

Implementación en GPU de un algoritmo eficiente de dinámica molecular con potenciales numéricos

Ignacio Eguinoa
Facultad de Informatica, UNLP

Resumen

En este trabajo se presenta una optimización del algoritmo de dinámica molecular sobre GPU para simulación de sistemas de partículas que interactúan mediante un potencial de Lennard-Jones.

Siguiendo las especificaciones del método de dinámica molecular se desarrolla, en primer lugar, una implementación propia de éste sobre GPU que evoluciona exclusivamente de acuerdo a un modelo de potencial de Lennard-Jones. Usando este desarrollo como referencia, se implementan adaptaciones para el cálculo del potencial y las fuerzas derivadas de éste basadas en el uso de valores tabulados, teniendo en cuenta tanto las propiedades de la función potencial como las características del hardware sobre el cual se trabaja.

Además de las mejoras en performance, se tiene en cuenta otro aspecto importante: la precisión numérica con la que se opera. Dado que la utilización de valores tabulados implica un error en el potencial y su fuerza resultante, es necesario evaluar la implicancia que tendrá esto sobre la correctitud de la simulación.

Las implementaciones actuales del método sobre arquitecturas GPU proveen una gran mejora en la performance con respecto a las versiones iniciales implementadas sobre CPU, lo cual es producto de las propiedades intrínsecamente paralelas del algoritmo y la arquitectura altamente paralelizada del hardware. Por su parte, las adaptaciones realizadas en este trabajo presentan una optimización extra manteniendo la calidad numérica de los resultados.

Dado que el potencial de Lennard-Jones representa un aporte importante para modelar las interacciones en distintos campos de fuerzas, las mejoras obtenidas pueden ser incorporadas fácilmente a distintas versiones existentes del método de dinámica molecular.

Por último, los resultados obtenidos alientan a evaluar este tipo de aproximaciones para realizar cálculos con formas funcionales similares que hayan sido implementados sobre GPU. De esta forma, se intenta explotar todas las propiedades que ofrece la arquitectura.

Índice general

1. Introducción	3
1.1. Objetivos	3
1.2. El metodo de dinamica molecular	3
1.2.1. Fundamentos	3
1.2.2. Esquema general del método	3
1.2.3. La función potencial	4
1.2.3.1. Potencial de Lennard-Jones	5
1.2.4. Condiciones de periodicidad	5
1.2.5. Estado del arte	5
2. La arquitectura GPU	6
3. Algoritmos	7
3.1. Algoritmo base sobre GPU (o Esquema general del algoritmo)	7
3.2. Calculo usando tablas de valores del potencial	8
3.3. Calculo usando tablas de valores de la derivada	8
4. Implementaciones	9
4.1. Esquema general de la implementación	9
4.2. Implementacion sobre CPU	9
4.3. Implementación usando tabla de valores potenciales	9
4.4. Implementación usando tabla de derivadas	9
5. Resultados	10
5.1. Sistemas de partículas utilizados	10
5.2. Performance	10
5.2.1. Evaluación de las implementaciones	10
5.2.2. Efectos del tamaño de bloque	10
5.3. Calidad numérica	10
6. Conclusiones	11

Capítulo 1

Introducción

1.1. Objetivos

El principal objetivo de la tesis es obtener una implementación del método de dinámica molecular basada en GPU, que siga un modelo de interacción de Lennard-Jones (potencial de no-unión), y que resuelva el cálculo de las fuerzas utilizando valores tabulados en el sistema de memoria provisto por la arquitectura.

La forma funcional que describe el potencial de Lennard-Jones permitiría obtener una buena aproximación del valor utilizando una tabla de resultados precalculados. Además, el sistema de memorias de la arquitectura, que ha evolucionado considerablemente desde su creación de esta, permite recuperar de forma eficiente el valor asociado en un esquema de tablas. De esta forma, se espera que la modificación resulte en una mejora de la performance manteniendo la correctitud en los resultados de la simulación.

