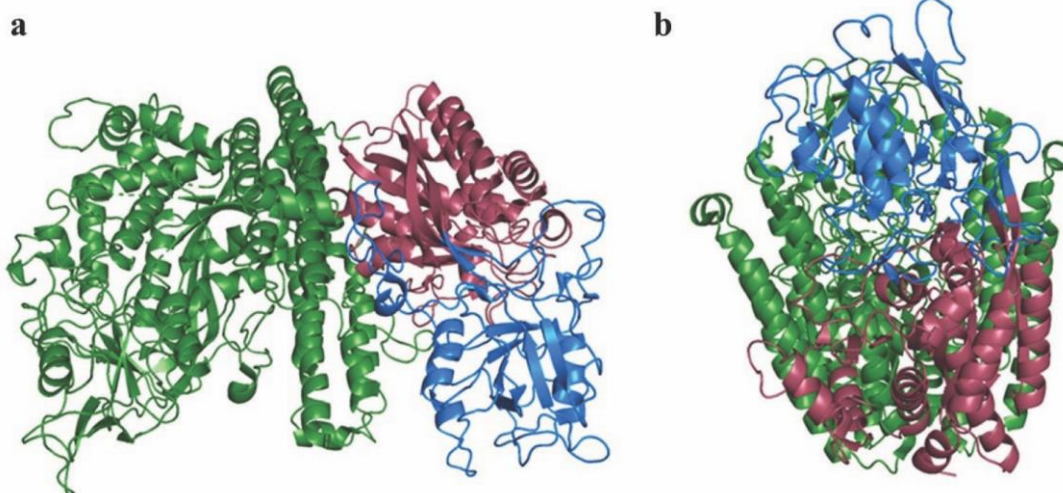


Figure 1. PSMA structural domains. a) lateral view and b) superior view. The protease domain (56-116 and 352-591) is colored in red, the apical domain (117-351) in blue, and the helical domain in green (592-750).

Zn active site, catalytic residues, and a substrate-binding arginine-rich patch. A water ligand bridges the two zinc atoms, each coordinated by endogenous ligands: Zn(1) by His553 and Glu425, Zn(2) by His377 and Asp453, both atoms bounded by Asp387. Glu424 and Tyr552 have the catalytic function. A substrate/inhibitor-binding cavity with an area of about 1100 Å², and a diameter and deep of about 20 Å, is formed in the interface between the three domains; this interface is considered large as it buries around 4600 Å². It is localized in the helical domain and is formed by two pockets, S1'(pharmacophore) and S1(non-pharmacophore). The cavity has an arginine patch (Arg463, Arg534, and Arg536) involved in the right orientation of the substrate for catalysis; it is aligned with the S1 which has a chlorine ion that keeps Arg534 in a conformation that allows the interaction with the substrate, while Arg536 and Arg463 are flexible conferring tolerance to different chemical groups [Figure 2]. The "glutamate sensor" is responsible of detecting the absence or presence of glutamate in S1' pocket and is formed by residues 692-704 together with Lys699 and Tyr700, which are also important for the specific binding of glutamate along with Arg210 in the apical domain [17-19].

The pharmacophore Glu-urea-Lys (iPSMA) [Figure 3] is capable of binding with PSMA because it has three carboxylic acid groups. The glutamate-urea fraction of the inhibitor has a predisposition to be oriented towards S1', and lysine is used for conjugation or derivatization, through the free amine, with a linker region or a chelating agent residing in S1. When both pockets are occupied, the hydrophobic contact (due to an aromatic agent) increases resulting in higher affinity [20,21]. The affinity to PSMA is due to the binding of glutamate by its -carboxylate, which

forms a bridge with the guanidinium group of Arg210, and hydrogen bonds with the hydroxide groups of Tyr552 and Tyr700, while the -carboxylate interacts with Lys699 and Ans257. The catalytic activity is carried out by Glu424, that extracts a proton from the water molecule situated between the zinc atoms and activates it [19,22]. The urea group serves as a zinc-binding group (ZBG) because the oxygen of the molecule interacts with Tyr552, His553, the active water molecule and Zn(1), while N groups form hydrogen bonds: N(1) with the main carbonyl chain of Gly-518 and the carboxylate of Glu-424, and N(2) with -carbonyl of Gly-518 [19, 23-24].

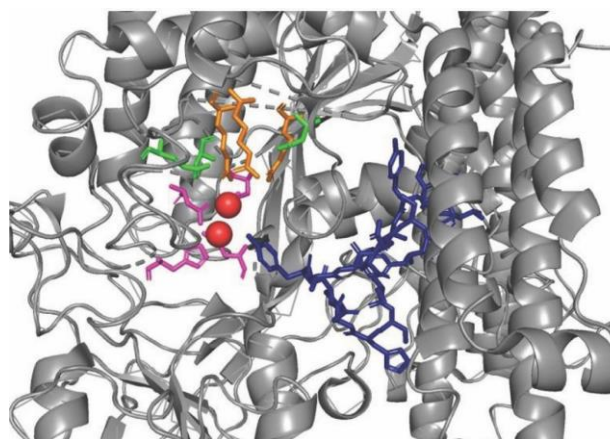


Figure 2. PSMA active site. Zinc ions are observed as red spheres, while the coordinating ligands of these ions are presented as magenta sticks. The substrate binding cavity is colored in blue, the arginine patch in orange and stabilizing ligands in green.

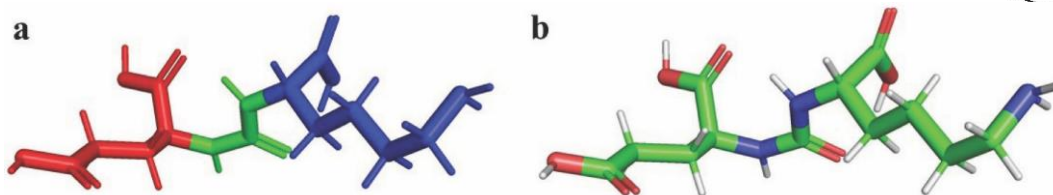


Figure 3. iPSMA structure (Glu-urea-Lys. a) The pharmacophore is composed by a glutamate (red), urea (green) and lysine (blue). b) Oxygen atoms are colored in red sticks, carbon colored in green sticks, hydrogen colored in white sticks and nitrogen colored in blue sticks.

2. Method

2.1 Theoretical model

To evaluate the structure of a biomolecule, we calculate the Root Mean Square Deviation (RMSD) between its atoms or residues, which allows comparing its molecular structure reached at any time t_1 , with respect to another structure that occurred at a reference t_2 , for this purpose we use the expression

$$RMSD_{t_1, t_2} = \frac{1}{M} \sqrt{\frac{\sum_{i=1}^N m_i \|r_i(t_1) - r_i(t_2)\|^2}{2}} \quad (1)$$

Where m_i is the mass of the atom or residue i in the position r_i , and therefore $M = \sum_{i=1}^N m_i$ is the total mass of the system.

It is possible to analyze the flexibility of the protein through the oscillations of its amino acids, this property can be estimated through the Root Mean Square Fluctuation (RMSF) which can be calculated using

$$RMSF(i) = \frac{1}{T} \sqrt{\sum_{t_1=1}^T \|r_i(t_1) - \bar{r}_i\|^2} \quad (2)$$

Where T is the time over which the average is calculated.

To have a rough measure for the compactness of a protein structure, we calculate the radius of gyration with

$$R_g = \sqrt{\frac{\sum_{i=1}^N m_i \|r_i - \bar{r}\|^2}{\sum_{i=1}^N m_i}} \quad (3)$$

Where m_i is the mass of atom i and r_i the position of atom i with respect to the center of mass of the molecule.

The Solvent Accessible Surface Area (SASA) of a molecule is the region of its surface that has contact with the solvent and is, therefore, an indicator of the structural changes generated during the folding of the protein because SASA is directly proportional to its free energy.

2.2 Methodology

For *in silico* analysis we used classical Molecular Dynamics (MD), the atomic coordinates for PSMA were extracted from the 1Z8L crystal structure in the Protein Data Bank. The protein was solvated using atomistic TIP3P water in a cubic box with at least 10 Å distance around the complex. We used the CHARMM force field. The two initial stages in the preparation of the systems, the stages of minimization and equilibration, were carried out using the NAMD2.6 program. First, the systems were minimized for 10000 steps, then thermalized for 10 ps at 300 K reinitializing the velocities every 20 ps. The equilibration of the systems was carried out with NAMD2.6 under periodic boundary conditions, time step 2 fs, cutoff 9 Å, Langevin damping 0.1/ps. The long-range electrostatic interactions are accounted for using the particle mesh Ewald method, with a maximum grid spacing of 1.0 Å. Bond lengths are maintained rigid with the SHAKE. A final run of 100 ns was executed to assure that all properties, such as potential energy, van der Waals and electrostatic interactions, are in thermodynamic equilibrium. After 15 ns these quantities remain stable.

After the initial equilibration phase, the production simulation (100 ns) was carried out using platform ACEMD. This simulation was conducted in the NVT ensemble which resulted from the equilibration phase. We use a longer timestep of 4 fs thanks to the use of the hydrogen mass repartition scheme implemented in ACEMD. Coordinate snapshots were generated every 5 ps collecting a total of 5000 conformational states for subsequent post-production analysis. The structural analysis presented in this work was held on 100 ns of simulation time. Molecular images displayed in this work were produced using PyMOL. All simulations referred were performed at our Cluster OLINKA, a platform designed to run molecular simulations to multiscale with GPUs.

3. Results and Discussion

It is noteworthy that the active site is centered around the two zinc ions, and separates the S1' and S1 pockets. One of the Zn²⁺ is considered the catalytic ion, coordinated by His553

and Glu425, and the other is the co-catalytic, coordinated by His377 and Asp453. Asp387 and a water molecule bridge the two zinc atoms thus forming a coordination sphere [19]. These five Zn-coordinating residues were implicated in site-mutagenesis experiments resulting -with exception of the specific Asp387Asn substitution-, in no detectable enzymatic activity or an enzymatic activity too low for determination of kinetic parameters of PSMA mutants. The importance of these amino acids for PSMA activity is further noted in the fact that they are highly conserved in PSMA orthologs from plants to mammals [17,25].

Zn atoms are crucial for the enzymatic function of PSMA and for the design of inhibitors. In fact, the presence of a zinc-binding group is a fundamental feature in the design of high-affinity inhibitors of PSMA; among the substances that exhibit these groups are thiols, phosphonates, hydroxamates, phosphinates, phosphoamidates, ureas, and sulfonamides [19]. Once established the crucial role of Zn atoms, this work elucidates the differences in the free energy profile and four structural parameters of PSMA with the two Zn atoms as cofactors and without them: RMSD and RMSF by atom and amino acid residue, radius of gyration and SASA.

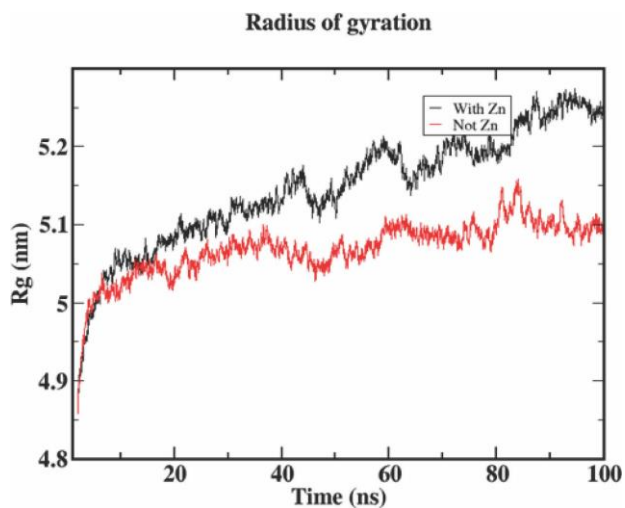


Figure 4. Radios of gyration (Rg) graphic over time showing the results from the simulation of the protein with and without the zinc atoms in the active site.

Radius of gyration gives an idea about the compactability of the molecule during time, since it provides an average of the expansion of its atoms with respect to its center of mass. When comparing the PSMA simulated with and without the zinc ions, it is visible that the PSMA with de zinc ions became more expanded over time than the other [Figure 4], consequently the zinc ions make the structure more receptive to the substrate/inhibitor by adopting an

expanded geometrical conformation that allows its entrance. On the other hand, the RMSF (which is associated with flexibility) shows that the zinc coordinating amino acids of the Zn(1): His-553 and Glu-425, including Asp-387, are more flexible when simulated with the ions, while in the case of Zn(2): His-377 and Asp453 the flexibility is slightly higher [Figure 5]. This information is consistent with the fact that Zn(1) is consider the catalytic ion and as a result the coordinating amino acids need more flexibility for the interaction with the substrate/inhibitor, whilst Zn(2) is considered co-catalytic and the amino acids may not need to be so flexible [19].