1.2. El método de dinámica molecular

1.2.1. Fundamentos

Los sistemas químicos de interés como las proteínas suelen ser complejos de estudiar debido a su gran tamaño. La superficie de energía libre es la función que describe la energía del sistema, y a partir de la cual se podrían derivar todas las propiedades termodinámicas y cinéticas de interés. Esta, sin embargo, es una función $3N$ dimensional (siendo N el número de partículas del sistema), lo que hace imposible derivar analíticamente las propiedades debido al altísimo costo computacional asociado. Otra forma de analizar estos sistemas es mediante una aproximación numérica, a través de simulaciones computacionales.

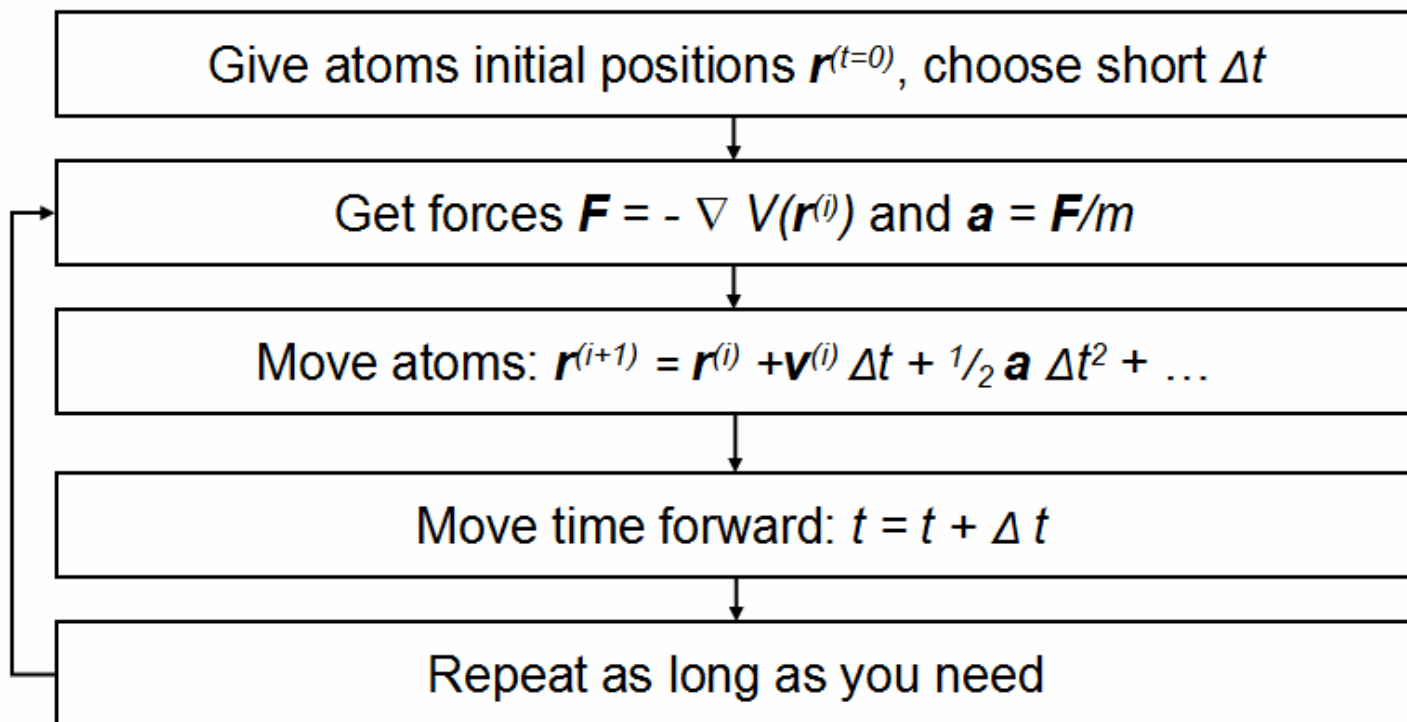
La simulación computacional de biomoléculas involucra la exploración de su superficie de energía libre, la cual, debido a la complejidad de estos sistemas, es altamente accidentada, contiene una gran cantidad de mínimos locales y de barreras de energéticas. De esta forma, si los parámetros de la simulación están correctamente definidos, el conjunto de conformaciones adoptadas durante la ejecución será representativo del ensamble real de conformaciones posibles del sistema, lo que nos permite estimar las propiedades de interés. Los métodos de simulación molecular nos permiten, entonces, obtener una serie de configuraciones representativas del sistema, de modo que las propiedades termodinámicas extraídas del mismo se correspondan de manera precisa con los valores reales. Una de las formas de obtener estas configuraciones es mediante el método de dinámica molecular. Este método implica simular la progresión temporal real del sistema, obteniendo distintas conformaciones a medida que avanza el tiempo de simulación.

La técnica de simulación de Dinámica Molecular se basa en resolver las ecuaciones de movimiento de Newton para cada átomo del sistema; así, en cada "paso" de la simulación se calcula la energía potencial entre las partículas y de esta se derivan las fuerzas que actúan sobre cada átomo. Las ecuaciones de Newton relacionan las fuerzas resultantes del potencial con la aceleración que tendrá cada partícula y, por lo tanto, el cambio en la velocidad y en la posición con el tiempo. Dado que el potencial es una función continua dependiente de la posición, los cambios en las velocidades y posiciones resultan de una integración a lo largo de la trayectoria de la partícula.

1.2.2. Esquema general del método

En la práctica, la integración de las Ecuaciones de Newton se calcula computacionalmente mediante el llamado Algoritmo de Verlet [14], el cual resuelve la integración en una ecuación discreta donde las posiciones están separadas por intervalos de tiempo ("dt"). Usando las fuerzas resultantes, junto con las posiciones y velocidades de las partículas correspondientes a la iteración, el algoritmo de Verlet calcula las nuevas posiciones y velocidades en un intervalo de tiempo posterior. De este modo se genera una trayectoria, determinada por las posiciones de las partículas en cada paso de la simulación. Esta trayectoria describe cómo cambia la conformación espacial del sistema a lo largo del tiempo. La elección del "dt" es una situación de compromiso, ya que un valor muy chico, si bien representa la propagación del movimiento de forma más precisa (más cercana al valor de la integración), requiere una mayor cantidad de cálculos para un mismo tiempo total de simulación.

El esquema general del método es el siguiente:



El resultado de una simulación de dinámica molecular es principalmente una trayectoria representada por la posición de cada una de las partículas a lo largo de un cierto tiempo. No se va a detallar acerca de los métodos para el procesamiento de este resultado pero es relevante decir que la longitud de trayectoria que podamos obtener en un tiempo de cómputo razonable es lo que limita el tipo de proceso químico que podremos estudiar. Para poder estudiar un cierto proceso de interés es necesario que la simulación alcance la escala de tiempo en la cual este ocurre. Un mayor tiempo de simulación equivale a realizar más pasos y por lo tanto más cálculos totales, aumentando así el tiempo de cómputo requerido.

Por su parte, simular sistemas más grandes (con mayor cantidad de partículas) implica una mayor cantidad de cálculos en cada paso y por lo tanto un mayor costo computacional. Además, la función que describe las propiedades del sistema tiene por naturaleza una mayor cantidad de variables, debido a esto es necesario correr la simulación durante más tiempo para poder llegar a un conjunto de conformaciones que sea representativo del ensamble real.