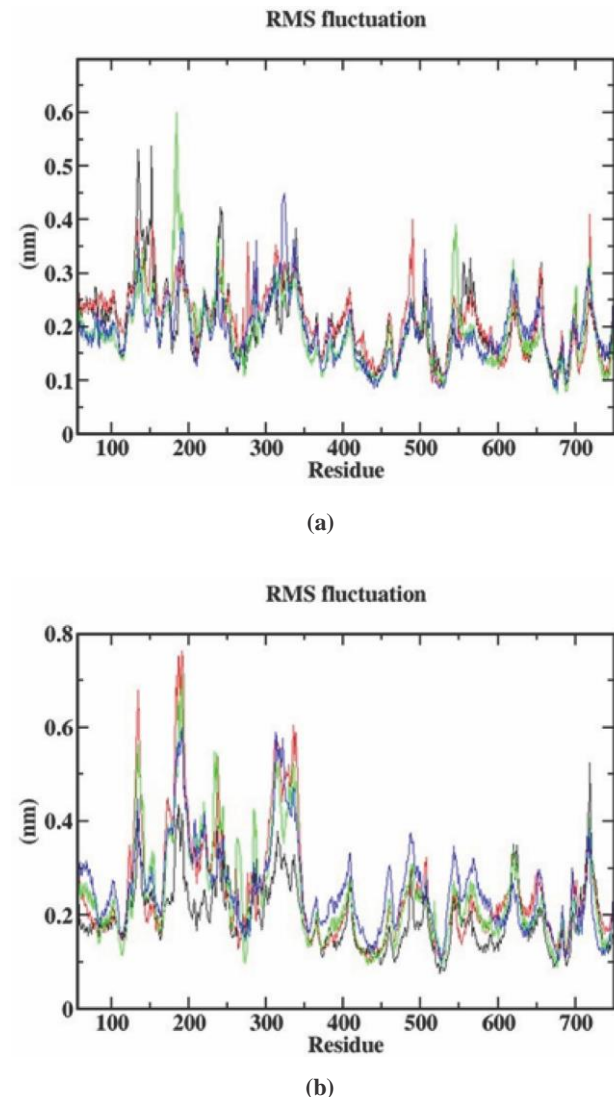


Figure 5. Root Mean Square Fluctuation (RMSF) graphic. The RMSF is related to flexibility and is calculated for each of the amino acids conforming the PSMA, the resulting variations are shown. a) PSMA without zinc and b) PSMA with the zinc ions.

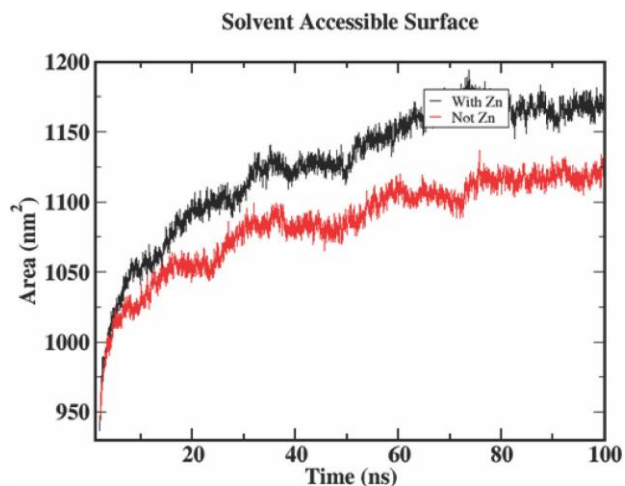


Figure 6. Solvent Accessible Surface Area (SASA) graphic showing the results over time for the PSMA with and without zinc ions.

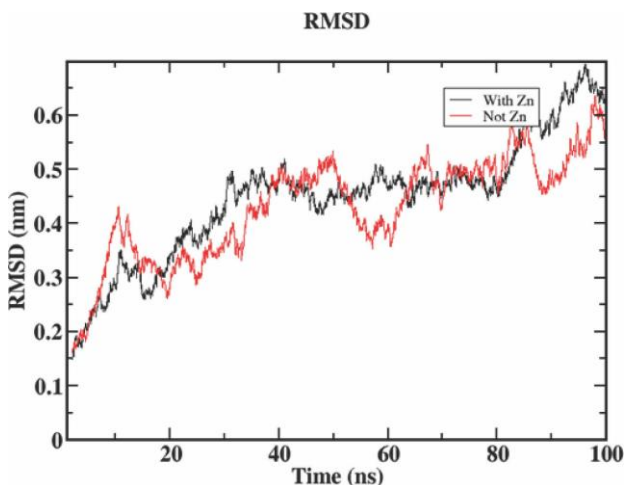


Figure 7. Root Mean Square Deviation (RMSD) graphic. The RMSD is related to structure stability of the atoms or residues over time of PSMA with and without the zinc ions.

In the RMSF, significant changes in flexibility between the two conditions are not observed in the substrate binding cavity (residues 687-704) and in the "glutarate sensor" (residues 692-704) indicating that the zinc atoms have no impact in these regions. The same situation was observed in the arginine patch (Arg-534, Arg-536 and Arg-463) meaning that the zinc ions do not play an important role in the right orientation of the substrate/inhibitor since the arginines are responsible of that task [17]. However, in Tyr-552 and Glu-424, to which catalytic activity and substrate recognition have been attributed respectively, more flexibility was observed in PSMA with the zinc ions confirming the significance of the interactions between the zinc ions and these amino acids in the active site [18]. The

results from SASA are consistent with R_g , accordingly with the fact that the PSMA with the zinc ions is more expanded, it also has more solvent accessible surface area conferring it more free energy [Figure 6]. The RMSD shows that the PSMA with the zinc ions has more structural stability as it has less variations during time compared with the simulation without them. Therefore, the zinc ions are important for the structural stability of the protein [Figure 7].

4. Conclusions

In silico analysis of the PSMA showed that its structural stability has a direct and intrinsic dependence of its active center, in fact, the tests performed on the protein excluding the Zn atoms are conclusive to affirm that PSMA not only loses affinity to bind with molecules like inhibitors, also their energy capacity is diminished; however, there are regions between residues 100-122 and 640-730 that exhibit structural stability regardless of the absence of the Zn atoms. The tests carried out on PSMA with and without the heart of its active center, the Zn atoms, allowed us to conceive the protein as a closed system, with less energy and flexibility (in the absence of atoms). This situation is completely opposite when they are incorporated, which can be interpreted that PSMA behaves like a biological trap dependent on Zn. The CHARMM force field used was appropriate for modeling the PSMA interactions and can be adapted to study the processes involved in the active center in the presence of a urea-based inhibitor and, of course, a therapeutic radiopharmaceutical.

Acknowledgments

The authors wish to thank, COPARMEX ESTADO DE MEXICO and UNIVERSIDAD MEXIQUENSE for scholarship support and Dra. Lorena Romero Salazar from Nanothermodynamics and Complex Systems Laboratory by granting facilities for the timely development of the project.

References

- [1] M. Ervik, F. Lam, J. Ferlay, L. Mery, I. Soerjomataram, F. Bray, Lyon, France: *International Agency for Research on Cancer. Cancer Today*(2016). <http://gco.iarc.fr/today>. Accessed 10 April 2018.
- [2] R. Carter, A. Feldman, J. Coyle, *Proceedings of The National Academy of Sciences*, **93**(2), 749-753 (1996). <https://doi.org/10.1073/pnas.93.2.749>
- [3] M. N. Pangalos J. M. Neefs, M. Somers, P. Verhasselt, M. Bekkers, L. van der Helm *et al.*, *Journal of Biological Chemistry*, **274**(13), 8470-8483 (1999). <https://doi.org/10.1074/jbc.274.13.8470>

- [4] J. T. Pinto, B. P. Suffoletto, T. M. Berzin, C. H. Qiao, S. Lin, W. P. Tong *et al.*, *Clinical Cancer Research*, **2(9)**, 1445-1451 (1996).
- [5] R. Lapidus, C. Tiffany, J. Isaacs, B. Slusher, *Te Prostate*, **45(4)**, 350-354 (2000).
[https://doi.org/10.1002/1097-0045\(20001201\)45:4<350::AID-PROS10>3.0.CO;2-U](https://doi.org/10.1002/1097-0045(20001201)45:4<350::AID-PROS10>3.0.CO;2-U)
- [6] J. Troyer, M. Beckett, G. Wright, *International Journal of Cancer*, **62(5)**, 552-558 (1995).
<https://doi.org/10.1002/ijc.2910620511>
- [7] A. P. Kiess, S. R. Banerjee, R. C. Mease, S. P. Rowe, A. Rao, C. A. Foss *et al.*, *European Journal of Nuclear Medicine and Molecular Imaging*, **59(3)**, 241 (2015).
- [8] S. Litje, S. Heskamp, A. S. Cornelissen, T. D. Poeppel, S. A. van den Broek, S. Rosenbaum-Krumme *et al.*, *Teranostics*, **5(12)**, 1388 (2015).
<https://doi.org/10.7150/thno.13348>
- [9] K. P. Maresca, S. M. Hillier, F. J. Femia, D. Keith, C. Barone, J. L. Joyal *et al.*, *Journal of Medicinal Chemistry*, **52(2)**, 347-357 (2008).
<https://doi.org/10.1021/jm800994j>
- [10] Z. Zhang, Z. Zhu, D. Yang, W. Fan, J. Wang, X. Li *et al.*, *Oncology letters*, **12(2)**, 1001-1006 (2016).
<https://doi.org/10.3892/ol.2016.4699>
- [11] A. Escudero-Castellanos, B. E. Ocampo-Garcia, G. Ferro-Flores, K. Isaac-Olive, C. L. Santos-Cuevas, A. Olmos-Ortiz *et al.*, *Journal of Radioanalytical and Nuclear Chemistry*, **314(3)**, 2201-2207 (2017).
<https://doi.org/10.1007/s10967-017-5555-9>
- [12] W. P. Fendler, K. Rahbar, K. Herrmann, C. Kratochwil and M. Eiber, *Journal of Nuclear Medicine*, **58(8)**, 1196-1200 (2017).
<https://doi.org/10.2967/jnumed.117.191023>
- [13] C. Kratochwil, F. Bruchertseifer, F. L. Giesel, M. Weis, F. A. Verburg, F. Mottaghy *et al.*, *Journal of Nuclear Medicine*, **57(12)**, 1941-1944 (2016).
<https://doi.org/10.2967/jnumed.116.178673>
- [14] C. M. Zechmann, A. Afshar-Oromieh, T. Armor, J. B. Stubbs, W. Mier, B. Hadaschik *et al.*, *European Journal of Nuclear Medicine and Molecular Imaging*, **41(7)**, 1280-1292 (2014).
<https://doi.org/10.1007/s00259-014-2713-y>
- [15] M. Benesova, M. Schafer, U. Bauder-Wist, A. Afshar-Oromieh, C. Kratochwil, W. Mier *et al.*, *Journal of Nuclear Medicine*, **56(6)**, 914-920 (2015).
<https://doi.org/10.2967/jnumed.114.147413>
- [16] C. Kratochwil, F. L. Giesel, M. Eder, A. Afshar-Oromieh, M. Benesova, W. Mier *et al.*, *European Journal of Nuclear Medicine and Molecular Imaging*, **42(6)**, 987 (2015).
<https://doi.org/10.1007/s00259-014-2978-1>
- [17] M. I. Davis, M. J. Bennett, L. M. Thomas, P. J. Bjorkman, *Proceedings of the National Academy of Sciences*, **102(17)**, 5981-5986 (2005).
<https://doi.org/10.1073/pnas.0502101102>
- [18] J. R. Mesters, C. Barinka, W. Li, T. Tsukamoto, P. Majer, B. S. Slusher *et al.*, *Te EMBO journal*, **25(6)**, 1375-1384 (2006).
<https://doi.org/10.1038/sj.emboj.7600969>
- [19] J. Pavlicek, J. Ptacek, C. Barinka, *Current Medicinal Chemistry*, **19(9)**, 1300-1309 (2012).
<https://doi.org/10.2174/092986712799462667>
- [20] S. R. Banerjee, C. A. Foss, M. Castanares, R. C. Mease, Y. Byun, J. J. Fox *et al.*, *Journal of Medicinal Chemistry*, **51(15)**, 4504-4517 (2008).
<https://doi.org/10.1021/jm800111u>
- [21] S. M. Hillier, K. P. Maresca, F. J. Femia, J. C. Marquis, C. A. Foss, N. Nguyen *et al.* *Cancer Research*, **69(17)**, 6932-6940 (2009).
<https://doi.org/10.1158/0008-5472.CAN-09-1682>
- [22] P. Mlcochova, A. Plechanovova, C. Bafinka, D. Mahadevan, J. W. Saldanha, L. Rulisek, J. Konvalinka, *Te FEBS Journal*, **274(18)**, 4731-4741 (2007).
<https://doi.org/10.1111/j.1742-4658.2007.06021.x>
- [23] C. Barinka, Y. Byun, C. L. Dusich, S. R. Banerjee, Y. Chen, M. Castanares *et al.*, *Journal of Medicinal Chemistry*, **51(24)**, 7737-7743 (2008).
<https://doi.org/10.1021/jm800765e>
- [24] D. Ferraris, K. Shukla, T. Tsukamoto, *Current Medicinal Chemistry*, **19(9)**, 1282-1294 (2012).
<https://doi.org/10.2174/092986712799462658>
- [25] H. S. Speno, R. Luthi-Carter, W. L. Macias, S. L. Valentine, A. R. Joshi, J. T. Coyle, *Moeluclear Pharmacology*, **55(1)**, 179-185 (1999).



6.4 Confirmación de artículo enviado

Gmail - [High-Throughput] Manuscript ID: high-throughput-394932 - ... <https://mail.google.com/mail/u/2?ik=917d0f78b9&view=pt&search=all...>



Luis Rolando Guarneros Nolasco <luisguarneros@gmail.com>

[High-Throughput] Manuscript ID: high-throughput-394932 - Submission Received

1 mensaje

Editorial Office <high-throughput@mdpi.com>

8 de noviembre de 2018, 1:54

Responder a: high-throughput@mdpi.com

Para: Felipe Rodriguez <fjrr@uaemex.mx>

Cc: Luis Guarneros-Nolasco <luisguarneros@gmail.com>, "Ketzasmin A. Terrón-Mejía" <ket.a.t.m@gmail.com>, Jorge Mulia-Rodríguez <jmr@uaemex.mx>, Daniel Osorio-Gonzalez <dog@uaemex.mx>, Roberto López-Rendón <roberto.lopez_rendon@gmail.com>, Felipe Rodriguez-Romero <fjrr@uaemex.mx>

Dear Dr. Rodriguez,

Thank you very much for uploading the following manuscript to the MDPI submission system. One of our editors will be in touch with you soon.