1.2.3. La función potencial

En la utilización del método de dinámica molecular, el potencial (V) que permite derivar las fuerzas entre partículas se obtiene modelando al sistema molecular mediante la mecánica clásica (método de mecánica molecular, MM). Usando un método de MM [12], se ignoran los electrones y la naturaleza cuántica de estos. La energía potencial del sistema, entonces, depende exclusivamente de las posiciones de los núcleos atómicos. Se modela cada molécula como un conjunto de sitios -que representan los átomos que la componen- y resortes -que representan los enlaces químicos entre estos- junto con un potencial parametrizado ad hoc. Este potencial es una función matemática que depende exclusivamente de las posiciones de los átomos, intentando ajustarse a las interacciones entre los componentes del sistema real. Este tipo de representación simplificada del sistema permite reducir la complejidad de los cálculos necesarios para la simulación, y por lo tanto el costo computacional asociado, pero limita el tipo de procesos que se pueden estudiar. Por ej. no se pueden estudiar reacciones químicas que impliquen ruptura o formación de enlaces ya que estos no son considerados con suficiente detalle.

Las interacciones representadas en el método de MM se pueden agrupar en dos tipos:

De unión: Describen las interacciones entre dos átomos unidos entre sí directamente o hasta 2 enlaces de distancia. Consisten en aquellos términos del potencial cuya energía se ve afectada por los estiramientos de los enlaces, las flexiones de los ángulos entre 2 átomos, y la rotación de dos átomos adyacentes sobre un eje (ángulos dihedros). Los estiramientos y las flexiones angulares son modeladas mediante un oscilador armónico, mientras que las rotaciones de los enlaces en el plano son modeladas mediante una función trigonométrica.

De no unión: describen la interacción entre átomos ubicados a más de 3 enlaces de distancia de la misma molécula, o bien entre átomos de moléculas distintas, y consisten en un término para la contribución electrostática computada con la Ley de Coulomb, y un término para la contribución de Van Der Waals modelada por un potencial de Leenard-Jones 12-6.

El esquema general de la función potencial suele tomar una forma similar a la que sigue:

$$V(\mathbf{r}) = \underbrace{\sum K_b(b-b_0)^2 + \sum K_\theta(\theta-\theta_0)^2 + \sum (V_n/2)(1+\cos[n\phi-\delta])}_{\text{interacciones de union}} + \underbrace{\sum (A_{ij}/r_{ij}^{12}) - (B_{ij}/r_{ij}^6) + (q_i q_j / r_{ij})}_{\text{interacciones de no union}}$$

En azul se representan los términos matemáticos que modelan las interacciones de unión (1. Término de estiramiento de enlaces; 2. Término de flexión angular; 3. Término de rotación de ángulos diedros), y en rojo las interacciones de no unión (Potencial de Leenard-Jones 12-6 e interacción Coulómbica).

Los parámetros que se observan en esta función potencial (las constantes K_b , K_θ y V_n , los valores de equilibrio b_0 , θ_0 y δ , las constantes A y B correspondientes al potencial Lenard-Jones, y las cargas q) deben ser ajustados especialmente para cada molécula individual que se desee incluir en la simulación. Estos parámetros se suelen obtener tanto de datos experimentales, como así también de rigurosos cálculos cuánticos, y en ocasiones emplean correcciones empíricas. Al conjunto o "set" de parámetros derivados para utilizarse de forma conjunta en una simulación se lo conoce como Campo de Fuerzas.

El modelo de interaccion(campo de fuerzas) es la principal diferencia entre los distintos paquetes de software para simulaciones de dinamica molecular. Este campo de fuerzas esta parametrizado especificamente para el tipo de sistema que se intenta simular, de forma que se ajuste lo mejor posible a la realidad. De esta forma, existen versiones especificas para simular sistemas biologicos (proteinas, acidos nucleicos), otros que intentan abarcar a sistemas de quimica organica en general, etc.

1.2.3.1. Potencial de Lennard-Jones

El calculo del potencial de no union es una parte importante en la implementacion del algoritmo ya que implica la mayor parte del costo computacional asociado a la simulacion. Esto se debe a que se debe calcular entre todos los elementos del sistema. En particular, es importante el termino correspondiente al potencial de Lennard-Jones porque este existe siempre, independientemente de la carga neta en las particulas que interaccionan.