Journal name: High-Throughput

Manuscript ID: high-throughput-394932

Type of manuscript: Article

Title: High-throughput of measure-preserving integrators for constant temperature molecular dynamics simulations on GPUs

Authors: Luis Guarneros-Nolasco, Ketzasmin A. Terrón-Mejía, Jorge Mulia-Rodríguez, Daniel Osorio-Gonzalez, Roberto López-Rendón, Felipe Rodriguez-Romero *

Received: 8 November 2018

E-mails: luisguarneros@gmail.com, ket.a.t.m@gmail.com, jmr@uaemex.mx, dog@uaemex.mx, roberto.lopez_rendon@gmail.com, fjrr@uaemex.mx

You can follow progress of your manuscript at the following link (login required):

http://susy.mdpi.com/user/manuscripts/review_info/aa2f465a14c16de79ac1676faf3de828

The following points were confirmed during submission:

1. High-Throughput is an open access journal with publishing fees of 350 CHF for an accepted paper (see <http://www.mdpi.com/about/apc/> for details). This manuscript, if accepted, will be published under an open access Creative Commons CC BY license (<https://creativecommons.org/licenses/by/4.0/>), and I agree to pay the Article Processing Charges as described on the journal webpage (<http://www.mdpi.com/journal/high-throughput/apc>). See <http://www.mdpi.com/about/openaccess> for more information about open access publishing.

Please note that you may be entitled to a discount if you have previously received a discount code or if your institute is participating in the MDPI Institutional Open Access Program (IOAP), for more information see <http://www.mdpi.com/about/ioap>. If you have been granted any other special discounts for your submission, please contact the High-Throughput editorial office.

2. I understand that:

a. If previously published material is reproduced in my manuscript, I will provide proof that I have obtained the necessary copyright permission. (Please refer to the Rights & Permissions website: <http://www.mdpi.com/authors/rights>).



b. My manuscript is submitted on the understanding that it has not been published in or submitted to another peer-reviewed journal. Exceptions to this rule are papers containing material disclosed at conferences. I confirm that I will inform the journal editorial office if this is the case for my manuscript. I confirm that all authors are familiar with and agree with submission of the contents of the manuscript. The journal editorial office reserves the right to contact all authors to confirm this in case of doubt. I will provide email addresses for all authors and an institutional e-mail address for at least one of the co-authors, and specify the name, address and e-mail for invoicing purposes.

If you have any questions, please do not hesitate to contact the High-Throughput editorial office at high-throughput@mdpi.com

Kind regards,

High-Throughput Editorial Office
St. Alban-Anlage 66, 4052 Basel, Switzerland
E-Mail: high-throughput@mdpi.com
Tel. +41 61 683 77 34
Fax: +41 61 302 89 18

*** This is an automatically generated email ***

Article

High-throughput of measure-preserving integrators for constant temperature molecular dynamics simulations on GPUs

Luis Guarneros-Nolasco ^{1,5}, Ketzasmin A. Terrón-Mejía ⁵, Jorge Mulia-Rodríguez ³, Daniel Osorio-González ⁴, Roberto López-Rendón ¹, Felipe Rodríguez-Romero ^{5,*}

¹ Laboratorio de Bioingeniería Molecular a Multiescala, Facultad de Ciencias, Universidad Autónoma del Estado de México, Av. Instituto Literario 100, Toluca 50000, Mexico; luisguarneros@gmail.com (L.G.-N.); roberto.lopez.rendon@gmail.com (R.L.-R.)

² Laboratorio de Venómica Computacional, Facultad de Ciencias, Universidad Autónoma del Estado de México, Av. Instituto Literario 100, Toluca 50000, Mexico; feliperfr@yahoo.com.mx (F.R.-R)

³ Laboratorio de Biotermodinámica Computacional, Facultad de Ciencias, Universidad Autónoma del Estado de México, Av. Instituto Literario 100, Toluca 50000, Mexico; jmr@uaemex.mx (J.M.-R)

⁴ Laboratorio de Biofísica Molecular, Facultad de Ciencias, Universidad Autónoma del Estado de México, Av. Instituto Literario 100, Toluca 50000, Mexico; dog@uaemex.mx (D.O.-G)

⁵ Instituto Tecnológico Superior de Zongolica, Km. 4 Carretera a la Compañía, Zongolica, Veracruz 95005, Mexico; luisguarneros@gmail.com (L.G.-N.); ket.a.t.m@gmail.com (K.A.T.-M.)

* Correspondence: feliperfr@yahoo.com.mx; Tel.: +52 722 296 5556

Abstract: Molecular dynamics simulation is currently the theoretical technique eligible to simulate a wide range of systems from soft condensed matter to biological systems. However, of the excellent results that the technique has arrogated, this approach remains computationally expensive, but with the emergence of the new supercomputing technologies based on graphics processing units graphical processing units-based systems GPUs, the perspective has changed. The GPUs allow performing large and complex simulations at a significantly reduced time. In this work, we present recent innovations in the acceleration of molecular dynamics in GPUs to simulate non-Hamiltonian systems. In particular, we show the performance of measure-preserving geometric integrator in the canonical ensemble, that is, at constant temperature. We provide a validation and performance evaluation of the code by calculating the thermodynamic properties of a Lennard-Jones fluid. Our results are in excellent agreement with reported data reported from literature, which were calculated with CPUs. The scope and limitations for performing simulations of high-throughput MD under rigorous statistical thermodynamics in the canonical ensemble are discussed and analyzed.

Keywords: High-Throughput Molecular Dynamics (HTMD) simulations; Nosé-Hoover Chain (NHC); Canonical ensemble; Graphics Processing Units (GPUs).

1. Introduction

The accelerated use of molecular simulation methods, such as molecular dynamics (MD) simulations, has motivated an increasing interest in the implementation of rigorous algorithms in modern supercomputing technologies, which is a difficult task. Due to the purely numerical nature of MD method, which emerge from the laws of statistical mechanics, they require the use of efficient schemes to sample the phase space, maintaining control of thermodynamic variables at large scales of time and length. The combination of powerful computational technologies and the strict laws of statistical mechanics placed in a same simulation code is a great challenge nowadays.

From a conceptual perspective, MD simulations (based on the integration of Newton's equations motion) are considered as numerical experiments that provides the means to sample the thermodynamic space of a given system through its evolution dynamical. In MD simulations, these

spaces or thermodynamic ensembles are characterized according to the type of physical variables that are to be evaluated. For example, the simplest case is where it is required to keep the energy constant, this case is the so-called microcanonical ensemble (NVE), where the thermodynamic condition is to maintain the energy constant (E), as well as the volume (V) and number of particles (N). However, for comparisons with experimental conditions, it is often convenient to perform simulations at constant temperature, which leads us to have a system in the canonical ensemble (NVT). In this case, the thermodynamic condition is to maintain the temperature constant (T), as well as the volume (V) and number of particles (N). The NVT ensemble is somewhat more difficult to generate than the NVE ensembles due to the requirement that the kinetic energy fluctuations must generate the correct distribution function corresponding to the ensemble. Temperature control algorithms comprise one of the fundamental parts in MD simulations.

Generally, when the Newtonian MD scheme is modified to maintain the temperature constant it is through the tools called thermostat algorithms (TAs). An efficient thermostat must meet certain requirements required of the statistical mechanics. For example, give a canonical distribution of speeds, and be ergodic, etc. Not only averages, but also fluctuations (dispersions) away from the average are also important. Several strategies to improve simulations at the canonical ensemble have been suggested in the literature [1-5]. The simplest method is through the periodic scaling of velocities; this method is economically and numerically stable. However, it does not guarantee that a canonical space distribution is obtained, but they are still commonly used. The lack of ergodicity for some thermostat algorithms or for different temperature ranges can lead to a dynamic that does not correctly reproduce the expected thermodynamic properties [6-9].

Simulating a thermal bath is not a simple task. To achieve an accurate and efficient integration in the constant-temperature ensemble, additional variables may be added to create a "thermostat" that controls the temperature of the simulated system [10]. From this idea arise the so-called extended Hamiltonian methods or Non-Hamiltonian methods [11-14]. These variables or extended degrees of freedom regulate the time-averaged values of temperature in order to reproduce as accurately as possible the phase space in the considered ensemble. However, the use of this type of methods, although accurate, should be used with care, for example, the mass of extended variables must be carefully chosen as they affect spontaneous fluctuations in the system. Lippert et al. exposed computational challenges that involve controlling the temperature of a system using external degrees of freedom [15]. Two fundamental observations were made. One of them was the high computational cost that carries the instantaneous updates of extended degrees of freedom. Second, observation to the wide separation between the timescales associated with the extended degrees of freedom and those associated with particle motion. The extended degrees of freedom generally evolve on timescales much longer than those of particle motion; this can result in numerical inaccuracy if the numerical precision employed is insufficient.

Several TAs have been developed to make use of the extended phase space of the system of interest by adding an extra dimension that takes into account the interaction of the system with the environment. The most popular TAs that are based on the extended phase space are the Berendsen [16], Andersen [17] and Nosé-Hoover (NH) methods [18,19]. Undoubtedly, the NH method is currently the most popular isothermal simulation method, however it has a drawback that their equations are non-Hamiltonian in structure due to the non-canonical coordinated transformation, which precludes the use of symplectic integration schemes [20], the NH thermostat has an ergodicity problem. The failure of this approach under the conditions usually used in molecular dynamics calculations was clarified by Martyna et al. [21]. They proposed that the shortcomings of the NH algorithm could be overcome by connecting a chain of thermostats to construct a NH chain (NHC). The figure 1 schematically shows the construction of the NHC method.

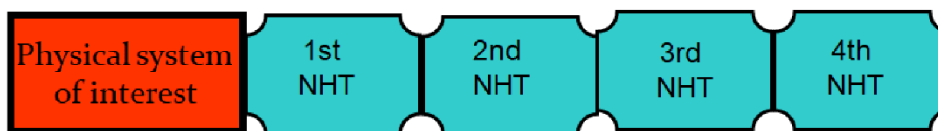


Figure 1. Schematic representation of the NHC method and its operating. The temperature of the physical system (red box) is controlled by the mechanism of the NHC equations (all four green boxes). That is, the first Nosé-Hoover thermostat (1st NHT) in the chain is coupled to the physical system, the second (2nd NHT) is coupled to the first Nosé-Hoover, the third (3rd NHT) is coupled to the second thermostat, the (4th NHT) is coupled to the third thermostat Nosé-Hoover, and so on, until forming a collective system of Nosé-Hoover chains (NHC).

When using non-Hamiltonian equations of motion, such as NHC equations, the numerical integrator should be consistent with a non-Hamiltonian generalization of Liouville's theorem [22, 23]. That means that the integrator must be symplectic, that is, must conserve exactly the invariant phase-space volume, must have an approximate energy conservation, and time reversibility between other properties [24]. A numerical integrator that achieves this is known as a "measure-preserving" algorithm. These numerical algorithms are based on decompositions of exponential operators, and the error in the total energy of the system is bounded [25]. Also, these algorithms represent the explicit integration of non-Hamiltonian dynamical systems. So far, Tuckerman et al. [26] have developed a consistent classical statistical theory of certain non-Hamiltonian dynamical systems and have developed a "systematic way" of designing equations of motion of extended molecular dynamics which are known as MTK equations, from Martyna, Tobias and Klein authors.

MTK algorithms are already applicable to a wide range of scientific problems and have been expanding their applications in several areas, such as: Physical Chemistry [27], Theoretical and Computational Chemistry [28], Materials Engineering [29], Macromolecular and Materials Chemistry [30], Biochemistry [31], Cellular and Cell Biology science [32], and so on. The implementation of the MTK algorithms is not trivial and its evaluation is computationally expensive. So far, major MD codes such as LAMMPS [33], AMBER [34], HOOMD-blue [35] and GROMACS [36] have implemented the MTK algorithms, being LAMMPS the most used code with these algorithms. But nevertheless, these codes were designed and work in programming environments in CPUs. On the other hand, with the advent of graphics processing units (GPUs), the field of High-throughput MD (HTMD) has become a fundamental tool for pharmaceutical research [37], accelerate the innovation in materials research [38], in revolutionizing the genome-wide [39], to explore large effects of structural changes ensemble of proteins [40], for drug discovery [41], just to mention a few applications. However, these areas have been developed using huge and expensive CPU clusters. Unfortunately, the generation of HTMD codes designed exclusively to perform molecular simulations on GPUs is very scarce. ACEMD [42] is the only code of MD specially optimized to run on GPUs and it is one of the world's fastest molecular dynamics engines. Nevertheless, ACEMD works with Langevin type thermostats, and to date, has not yet implemented the MTK algorithms.

The state of the art of HTMD reports that little effort has been directed to the efficient implementation of MTK algorithms on GPUs. The motivation for the use of integrators that explicitly preserve the measurement of the space phase arises as a need to have an efficient tool that guarantees ergodic sampling at large scales of time and length, this is only possible today using GPUs. In this work, we show the performance of measure-preserving geometric integrator in the canonical ensemble on GPUs, under the scheme of the MTK algorithms, which are the basis of the NHC thermostat. The capacity of our HIMD code can simulate more than one million Lennar-Jones type point particles achieving an average of 0.27 ns/day. In smaller systems, of the order of 30,000 particles, we achieve a production of 30 ns/day, it should be noted that most of the simulation results

reported today are of this number of atoms, but with a production of ns/day of half or less than half. So far, no code with those capabilities has been reported using NHC thermostat.

2. Methodology

It is well-known that a symplectic integrator conserves the Hamiltonian and therefore achieves stable integration over a long time. This is because there exists a value that is exactly conserved by the approximated propagator. These numerical methods are based on the decompositions of exponential operators. A system that preserve the generalized phase-space metric, it will be consistent with the generalization of Liouville's theorem. The details of Liouville operator algorithms derivation for non-Hamiltonian systems, such as NHC thermostat, can be found in original papers sources [43-46], therefore we shall only outline the main NHC equations.

In order to generate a sampling of the canonical distribution (NVT), Hamilton's equations must be supplemented by a mechanism that allows the system to exchange energy with its surroundings. One popular method for mimicking the influence of the surroundings in MD is so called extended phase space approach. As already mentioned in the Introduction section, Nosé-Hoover method is one of the best schemes to generate a canonical distribution but has ergodicity problems. The failure of this approach was solved by Martyna et al. introducing the NCH thermostat [21]. The main idea behind the NCH thermostat is that the physical position and momentum variables of the particles in the system are coupled to additional phase space variables that mimic the effect of the surroundings by controlling the fluctuations in the instantaneous kinetic energy.

The NHC thermostat [21] is a non-Hamiltonian MD scheme for generating the canonical ensemble. In this method, the ordinary phase space is extended to include a set of M thermostat variables η_1, \dots, η_M and their conjugate momenta $p_{\eta_1}, \dots, p_{\eta_M}$, which serve to drive the fluctuations of the kinetic energy in such a way that they average to the proper canonical value. These variables act as a heat bath coupled to the system. The equations of motion for an NHC system are

$$\begin{aligned} \dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i} \\ \dot{\mathbf{p}}_i &= \mathbf{F}_i - \frac{p_{\eta_1}}{Q_1} \mathbf{p}_i \\ \dot{\eta}_k &= \frac{p_{\eta_k}}{Q_k} \quad k = 1, \dots, M \\ p_{\eta_k} &= G_k - \frac{p_{\eta_{k+1}}}{Q_{k+1}} p_{\eta_k} \quad k = 1, \dots, M-1 \\ \dot{p}_{\eta_M} &= G_M \end{aligned} \quad (1)$$

where the thermostat forces are

$$\begin{aligned} G_1 &= \sum_{i=1}^N \frac{\mathbf{p}_i^2}{m_i} - 3Nk_bT \\ G_k &= \frac{p_{\eta_{k-1}}^2}{Q_{k-1}} - k_bT \end{aligned} \quad (2)$$

The parameter Q_1, \dots, Q_M are mass-like parameters (having units of energy*time²) that determine the time scale on which the heat-bath variables evolve, and k_b is the Boltzmann constant. The term $-(p_{\eta_1}/Q_1)\mathbf{p}_i$ in the momentum equation acts as a kind of dynamic frictional force. Although the average $\langle p_{\eta_1} \rangle = 0$, instantaneously, p_{η_1} can be positive or negative and can, therefore, act to damp or boost the momentum. According to the equation for p_{η_1} , if the kinetic energy is larger than $3Nk_bT/2$, p_{η_1} will increase and have a greater damping effect on the momenta, while if the kinetic energy is less than $3Nk_bT/2$, p_{η_1} will decrease, become negative, and have a boosting effect. In this way, the NHC system acts as a "thermostat" regulating the kinetic energy. In a similar manner, the $(k+1)$ th heat-bath variable serves to modulate the fluctuations of the k th variable so that each heat-bath variable (except the M th variable) is "driven" to have a proper canonical average. Equations (1) have the conserved energy

$$H' = H(\mathbf{p}, \mathbf{r}) + \sum_{k=1}^M \frac{p_{\eta_k}^2}{2Q_k} + k_bT \left[dN\eta_1 + \sum_{k=2}^M \eta_k \right] \quad (3)$$

where $H(\mathbf{p}, \mathbf{r})$ is the Hamiltonian of the physical system. When $M = 1$, the NHC system reduces to the simpler Nosé-Hoover system [19], which does not generate the corresponding canonical distribution.

3. Results

3.1. Code validation

In this section, we show the results of an implementation of NHC thermostat on GPUs. First, we performed the validation of the code simulating a system composed of 864 particles of a Lennar-Jones (LJ) fluid. A comparison with the data of the literature is included, we show that the equilibrium properties are the same for the simulation time considered here, but our data are calculated in GPUs while those of literature are calculated in CPUs. The details of the simulations presented can be found in appendix A of this manuscript.

The Figure 2 shows the equilibrium properties of LJ fluid in bulk phase. The comparison between literature data (calculations in CPUs) from Johnson et al. [47] and simulation data from this work (calculations in GPUs) is presented. Figure 2a shows the behavior of the potential energy as a function (U^*) of density (ρ^*) for a wide range of temperatures, from $T = 2.0$ to $T = 6.0$. These values are in reduced units. In our case, the error bars are included. Figure 2b shows the behavior of the pressure (P^*) as a function of density (ρ^*) for a wide range of temperatures, from $T = 2.0$ to $T = 6.0$. All values are in reduced units. We can see that the agreement between the data of the literature calculated in CPUs and the data calculated in this work in GPUs is excellent. This is a test of the excellent implementation of the NHC algorithm in GPUs. These results show that the canonical ensemble given by the NHC algorithm correctly produces thermodynamic properties in equilibrium, performed in GPUs as is showed in this work.

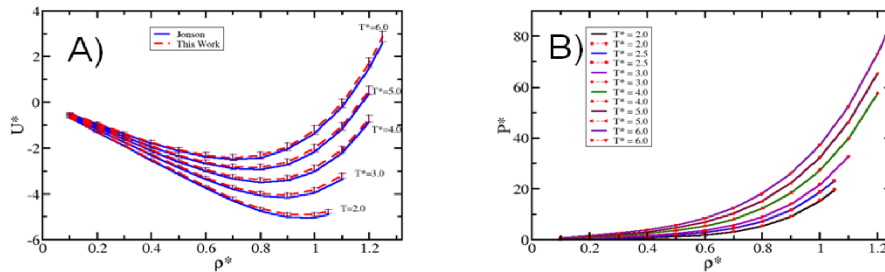


Figure 2. Equilibrium properties of Lennard-Jones fluid in bulk phase: (a) Potential energy vs. density. The blue line corresponds to literature data from Johnson et al. [47], which were calculated in CPUs, while the red dotted line are simulation data from this work but calculated in GPUs. The symbol of the error bars is included in our calculations; (b) Pressure vs. density. The continuous lines correspond to the data reported by Johnson et al. [47], while the data with symbols correspond to data obtained in this work. All units are dimensionless.

For a thermostat to be entirely consistent with the canonical ensemble, it should generate total energies according to the Boltzmann distribution for that system and generate kinetic energies consistent with the Maxwell-Boltzmann distribution [23]. To verify this rule, first we evaluate the evolution of the conserved quantity (ΔU) given by equation (3) for a LJ fluid at $T^* = 2.0$, $\rho^* = 0.7$, and $N = 864$. The result is displayed in Fig. 3A. The instantaneous (ΔU) has the definition $|(U_i - U_0)/U_0|$ and the cumulative average (ΔU) is denoted by $\Delta U = \frac{1}{N_c} \sum_i |(U_i - U_0)/U_0|$, where i runs over the number of configurations N_c . U_i is the total energy of the whole system including the atoms and thermostat variables. No drift in the conserved quantity was observed in any of the simulations as we expected. Fig. 3B shows a symmetric density distribution for this system.

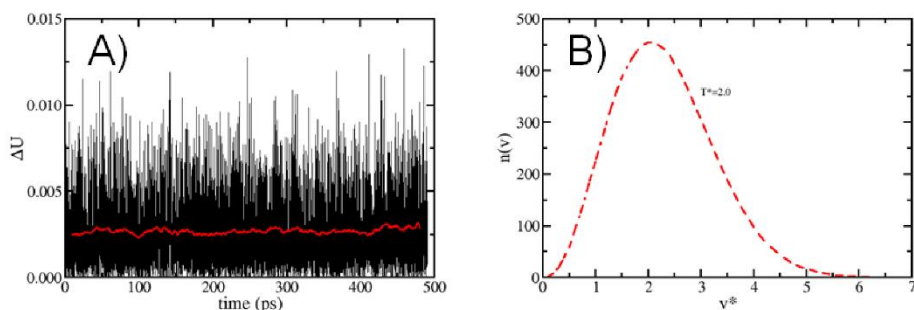


Figure 3. (a) Evolution of conserved quantity (ΔU) as a function of simulation time (ps). The red line corresponds to average energy while the black line corresponds to instant energy; (b) Histogram of cumulative momentum distribution obtained from a trajectory calculated using NHC method. For this case $T^* = 2.0$, $\rho^* = 0.7$, and $N = 864$.

The NHC motion equations increase the calculation time and the complexity due to the coupling of the thermostats; this causes the ergodicity of the dynamics to increase when increasing the available phase space of the dynamics. One of the main goals of using GPUs in MD simulations is to break with the time/length scales, that is, to simulate larger systems and longer times, preserving the correct application of the laws of statistical mechanics. To show the effectiveness of the NHC algorithm in the canonical assembly, we analyze the evolution of the temperature reduce (T^*), the kinetic energy reduce (K^*), as well as the distribution of the kinetic energy reduce $f(K^*)$. The results are show in the figure 4. We can see that $f(K^*)$ obeys to the correct distribution dictated by the canonical assembly [25]. This result shows a good agreement with similar results in the evaluation of geometrical properties that are applied to the algorithms designed to maintain the temperature in the canonical ensemble [3]. It is important to emphasize the conservation of temperature, which shows any no drift. The calculations on CPUS presented in this section were performed with an in-house code.

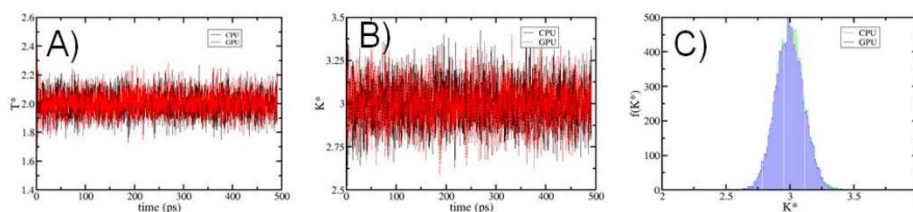


Figure 4. Sampling of temperature between CPUS and GPUs platforms; (a) Evolution of the reduced temperature (T^*) of the system at equilibrium; (b) The behavior of the reduced kinetic energy (K^*); (c) The numerical distribution reduced kinetic energy $f(K^*)$ for the same system is showed. The details of these simulations can be found in the Appendix A. All thermodynamics are reported in dimensionless units.

3.2. Code performance

After showing the excellent comparison of thermodynamic properties in equilibrium obtained with both CPUs and GPUs for a LJ fluid in phase bulk, we will show below the performance of our code in different GPUs architectures. The performance benchmark test is based on LJ system also. These benchmarking simulations were run on our cluster Olinka [48], which is equipped with the

latest GPU technologies. As we mentioned before, the true power of using GPUs is to simulate systems composed of a large number of atoms, as clusters based on GPU technologies allow.

To know some advantages like high efficiency, high speed and low cost of the NHC thermostat on GPUs, the execution time per step of MD and the speedup were measured, both parameters as a function of the number of particles. The result is shown in figure 5. We observe from Figure 5A, that the computational cost of time per step increases exponentially in the CPUs. As the number of particles increases, the time per step also increases. In the GPUs, the computational cost of time per step is practically constant. Speedup in Figure 5B is measured as the ratio of wall time elapsed for carrying out a specific simulation. In this case, we observe that for a system of 4000 particles a speedup of 45 is achieved, while a system of 23328 achieves a speedup of 60, therefore, the speedup achieved by going from a system of 4000 particles to a system of 23328 is of the order of 25. A speedup of this magnitude is significant because of the wide applications of MD simulations to biological systems. These tests were measured in the GPU model GTX-1080 for systems of LJ particles.

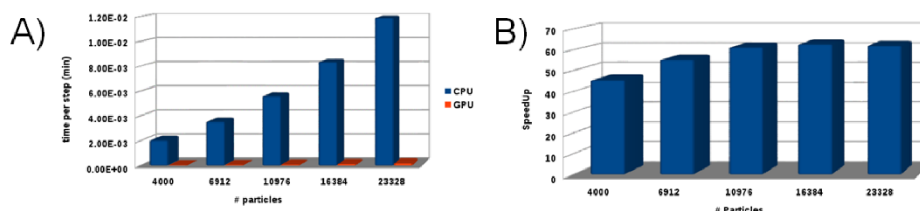


Figure 5. Performance of NHC thermostat on GPU: (a) Performance CPU vs. GPU of time per step as a function of particles number; (b) Speedup of NHC thermostat as a function of particles number. These tests were measured in the GPUs model GTX-1080 for a system of 4000, 6912, 10976, 16384 and 23328 particles. The Speedup obtained by migrate the NHC algorithms to GPUs is approach 60 times faster than the CPU version.

The performance tests of NHC thermostat were carried out on different GPUs manufactured by NVIDIA. This performance is based on all architectures of NVIDIA GPU series, from first generation Tesla (2007–09), Fermi (2010–11), Kepler (2012–14), to Maxwell (2015–16). In Figure 6, a complete analysis of the performance of NHC thermostat is shown. Figure 6 shows that the best performance of NHC thermostat is achieved with GTX 1080 GPU cards, achieving a production of 62 ns/day. It has been documented that the best MD codes running on GPUs reach their best performance on these cards [42]. This performance is achieved in single GPUs for a system of 10976 particles type LJ.

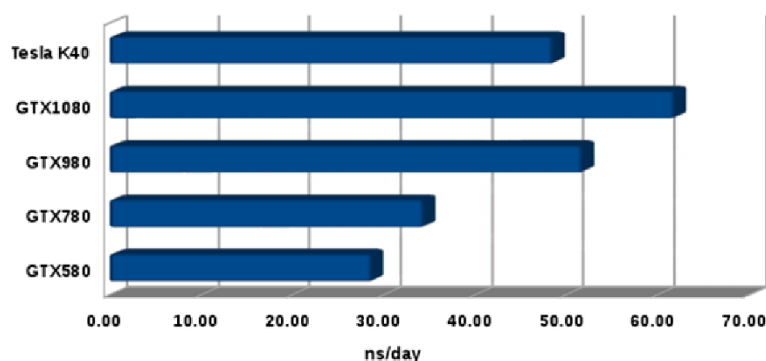


Figure 6. Performance Evaluation and Benchmarking of NHC thermostat in different GPUs architectures. The performance of our code is tested on a system of constituted by 10976 Lennard-Jones type particles.

Currently, our code can simulate up to 1.2 million particles. The benchmarking production is presented in the Figure 7. Simulating more than one million particles is a challenge that keeps many research groups busy. This size of systems is the maximum that a GPUs supports, due to the capacity of the memory. GPUs with greater memory capacity could simulate larger systems, and of course with a higher ns/day production. Figure 7A shows the decay of ns / day production as the particle number increases. This behavior is normal due to the cost of calculating the forces. The maximum size that we manage to simulate is about of one 1.2 million of LJ particles, obtaining only 0.27 ns/day on GPUs model GTX-1080, so it is seen more clearly in figure 7B. Figure 7C shows an example of the configuration of this system, which measures approximately 40 nm. That order of magnitude is of great importance where the capabilities of the HTMD are exploited.

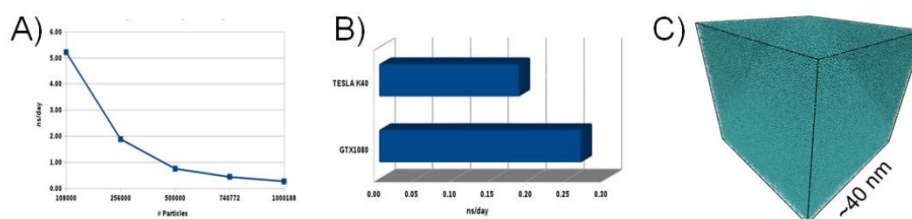


Figure 7. HTMD of NHC thermostat performance on GPUs; (a) Maximum performance on a GPUs GTX-1080; (b) Comparations of production in ns/day between two GPUs architectures to simulate one million of LJ particles; (c) A snapshot of a system composed by one million of particles.

4. Discussion

The key contribution of HTMD molecular simulations technologies is to accelerate the innovation in research both simple systems and complex systems. We think that new computational algorithms and strong collaborations between chemists, physicists, biologists, mathematicians, engineers and medical professionals will reinforce this area of research. However, the problems and challenges related to biomolecules, which are often expensive in the sense of numerical simulation, open a research window with a major difficult to answer. A question that has not been fully addressed to date is to what extent the kinetic properties can be reproduced correctly, this using MD to see extent simulations can be accelerated by these simulations methods [49].

Here, we present a new tool for enabling HTMD simulation under a strict methodology based on the NHC method. The accurate simulation in the canonical ensemble is a problem of great scientific importance. With this software we would like to perform simulations in order to be able to test predictions that serve as a point of reference in real experiments, in such a way that we can accelerate promising results. With molecular simulation methods, one is normally obliged to comply with the laws of statistical mechanics, describing with precision the statistical ensemble under study, in this case the canonical ensemble. Here we show how to combine the most advanced supercomputing technologies in conjunction with the highest technologies that generate integrators that preserve the measurements of the phase space for the canonical set at a very low computational cost. We also show the potential of GPUs that allow us to explicitly evaluate with enough precision the simulation of simple systems under the scheme of the NHC thermostat, thus exceeding the exact temperature modeling at large scales with respect to other temperature control algorithms. Accelerated molecular dynamics on GPUs as provided by our HTMD platform should be of wide interest for a large number of computational scientists as it provides performance comparable to that achievable on standard CPU supercomputers in a laboratory environment. Even research groups that have access to High-Performance Computing can find a useful tool in our code, which has the ability

to run simulations locally for longer periods of time and with greater flexibility. Finally, we see that the race to have more and better MD codes in GPUS is in constant growth, and we need more innovation in this sense.

5. Conclusions

The evolution and innovation in the development MD software on GPUs is in constant expansion. However, the use of GPUs acceleration of condensed-phase matter MD simulations is still in its infancy. The pressure to achieve maximum performance has led to the use of approximations in statistical methodologies trying to avoid a real rigorous validation. The development of MD codes in GPUs seems to be an established and extremely active field, however, very few codes can be considered ready for production and even very few achieve the desired goal of making direct comparisons with real experiments, without making approximations. However, the current benefits of GPUs are enticing, and this is driving both code and hardware development. Despite the substantial progress made in the development of the code, the difficulty in programming GPUs devices persists, programming complex algorithms such as the NHC thermostat makes some groups choose to implement simpler thermostats, which are efficient at long times, as example explicit are the NH and Langevin thermostats. With constant anticipated release of new technology NVIDIA will undoubtedly bring more competition in the development of more efficient software with more demanding implementations such as those presented in this work. It is anticipated that with the release of new versions of GPUs, MD codes will evolve rapidly, while research with these methodologies will increase exponentially in the coming years, but the limitation of implementing complex methodologies such as those presented in this paper is a latent challenge.

Author Contributions: L. G.-N. and F. R.-R. Software, Supervision and writing—original draft; J. M.-R. Methodology; D. O.-G. Methodology; R.L.-R., K.A.T.-M. and F.R.-R. Conceptualization, R.L.-R. and F.R.-R. Funding acquisition, Project administration, Resources. All authors contributed to the reading, writing, and approval of the original manuscript.

Funding: F.R.R. and R.L.R. thank the financial support from SlyEA-UAEM (Projects 3475/2013CHT and 3585/2013CHT).

Acknowledgments: All simulations reported in this work and the development of our HTMD code were performed at the Supercomputer OLINKA, located at the Laboratory of Molecular Bioengineering at Multiscale (LMBM), at the Universidad Autónoma del Estado de México. L.G.-N. acknowledges CONACyT México for supporting his PhD studies.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Code available: The code presented in this paper is available online from our web site: <http://www.himd.uaemex-labs.org.mx/>

Appendix A

Computational details. The performed simulations in this work are split in two main blocks. The first one corresponds to the validation of the code developed, and the second one regards the scope of performance obtained at the implementation of Nose-Hoover chains in GPU architecture. For the validation of the code, we reproduced some data reported by Jonson et. al. [47]. These data correspond to a fluid of Lennard-Jones particles, simulated in a canonical ensemble (NVT), the conditions of temperature and density are in range of 2.0 to 6.0 and 0.1 to 1.3 respectively. These intervals of thermodynamic conditions exclude metastable points over solidification and vaporization curves [47], while the number of particles in the system is 864. Other parameters of simulation are time step $\Delta t = 0.002$, characteristic time scale of particles $\tau_T = 0.2$, cutoff radius $r_C = 4.0\sigma$, Verlet radius $r_V = 4.5\sigma$. Units of reference selected are: size of particles $\sigma = 3.405\text{\AA}$, mass of particles $m = 39.95\text{gmol}^{-1}$ and the depth of potential $\varepsilon/k_b = 119.8K$. Simulations were equilibrated with 20000 steps and the thermodynamic properties, energy and pressure, were averaged y the stage of production, this stage consist of 40000 MD steps. We use velocity Verlet

algorithm of integration [50] and the Nose-Hoover chains as thermostat for keeping the temperature in the ensemble NVT.

Second block of simulations is focused on analyzing the performance of NHC executed in the GPU. For these simulations, it is used one system of LJ particles under the same thermodynamic parameters ($\rho = 0.7, T = 2.0$), but in this case the number of particles is 4000, 6912, 10976, 16384 and 23328. Next group of simulations is centered on exploring the behavior of the code in different models of the GPUs, the selected system for this test is of 10976 particles under the thermodynamics conditions. The models of GPUs used are GTX580, GTX780, GTX980, GTX1080 and Tesla K40. Finally, the last group of simulations explores the capacity with respect to the size of the systems that will be studied with this code developed. We estimated real time simulated that will reach systems with 108000, 256000, 500000, 740772 and 1000118 particles in the canonical ensemble, using the thermostat of Nose-Hoover chains. All the quantities are presented in reduced units unless otherwise indicated.

References

1. Hu, Y.; Sinnott, S.B. Constant temperature molecular dynamics simulations of energetic particle–solid collisions: comparison of temperature control methods. *Journal of Computational Physics*, 2004, 200, 251–266, doi:10.1016/j.jcp.2004.03.019.
2. Houndonoubo, Y. A.; Laird, B. B. Constant-temperature molecular-dynamics algorithms for mixed hard-core/continuous potentials *J. Chem. Phys.* **2002**, 117, 1001-1009, <http://dx.doi.org/10.1063/1.1485072>
3. Tapias, D.; Sanders, D.P.; Bravetti, A. Geometric integrator for simulations in the canonical ensemble. *J. Chem. Phys.* **2016**, 145, 084113, <http://dx.doi.org/10.1063/1.4961506>.
4. Samoletov, A.A.; Dettmann, C. P.; Chaplain, M.A.J. Thermostats for “Slow” Configurational Modes. *J. Stat. Phys.* **2007**, 128, 1321–1336. DOI: 10.1007/s10955-007-9365-2.
5. Hoover, W.G.; Sprott, J.C.; Patra, P.K. Ergodic time-reversible chaos for Gibbs’ canonical oscillator. *Physics Letters A* **2015**, 379, 2935–2940. <http://dx.doi.org/10.1016/j.physleta.2015.08.034>
6. Hoover, W.G.; Sprott, J.C.; Hoover, C.G. Ergodicity of a singly-thermostated harmonic oscillator. *Commun Nonlinear Sci Numer Simulat* **2016**, 32, 234–240. <http://dx.doi.org/10.1016/j.cnsns.2015.08.020>
7. Jepps, O.G.; Rondoni, L. Deterministic thermostats, theories of nonequilibrium systems and parallels with the ergodic condition. *J. Phys. A: Math. Theor.* **2010**, 43, 133001. doi:10.1088/1751-8113/43/13/133001.
8. Legoll, F.; Luskin, M.; Moeckel, R. Non-Ergodicity of the Nosé–Hoover Thermostated Harmonic Oscillator. *Arch. Rational Mech. Anal.* **2007**, 184, 449–463, DOI: 10.1007/s00205-006-0029-1.
9. Legoll, D.; Luskin, M.; Moeckel, R. Non-ergodicity of Nosé–Hoover dynamics *Nonlinearity* 2009, 22, 1673–1694, doi:10.1088/0951-7715/22/7/011
10. G. J. Martyna, M. E. Tuckerman, D. J. Tobias, and M. L. Klein, Explicit reversible integrators for extended systems dynamics *Mol. Phys.* 1996, 87, 1117-1157, <https://doi.org/10.1080/00268979600100761>
11. Ezra, G.S. Reversible measure-preserving integrators for non-Hamiltonian systems *J. Chem. Phys.* **2006**, 125, 034104, DOI: 10.1063/1.2215608
12. Sergi, A.; Ferrario, M. Non-Hamiltonian equations of motion with a conserved energy *Physical Review E*, 2001, 64, 056125, <https://doi.org/10.1103/PhysRevE.64.056125>
13. Ciccotti, G.; Ferrario, M. Holonomic Constraints: A Case for Statistical Mechanics of Non-Hamiltonian Systems *Computation* **2018**, 6, 11; doi:10.3390/computation6010011
14. Giusteri, G.G.; Podio-Guidugli, P.; Fried, E. Continuum balances from extended Hamiltonian dynamics *J. Chem. Phys.*, 2017, 146, 224102, <http://dx.doi.org/10.1063/1.4984823>
15. Lippert, R. A.; Predescu, C.; Ierardi, D.J.; Mackenzie, K.M.; Eastwood, M.P.; Dror, R.O.; Shaw, D. E. Accurate and efficient integration for molecular dynamics simulations at constant temperature and pressure *J. Chem. Phys.* **2013**, 139, 164106, <http://dx.doi.org/10.1063/1.4825247>
16. Berendsen, H.J.C.; Postma, J.P.M.; van-Gunsteren, W.F.; DiNola, A.; Haak, J.R. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.* **1984**, 81, 3684-3690, <https://doi.org/10.1063/1.448118>.
17. Andersen, H.C. Molecular dynamics simulations at constant pressure and/or temperature. *J. Chem. Phys.* **1980**, 72, 2384-2393. <https://doi.org/10.1063/1.439486>.
18. Nosé, S. A unified formulation of the constant temperature molecular dynamics methods. *J. Chem. Phys.* **1984**, 81, 511-519. <http://dx.doi.org/10.1063/1.447334>.



19. Hoover, W.G. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A* **1985**, *31*, 1695-1697. <https://doi.org/10.1103/PhysRevA.31.1695>.
20. Martyna, G.J.; Tuckerman, M.E. Symplectic reversible integrators: Predictor–corrector methods *J. Chem. Phys.* **1995**, *102*, 8071-8077, <https://doi.org/10.1063/1.469006>
21. Martyna, G.J.; Klein, M.L.; Tuckerman, M. Nosé–Hoover chains: The canonical ensemble via continuous dynamics. *J. Chem. Phys.* **1992**, *97*, 2635-2643. <https://doi.org/10.1063/1.463940>
22. Tuckerman, M.E.; Mundy, C.J.; Martyna, G.J. On the classical statistical mechanics of non-Hamiltonian systems. *Europhys. Lett.* **1999**, *45*, 149-155, <https://doi.org/10.1209/epl/i1999-00139-0>
23. Tuckerman, M.E.; Liu, Y.; Ciccotti, G.; Martyna, G.J. Non-Hamiltonian molecular dynamics: Generalizing Hamiltonian phase space principles to non-Hamiltonian systems *J. Chem. Phys.* **2001**, *115*, 1678-1702, <https://doi.org/10.1063/1.1378321>
24. McLachlan, R.I.; Quispel, R.R.W. Geometric integrators for ODEs, *J. Phys. A: Math. Gen.* **2006**, *39*, 5251–5285 doi:10.1088/0305-4470/39/19/S01
25. Tuckerman, M.E.; Martyna, G.J. Understanding Modern Molecular Dynamics: Techniques and Applications, *J. Phys. Chem. B* **2000**, *104*, 159-178, DOI: 10.1021/jp992433y
26. Tuckerman, M.E. *Statistical Mechanics: Theory and Molecular Simulation*, Oxford University Press, 1st Edition, 2013; 696 pages, ISBN-13: 978-0198525264.
27. Andoh, Y.; Yoshii, N.; Yamada, A.; Okazaki, S. Evaluation of atomic pressure in the multiple time-step integration algorithm **2017**, *38*, 704-713, DOI: 10.1002/jcc.24731.
28. Van Houteghem, M.; Ghysels, A.; Verstraelen, T.; Poelmans, W.; Waroquier, M.; Van Speybroeck, V. Critical Analysis of the Accuracy of Models Predicting or Extracting Liquid Structure Information *J. Phys. Chem. B*, **2014**, *118*, 2451–2470, DOI: 10.1021/jp411737s
29. Sencer Selcuk, S.; Xunhua Zhao, X.; Selloni, A.; Structural evolution of titanium dioxide during reduction in high-pressure hydrogen *Nature Materials* **2018**, *17*, 923–928, <https://doi.org/10.1038/s41563-018-0135-0>
30. Allen, D.; Lorenz, C.D. A novel method for constructing continuous intrinsic surfaces of nanoparticles *J. Mol. Model.* **2017**, *23*, 219, DOI 10.1007/s00894-017-3378-9
31. Ayee, M.A.A.; LeMaster, E.; Shentu, T.P.; Singh, D.K.; Barbera, N.; Soni, D.; Tirupathi, Ch.; Subbaiah, P.V.; Berdyshev, E.; Bronova, I.; Cho, M.; Akpa, B.S.; Levitan, I. Molecular-Scale Biophysical Modulation of an Endothelial Membrane by Oxidized Phospholipids *Biophysical Journal*, **2017**, *112*, 325–338, <https://doi.org/10.1016/j.bpj.2016.12.002>
32. Sadeghi, M.; Weik, T.R.; Noé, F. Particle-based membrane model for mesoscopic simulation of cellular dynamics *J. Chem. Phys.* **2018**, *148*, 044901, <https://doi.org/10.1063/1.5009107>
33. S. Plimpton, Fast Parallel Algorithms for Short-Range Molecular Dynamics, *J Comp Phys*, *117*, 1-19 (1995).
34. Cheng, A.; Merz, K.M. Application of the Nose'-Hoover Chain Algorithm to the Study of Protein Dynamics *J. Phys. Chem.* **1996**, *100*, 1927-1937, DOI: 10.1021/jp951968y
35. J. A. Anderson, C. D. Lorenz, and A. Travesset. General purpose molecular dynamics simulations fully implemented on graphics processing units *Journal of Computational Physics* *227*(10): 5342-5359, May 2008. 10.1016/j.jcp.2008.01.047
36. Hess B, Kutzner C, van der Spoel D, Lindahl E. GROMACS 4: Algorithms for highly efficient, load balanced, and scalable molecular simulation. *J. Chem. Theory. Comput.* **2008**, *4*, 435-447.
37. Stanley, N.; De Fabritiis, G. High throughput molecular dynamics for drug discovery *In Silico Pharmacology* **2015**, *3*, 3 DOI 10.1186/s40203-015-0007-0.
38. Erucar, I.; Keskin, S. High-Throughput Molecular Simulations of Metal Organic Frameworks for CO2 Separation: Opportunities and Challenges *Front. Mater.* **2018**, *5*, 4, doi: 10.3389/fmats.2018.00004
39. Hospital, A.; and Josep Ll Gelpi, J.Ll. High-throughput molecular dynamics simulations: toward a dynamic view of macromolecular structure *WIREs Comput Mol Sci* **2013**, doi: 10.1002/wcms.1142
40. Yamamoto, E.; Kalli, A.C.; Yasuoka, K.; Sansom, M.S.P. Interactions of Pleckstrin Homology Domains with Membranes: Adding Back the Bilayer via High-Throughput Molecular Dynamics *Structure* **2016**, *24*, 1421–1431, <http://dx.doi.org/10.1016/j.str.2016.06.002>
41. Harvey, M.J.; De Fabritiis, G. High-throughput molecular dynamics: the powerful new tool for drug discovery *Drug Discovery Today* **2012**, *17*, 1059-1062, <http://dx.doi.org/10.1016/j.drudis.2012.03.017>
42. Harvey, M.; Giupponi, G.; De Fabritiis, G. ACEMD: accelerated molecular dynamics simulations in the microseconds timescale. *J Chem Theory Comput* **2009**, *5*, 1632–1639. DOI: 10.1021/ct9000685



43. M.E.; Alejandre, J.; López-Rendón, R.; Jochim, A.L.; Martyna, G.J. A Liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal–isobaric ensemble. *J. Phys. A: Math. Gen.* **2006**, *39*, 5629–5651, doi:10.1088/0305-4470/39/19/S18.
44. Yu, T.-Q.; Alejandre, J.; López-Rendón, R.; Martyna, G.J.; Tuckerman, M.E. Measure-preserving integrators for molecular dynamics in the isothermal–isobaric ensemble derived from the Liouville operator. *Chem. Phys.* **2010**, *370*, 294–305, doi: 10.1016/j.chemphys.2010.02.014.
45. Yu, T.-Q.; Tuckerman, M.E. Constrained molecular dynamics in the isothermal-isobaric ensemble and its adaptation for adiabatic free energy dynamics *Eur. Phys. J. Special Topics* **2011**, *200*, 183–209, DOI: 10.1140/epjst/e2011-01524-x
46. Romero-Bastida, M.; López-Rendón, R. Anisotropic pressure molecular dynamics for atomic fluid systems. *J. Phys. A: Math. Theor.* **2007**, *40*, 8585–8598, doi:10.1088/1751-8113/40/29/026.
47. Johnson, J.K.; Zollweg, J.A.; Gubbins K.E. The Lennard-Jones equation of state revisited *Mol. Phys.* **1993**, *78*, 591-618, <https://doi.org/10.1080/00268979300100411>
48. Olinka Supercomputer. Available online: <http://www.olinka.uaemex-labs.org.mx/>.
49. Feig, M. Kinetics from Implicit Solvent Simulations of Biomolecules as a Function of Viscosity *J. Chem. Theory Comput.* **2007**, *3*, 1734-1748, DOI: 10.1021/ct7000705.
50. Swope C. W.; Andersen C. H.; Berens H. P.; Wilson R. K. A computer simulation method for the calculation of equilibrium constants for the formation of clusters of molecules: Applications to small water clusters. *J. Chem. Phys.* **1982**, *76*, 637-649. <https://doi.org/10.1063/1.442716>



CAPÍTULO 7

7.1 Discusión general

Una de las aplicaciones más importantes para la computación de GPU ha sido la aceleración de los cálculos de simulaciones de dinámica molecular basadas en la mecánica clásica. A pesar de contar con algoritmos eficientes y técnicas de procesamiento paralelo a gran escala, las demandas de la investigación biomédica requieren simulaciones en complejos biomoleculares de tamaño y sofisticación crecientes, en escalas de tiempo más largas, con mejores muestreos y con mejores campos de fuerza, así como novedosos algoritmos para su ejecución y simplificación de cálculos. Cada una de estas dimensiones crea demanda de más cómputo, algo que las GPU pueden ayudar a resolver.

Bajo este contexto se han diseñado y programado bajo la tecnología de GPU software para realizar dinámica molecular, entre los que destacan los siguientes:

Software	Descripción
NAMD [56]	Es uno de los primeros paquetes en incorporar la aceleración de GPU. Es un código paralelo de dinámica molecular diseñado para la simulación de alto rendimiento de grandes sistemas biomoleculares. Basado en los objetos paralelos de Charm ++, NAMD escala a cientos de núcleos para simulaciones típicas y más allá de 500,000 núcleos para las simulaciones más grandes. NAMD utiliza el popular programa de gráficos moleculares VMD para la configuración de simulación y el análisis de la trayectoria, pero también es compatible con AMBER, CHARMM y X-PLOR. NAMD se distribuye de forma gratuita con el código fuente.
GROMACS [57]	GROMACS es un paquete versátil para realizar dinámica molecular, es decir, simular las ecuaciones de movimiento newtonianas para sistemas con cientos o millones de partículas. Está diseñado principalmente para moléculas bioquímicas como proteínas, lípidos y ácidos nucleicos que tienen muchas interacciones enlazadas complicadas, pero dado que GROMACS es extremadamente rápido para calcular las interacciones no enlazadas (que generalmente dominan las

	<p>simulaciones) muchos grupos también lo están utilizando para la investigación en sistemas no biológicos, por ejemplo, polímeros.</p>
HOOMD-blue [58]	<p>HOOMD-blue es un software disponible de forma gratuita diseñado explícitamente para la ejecución de GPU que comprende un conjunto de herramientas de simulación de partículas de uso general. Se especializa en simulaciones de dinámica molecular de sistemas de polímeros. HOOMD-blue mantiene todos los datos de simulación dentro de la memoria de la GPU para superar el cuello de botella de transferencia de la CPU a la GPU. Se utilizan varios algoritmos y enfoques específicos de GPU, incluida la clasificación de los átomos para reducir la divergencia de las ramas, el uso efectivo de las listas de pares y las optimizaciones que aprovechan las operaciones atómicas, junto con otras características que se encuentran solo en el estado del arte de GPUs.</p>
ACEMD [59]	<p>Es un software de dinámica molecular de nivel de producción especialmente optimizado para ejecutarse en unidades de procesamiento de gráficos (GPU) NVIDIA y es uno de los motores de dinámica molecular más rápidos del mundo. Este software cuenta con una potente interfaz de scripts y extensiones en Python que utiliza HTMD, permite el uso de los populares formatos de campos de fuerza CHARMM y AMBER sin ningún cambio, y permite la ejecución de múltiples hosts para los métodos de intercambio de réplicas.</p> <p>ACEMD se ha utilizado para realizar simulaciones de dinámica molecular de proteínas globulares y de membrana, oligosacáridos, ácidos nucleicos y polímeros sintéticos.</p>
LAMMPS [60]	<p>LAMMPS es un código clásico para dinámica molecular. Cuenta con potenciales para para materiales de estado sólido y materia blanda, así como sistemas de grano grueso o mesoscópicos. Se puede utilizar para modelar átomos o, más generalmente, como un simulador de partículas paralelas en la escala atómica, meso o continua.</p> <p>LAMMPS se ejecuta en procesadores individuales o en paralelo utilizando técnicas de paso de mensajes y una descomposición espacial del dominio de simulación. Muchos de sus modelos tienen versiones que proporcionan</p>

un rendimiento acelerado en CPU, GPU e Intel Xeon Phi. El código está diseñado para ser fácil de modificar o ampliar con nuevas funciones

Por lo anterior, una característica importante que difiere el software desarrollado HIMD es que se implementa los ensambles NVE y NVT en el algoritmo basado en los operadores de Liouville, siendo esto lo novedoso de nuestro software al incluirlo totalmente en procesamiento de GPU.

Del mismo modo, otra característica primordial se basa en que el ensamble NVT implementado en el código HIMD se programó al 100% en GPU haciendo uso de las cadenas de termostatos de Nose-Hoover, con lo cual ninguno de los anteriores softwares comentados utilizados para dinámica molecular lo implementa de esta forma, ya que algunas rutinas las trabaja aún a nivel de CPU, ocasionando con esto que el procesamiento sea lento en comparación con la propuesta de nuestro código.

Para la validación del código, se reprodujeron algunos datos reportados por Jonson et. Al [47]. Estos datos corresponden a un fluido de partículas de Lennard-Jones, simulado en un ensamble canónico. (NVT)

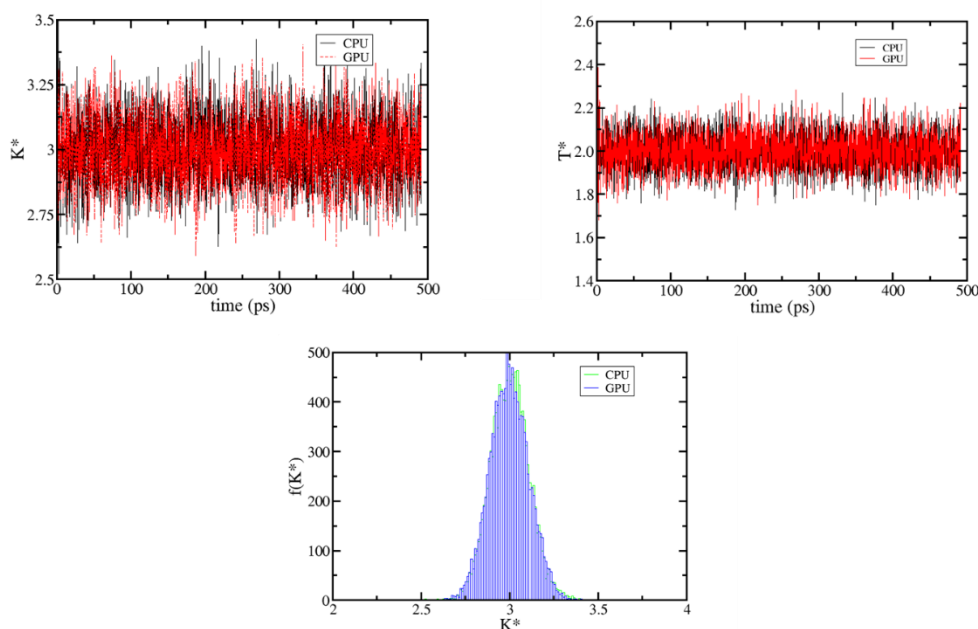


Fig. 13 Energía cinética y conservación de la temperatura de un sistema con 864 partículas a una densidad reducida de 0.7 entre CPU y GPU

Uno de los objetivos principales del uso de GPU en simulaciones de MD es romper con las escalas de tiempo / longitud, es decir, simular sistemas más grandes y tiempos más largos, preservando la aplicación correcta de las leyes de la mecánica estadística. A continuación, se muestran los resultados obtenidos al ejecutar los sistemas con diferente número de partículas, en donde se demuestra que al calcular con GPU, los tiempos de ejecución disminuyen respecto a que si se realizara en CPU. Resaltando que el desempeño mejora al trabajar con sistemas más grandes, ya que se utiliza eficientemente la arquitectura de los procesadores de las tarjetas gráficas.

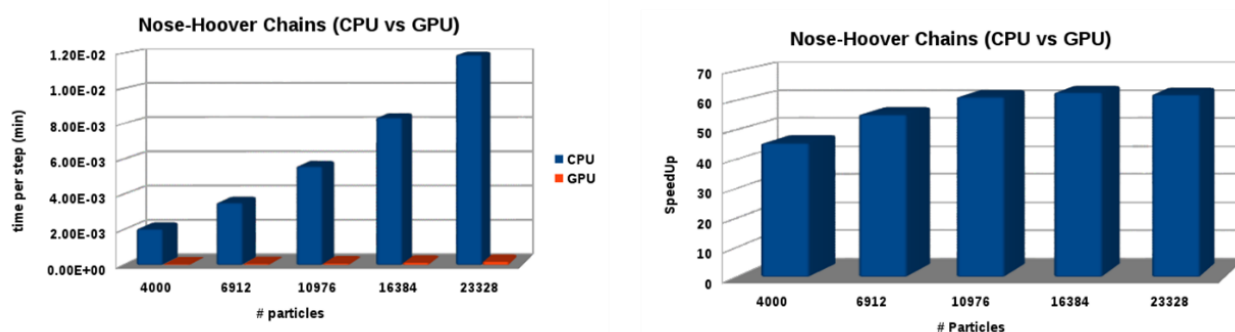


Fig. 14 En la gráfica de la izquierda se muestra el tiempo de ejecución por paso de MD mientras que, en la gráfica de la derecha se muestra la aceleración (speedup) al ejecutar las rutinas del termostato de Nose-Hoover en el GPU, ambas como función del número de partículas.

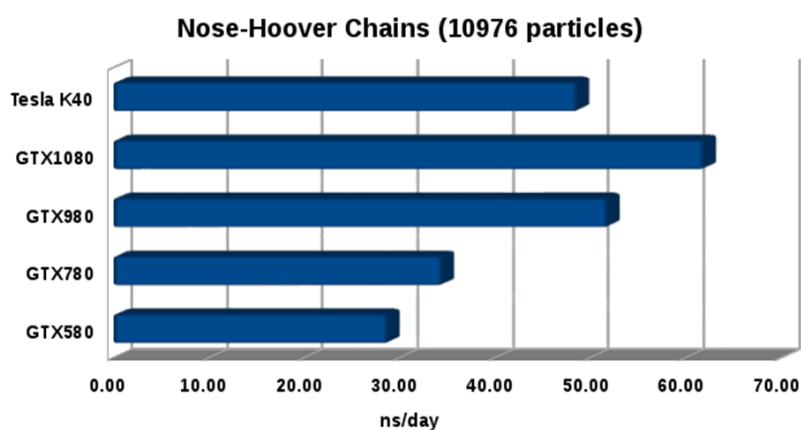


Fig. 15 Performance del código en simulaciones con sistemas de 10976 partículas en las diferentes modelos de tarjetas de Nvidia

De los múltiples códigos de simulación molecular disponibles en el mercado, LAMMPS se caracteriza por tener un alto desempeño y es uno de los pocos que implementa las Cadenas de Nose-Hoover, resaltando que es un código con varios años de desarrollo y en el cual, las cadenas se

encuentran codificadas para su ejecución en CPU. Derivado de esto, se realizó la validación de los códigos al ejecutar un sistema con 10,976 partículas de un fluido simple tipo Lennard-Jones y de los resultados obtenidos se observó que nuestro software HIMD se encuentra en concordancia con los de LAMMPS. A continuación, se muestra la temperatura instantánea del sistema.

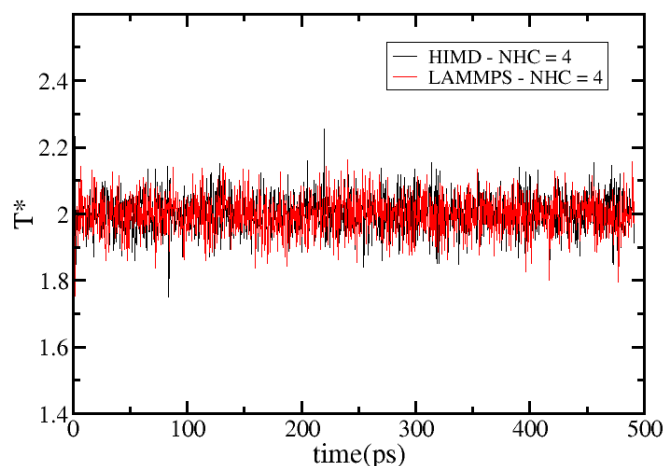


Fig. 16 Comparación de resultados de conservación de la temperatura con 4 cadenas de NHC ejecutadas en LAMMPS y HIMD.

La aportación principal de nuestro código se puede observar en los desempeños obtenidos al realizar la ejecución de comparación con el código LAMMPS al simular un sistema de 10,976 partículas en los diferentes termostatos, principalmente cuando se compara las cadenas de Nose-Hoover. A continuación, se muestra la gráfica de desempeño.

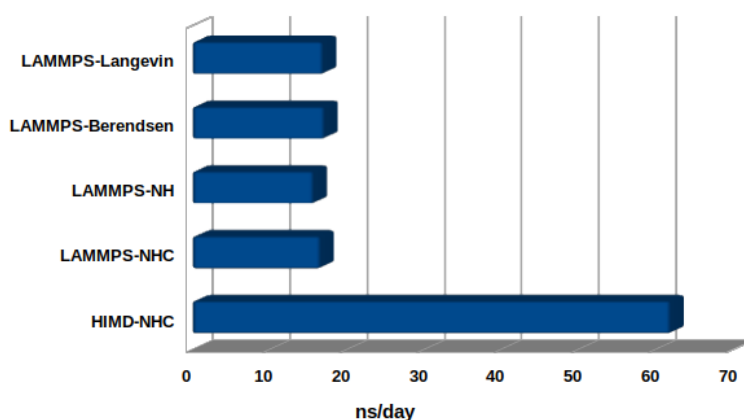


Fig. 17 Gráfica de desempeño al ejecutar las cadenas de Nose-Hoover en CPU con LAMMPS y en nuestro código HIMD (SIMAT)



CAPÍTULO 8

8.1 Conclusiones

En este trabajo hemos implementado los algoritmos de los ensambles NVE y NVT basados en el operador de Liouville y las cadenas de termostatos de Nose-Hoover en un nuevo software de dinámica molecular en el que se validaron los resultados obtenidos de las simulaciones haciendo uso de Unidades de Procesamiento Gráfico (GPU) contra lo reportado por la bibliografía, en donde resaltamos lo más importante:

El migrar hacia una tecnología de Unidades de Procesamiento Gráfico mejoró considerablemente el desempeño de los algoritmos implementados para CPU, logrando con esto una disminución en el tiempo de ejecución de las simulaciones realizadas con determinados números de partículas lo que dio origen al diseño y desarrollo del código SIMAT.

La implementación de las cadenas de Nose-Hoover para su ejecución en CPU aumenta considerablemente el tiempo de ejecución conforme crece el sistema en número de partículas. Con la implementación en GPU en SIMAT se tiene un mejor desempeño, conforme el sistema crece en el tamaño de partículas, con lo que el software ha logrado simular hasta 1.2 millones de partículas.

Demostramos que la tecnología basada en GPU permite mejorar los desempeños de cálculos en donde se requiere gran poder de cómputo, como es el caso de las simulaciones de dinámica molecular.

Una de las perspectivas de este trabajo es incluir rutinas de análisis de datos, por ejemplo, el del Root-mean-square (RMSD) o Radios of Gyration con la finalidad de contar con un software integrado para la simulación de dinámica molecular.



Referencias

1. Hu, Y.; Sinnott, S.B. Constant temperature molecular dynamics simulations of energetic particle–solid collisions: comparison of temperature control methods. *Journal of Computational Physics*, **2004**, 200, 251–266, doi:10.1016/j.jcp.2004.03.019.
2. Houndonougbo, Y. A.; Laird, B. B. Constant-temperature molecular-dynamics algorithms for mixed hard-core/continuous potentials *J. Chem. Phys.* **2002**, 117, 1001-1009, <http://dx.doi.org/10.1063/1.1485072>
3. Tapias, D.; Sanders, D.P.; Bravetti, A. Geometric integrator for simulations in the canonical ensemble. *J. Chem. Phys.* **2016**, 145, 084113, <http://dx.doi.org/10.1063/1.4961506>.
4. Samoletov, A.A.; Dettmann, C. P.; Chaplain, M.A.J. Thermostats for “Slow” Configurational Modes. *J. Stat. Phys.* **2007**, 128, 1321–1336. Doi: 10.1007/s10955-007-9365-2.
5. Hoover, W.G.; Sprott, J.C.; Patra, P.K. Ergodic time-reversible chaos for Gibbs’ canonical oscillator. *Physics Letters A* **2015**, 379, 2935–2940. <http://dx.doi.org/10.1016/j.physleta.2015.08.034>
6. Hoover, W.G.; Sprott, J.C.; Hoover, C.G. Ergodicity of a singly-thermostated harmonic oscillator. *Commun Nonlinear Sci Numer Simulat* **2016**, 32, 234–240. <http://dx.doi.org/10.1016/j.cnsns.2015.08.020>
7. Jepps, O.G.; Rondoni, L. Deterministic thermostats, theories of nonequilibrium systems and parallels with the ergodic condition. *J. Phys. A: Math. Theor.* **2010**, 43, 133001. doi:10.1088/1751-8113/43/13/133001.
8. Legoll, F.; Luskin, M.; Moeckel, R. Non-Ergodicity of the Nosé–Hoover Thermostated Harmonic Oscillator. *Arch. Rational Mech. Anal.* **2007**, 184, 449–463, DOI: 10.1007/s00205-006-0029-1.
9. Legoll, D.; Luskin, M.; Moeckel, R. Non-ergodicity of Nosé–Hoover dynamics. *Nonlinearity* **2009**, 22, 1673–1694, doi:10.1088/0951-7715/22/7/011
10. G. J. Martyna, M. E. Tuckerman, D. J. Tobias, and M. L. Klein, Explicit reversible integrators for extended systems dynamics *Mol. Phys.* **1996**, 87, 1117-1157, <https://doi.org/10.1080/00268979600100761>
11. Ezra, G.S. Reversible measure-preserving integrators for non-Hamiltonian systems *J. Chem. Phys.* **2006**, 125, 034104, DOI: 10.1063/1.2215608
12. Sergi, A.; Ferrario, M. Non-Hamiltonian equations of motion with a conserved energy *Physical Review E*, **2001**, 64, 056125, <https://doi.org/10.1103/PhysRevE.64.056125>
13. Ciccotti, G.; Ferrario, M. Holonomic Constraints: A Case for Statistical Mechanics of Non-Hamiltonian Systems *Computation* **2018**, 6, 11; doi:10.3390/computation6010011
14. Giusteri, G.G.; Podio-Guidugli, P.; Fried, E. Continuum balances from extended Hamiltonian dynamics *J. Chem. Phys.*, **2017**, 146, 224102, <http://dx.doi.org/10.1063/1.4984823>
15. Lippert, R. A.; Predescu, C.; Ierardi, D.J.; Mackenzie, K.M.; Eastwood, M.P.; Dror, R.O.; Shaw, D. E. Accurate and efficient integration for molecular dynamics simulations at constant temperature and pressure *J. Chem. Phys.* **2013**, 139, 164106, <http://dx.doi.org/10.1063/1.4825247>
16. Berendsen, H.J.C.; Postma, J.P.M; van-Gunsteren, W.F.; DiNola, A.; Haak, J.R. Molecular dynamics with coupling to an external bath. *J. Chem. Phys.* **1984**, 81, 3684-3690, <https://doi.org/10.1063/1.448118>.



17. Andersen, H.C. Molecular dynamics simulations at constant pressure and/or temperature. *J. Chem. Phys.* **1980**, 72, 2384-2393. <https://doi.org/10.1063/1.439486>.
18. Nosé, S. A unified formulation of the constant temperature molecular dynamics methods. *J. Chem. Phys.* **1984**, 81, 511-519. <http://dx.doi.org/10.1063/1.447334>.
19. Hoover, W.G. Canonical dynamics: Equilibrium phase-space distributions. *Phys. Rev. A* **1985**, 31, 1695-1697. <https://doi.org/10.1103/PhysRevA.31.1695>.
20. Martyna, G.J.; Tuckerman, M.E. Symplectic reversible integrators: Predictor–corrector methods *J. Chem. Phys.* **1995**, 102, 8071-8077, <https://doi.org/10.1063/1.469006>
21. Martyna, G.J.; Klein, M.L.; Tuckerman, M. Nosé–Hoover chains: The canonical ensemble via continuous dynamics, *J. Chem. Phys.* **1992**, 97, 2635-2643. <https://doi.org/10.1063/1.463940>
22. Tuckerman, M.E.; Mundy, C.J.; Martyna, G.J. On the classical statistical mechanics of non-Hamiltonian systems. *Europhys. Lett.* **1999**, 45, 149-155, <https://doi.org/10.1209/epl/i1999-00139-0>
23. Tuckerman, M.E.; Liu, Y.; Ciccotti, G.; Martyna, G.J. Non-Hamiltonian molecular dynamics: Generalizing Hamiltonian phase space principles to non-Hamiltonian systems *J. Chem. Phys.* **2001**, 115, 1678-1702, <https://doi.org/10.1063/1.1378321>
24. McLachlan, R.I.; Quispel, R.R.W. Geometric integrators for ODEs, *J. Phys. A: Math. Gen.* **2006**, 39, 5251–5285 doi:10.1088/0305-4470/39/19/S01
25. Tuckerman, M.E.; Martyna, G.J. Understanding Modern Molecular Dynamics: Techniques and Applications, *J. Phys. Chem. B* **2000**, 104, 159-178, DOI: 10.1021/jp992433y
26. Tuckerman, M.E. Statistical Mechanics: Theory and Molecular Simulation, *Oxford University Press, 1st Edition*, **2013**; 696 pages, ISBN-13: 978-0198525264.
27. Andoh, Y.; Yoshii, N.; Yamada, A.; Okazaki, S. Evaluation of atomic pressure in the multiple time-step integration algorithm **2017**, 38, 704-713, DOI: 10.1002/jcc.24731.
28. Van Houteghem, M.; Ghysels, A.; Verstraelen, T.; Poelmans, W.; Waroquier, M.; Van Speybroeck, V. Critical Analysis of the Accuracy of Models Predicting or Extracting Liquid Structure Information *J. Phys. Chem. B*, **2014**, 118, 2451–2470, DOI: 10.1021/jp411737s
29. Sencer Selcuk, S.; Xunhua Zhao, X.; Selloni, A.; Structural evolution of titanium dioxide during reduction in high-pressure hydrogen *Nature Materials* **2018**, 17, 923–928, <https://doi.org/10.1038/s41563-018-0135-0>
30. Allen, D.; Lorenz, C.D. A novel method for constructing continuous intrinsic surfaces of nanoparticles *J. Mol. Model.* **2017**, 23, 219, DOI 10.1007/s00894-017-3378-9
31. Ayee, M.A.A.; LeMaster, E.; Shentu, T.P.; Singh, D.K.; Barbera, N.; Soni, D.; Tirupathi, Ch.; Subbaiah, P.V.; Berdyshev, E.; Bronova, I.; Cho, M.; Akpa, B.S.; Levitan, I. Molecular-Scale Biophysical Modulation of an Endothelial Membrane by Oxidized Phospholipids *Biophysical Journal*, **2017**, 112, 325–338, <https://doi.org/10.1016/j.bpj.2016.12.002>
32. Sadeghi, M.; Weik, T.R.; Noé, F. Particle-based membrane model for mesoscopic simulation of cellular dynamics *J. Chem. Phys.* **2018**, 148, 044901, <https://doi.org/10.1063/1.5009107>
33. S. Plimpton, Fast Parallel Algorithms for Short-Range Molecular Dynamics, *J. Comp. Phys.* **1995**, 117, 1-19.
34. Cheng, A.; Merz., K.M. Application of the Nose–Hoover Chain Algorithm to the Study of Protein Dynamics *J. Phys. Chem.* **1996**, 100, 1927-1937, DOI: 10.1021/jp951968y



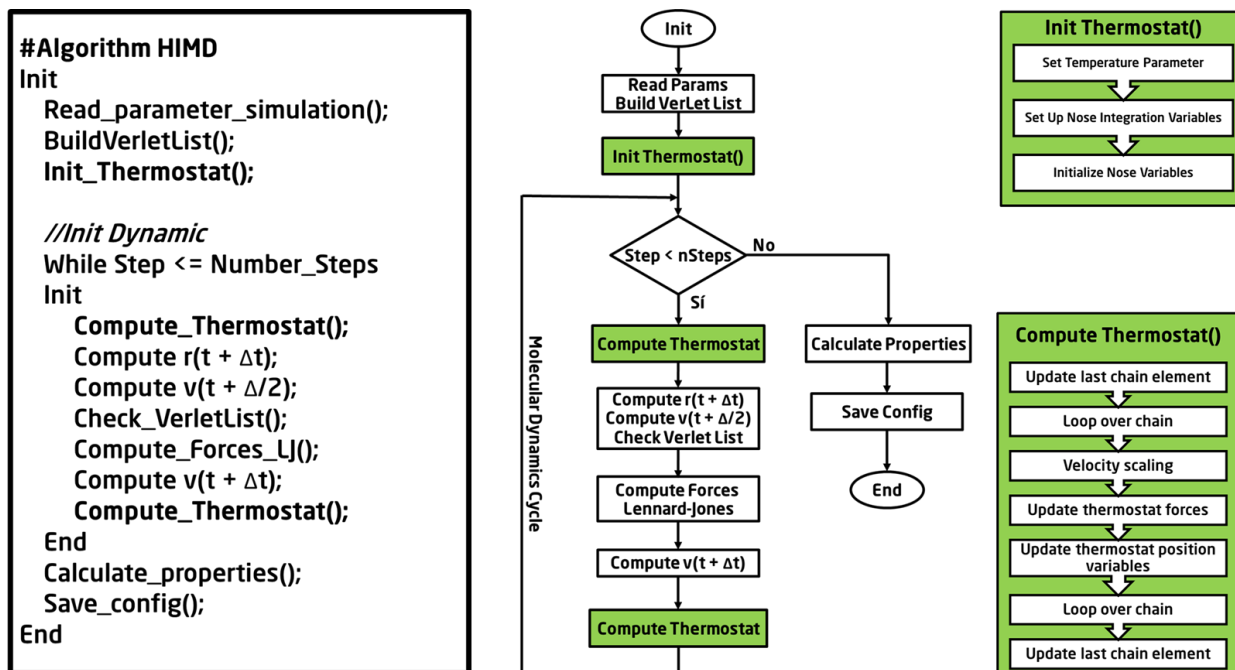
35. J. A. Anderson, C. D. Lorenz, and A. Travesset. General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of Computational Physics* **227**(10): 5342-5359, **2008**. 10.1016/j.jcp.2008.01.047
36. Hess B, Kutzner C, van der Spoel D, Lindahl E. GROMACS 4: Algorithms for highly efficient, load balanced, and scalable molecular simulation. *J. Chem. Theory. Comput.* **2008**, 4, 435-447.
37. Stanley, N.; De Fabritiis, G. High throughput molecular dynamics for drug discovery In Silico Pharmacology **2015**, 3, 3 DOI 10.1186/s40203-015-0007-0.
38. Erucar, I.; Keskin, S. High-Throughput Molecular Simulations of Metal Organic Frameworks for CO₂ Separation: Opportunities and Challenges *Front. Mater.* **2018**, 5, 4, doi: 10.3389/fmats.2018.00004
39. Hospital, A.; and Josep Ll Gelpi, J.Ll. High-throughput molecular dynamics simulations: toward a dynamic view of macromolecular structure *WIREs Comput Mol Sci* **2013**, doi: 10.1002/wcms.1142
40. Yamamoto, E.; Kalli, A.C.; Yasuoka, K.; Sansom, M.S.P. Interactions of Pleckstrin Homology Domains with Membranes: Adding Back the Bilayer via High-Throughput Molecular Dynamics *Structure* **2016**, 24, 1421–1431, <http://dx.doi.org/10.1016/j.str.2016.06.002>
41. Harvey, M.J.; De Fabritiis, G. High-throughput molecular dynamics: the powerful new tool for drug discovery *Drug Discovery Today* **2012**, 17, 1059-1062, <http://dx.doi.org/10.1016/j.drudis.2012.03.017>
42. Harvey, M.; Giupponi, G.; De Fabritiis, G. ACEMD: accelerated molecular dynamics simulations in the microseconds timescale. *J Chem Theory Comput* **2009**, 5, 1632–1639. DOI: 10.1021/ct9000685
43. M.E.; Alexandre, J.; López-Rendón, R.; Jochim, A.L.; Martyna, G.J. A Liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal–isobaric ensemble. *J. Phys. A: Math. Gen.* **2006**, 39, 5629–5651, doi:10.1088/0305-4470/39/19/S18.
44. Yu, T.-Q.; Alexandre, J.; López-Rendón, R.; Martyna, G.J.; Tuckerman, M.E. Measure-preserving integrators for molecular dynamics in the isothermal–isobaric ensemble derived from the Liouville operator. *Chem. Phys.* **2010**, 370, 294–305, doi: 10.1016/j.chemphys.2010.02.014.
45. Yu, T.-Q.; Tuckerman, M.E. Constrained molecular dynamics in the isothermal-isobaric ensemble and its adaptation for adiabatic free energy dynamics *Eur. Phys. J. Special Topics* **2011**, 200, 183–209, DOI: 10.1140/epjst/e2011-01524-x
46. Romero-Bastida, M.; López-Rendón, R. Anisotropic pressure molecular dynamics for atomic fluid systems. *J. Phys. A: Math. Theor.* **2007**, 40, 8585–8598, doi:10.1088/1751-8113/40/29/026.
47. Johnson, J.K.; Zollweg, J.A.; Gubbins K.E. The Lennard-Jones equation of state revisited *Mol. Phys.* **1993**, 78, 591-618, <https://doi.org/10.1080/00268979300100411>
48. Olinka Supercomputer. Available online: <http://www.olinka.uaemex-labs.org.mx/>.
49. Feig, M. Kinetics from Implicit Solvent Simulations of Biomolecules as a Function of Viscosity *J. Chem. Theory Comput.* **2007**, 3, 1734-1748, DOI: 10.1021/ct7000705.
50. Swope C. W.; Andersen C. H.; Berens H. P.; Wilson R. K. A computer simulation method for the calculation of equilibrium constants for the formation of clusters of molecules: Applications to small water clusters. *J. Chem. Phys.* **1982**, 76, 637-649. <https://doi.org/10.1063/1.442716>
51. NVIDIA Corp., CUDA C Programming Guide., NVIDIA Cor., Santa Clara California. Version 9.0. (**2018**), 2-25.



52. Humphrey, W., Dalke, A. and Schulten, K., "VMD - Visual Molecular Dynamics", *J. Molec. Graphics*, **1996**, vol. 14, pp. 33-38. <http://www.ks.uiuc.edu/Research/vmd/>
53. Turner PJ. XMGRACE, Version 5.1.19. Center for Coastal and Land-Margin Research, Oregon Graduate Institute of Science and Technology, Beaverton, OR; **2005**. <http://plasma-gate.weizmann.ac.il/Grace/>
54. D. Frenkel and B. Smit, *Understanding molecular simulation*, Academic, New York, 2002
55. <http://www.charmm.org>
56. James C. Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D. Skeel, Laxmikant Kale, and Klaus Schulten. Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry*, 26:1781-1802, **2005**
57. Hess, B., Kutzner, C., van der Spoel, D. and Lindahl, E. (**2008**) GROMACS 4: Algorithms for Highly Efficient, Load-Balanced, and Scalable Molecular Simulation, *J. Chem. Theory Comput.*, 4, 435-447.
58. J. A. Anderson, C. D. Lorenz, and A. Travesset. General purpose molecular dynamics simulations fully implemented on graphics processing units, *Journal of Computational Physics* 227(10): 5342-5359, May **2008**. 10.1016/j.jcp.2008.01.047
59. M. J. Harvey and G. De Fabritiis, An implementation of the smooth particle-mesh Ewald (PME) method on GPU hardware, *J. Chem. Theory Comput.*, 5, 2371–2377, **2009**
60. S. Plimpton, Fast Parallel Algorithms for Short-Range Molecular Dynamics, *J Comp Phys*, 117, 1-19 (**1995**)
61. Humphrey, W., Dalke, A. and Schulten, K., "VMD - Visual Molecular Dynamics", *J. Molec. Graphics*, **1996**, vol. 14, pp. 33-38

Anexos

Anexo 1. Pseudocódigo y diagrama de flujo del software HIMD



Anexo 2. Infraestructura utilizada

Este proyecto se realizó en las instalaciones de los Laboratorios de Bioingeniería Molecular a Multiescala (LBMM) y de Venómica Computacional, a cargo de los Drs. Roberto López Rendón y Felipe de Jesús Romero directores de este proyecto de tesis. Como la naturaleza de estudio de este proyecto es usar herramientas puramente numéricas, es necesario hacer uso de computadoras diseñadas exclusivamente para tales fines. El LBMM cuenta con la supercomputadora OLINKA. Además, se cuenta con computadoras personales para el análisis y discusión de los resultados que permitan dar seguimiento puntual a cada uno de los objetivos planteados. OLINKA está catalogada como una de las plataformas equipadas con la más alta tecnología en unidades de procesamiento gráficos conocidas como GPUS a nivel Latinoamérica. Adicionalmente haremos uso de software libre, como VMD [61] y XMGRACE [53] para realizar la visualización de gráficos y con ello llevar a cabo los análisis a los que tenga lugar.