La ecuacion que representa este potencial es:

La forma funcional de esta interaccion resulta:

A partir de los valores del grafico, resulta razonable pensar que, si se desprecia el aporte de este potencial para interacciones entre particulas a una distancia mayor que cierto valor de corte, se podra obtener una buena aproximacion del comportamiento a la vez que se reduce considerablemente la cantidad de calculos necesarios en cada paso de la simulacion.

Como se mencionó previamente, el potencial de L-J es uno de los componentes que se encuentra en la mayoría los campos de fuerza existentes, por lo que optimizar el calculo de este tendria una gran ventaja. Ademas, al ser una interaccion de no union, existe, al menos debilmente, entre todos los pares de particulas del sistema, lo que implica una gran cantidad de calculos por paso. El calculo de potenciales de no union representa asi la mayor parte del costo computacional asociado a la simulacion, existen algunas optimizaciones para el calculo del aporte electroestatico pero no para el componente de L-J de manera que este genera un cuello de botella en las mejoras del tiempo de ejecucion.

1.2.4. Condiciones de periodicidad

Real systems are almost infinite in size

In vacuo simulation are rarely a good idea

Periodic boundary conditions: -Simulation boxes and nearest image conventions

-Infinite electrostatics: Ewald summation

-Problem: physical properties depend on box size

1.2.5. Estado del arte

Existen distintas aproximaciones para evaluar las interacciones que apuntan a optimizar el costo computacional asociado al calculo. Algunas implementaciones permiten definir un potencial de solvente implicito, (referencia : Born solvation model [V. Tsui D.A. Case, Biopolymers (Nucl. Acid. Sci.) 56, 275-291 (2001)].) Otra optimizacion posible es para el calculo del potencial de interaccion electrostatico : PME. Este es importante ya que permite simplificar el calculo del aporte electroestatico el cual, al ser un potencial de no-union, implica una gran cantidad de calculos.

Las capacidades de cálculo actuales permiten realizar simulaciones de dinámica molecular en sistemas de hasta XXXXXXXXXX átomos aproximadamente sobre un intervalo de tiempo del orden de YYYYYYY micro/pico/*****segundos).

Capítulo 2

La arquitectura GPU

Capítulo 3

Algoritmos

A partir de la descripción del método presentada en el capítulo 1 se puede llegar a una implementación secuencial en forma casi directa. Este tipo de versiones son las que se utilizaron inicialmente para realizar las simulaciones. La utilidad que ha mostrado esta técnica desde sus inicios y la posibilidad que ofrece para nuevas aplicaciones hizo que se desarrollen diversas implementaciones para aprovechar los nuevos modelos de cómputo disponibles, los cuales han avanzado considerablemente teniendo en cuenta que la técnica fue creada hace ya un largo tiempo. Entre los nuevos algoritmos, una gran cantidad se ha enfocado en explotar las características intrínsecamente paralelas del método. En los últimos años, con avance de las arquitecturas de GPU y su utilización para cómputo de propósito general, estos algoritmos han sido portados para adaptarse a las características propias de GPGPU. De las versiones mencionadas en el capítulo 1, gran parte de ellas han sido portadas para poder ejecutarse sobre estas arquitecturas. (referencias `amberGPU??` `gromacs gpu??`).

En el capítulo anterior se mostraron las ventajas que brinda el método de GPGPU para aplicaciones altamente paralelizables. De la misma forma, la implementación del método de dinámica molecular sobre esta arquitectura resulta en un aumento muy importante en la performance en forma casi directa. Además de las mejoras en la performance, el bajo costo del hardware en relación a su poder de cómputo que provee la arquitectura hace que estas implementaciones sean un estándar de uso en el área.

De esta forma, hoy en día se puede encontrar una gran variedad de software para simulaciones de dinámica molecular implementado sobre GPU. A pesar de la gran cantidad de modificaciones que se han agregado y la evolución en el hardware existente, se sigue dedicando un gran esfuerzo para continuar optimizando el algoritmo con el fin de poder simular nuevos procesos y sistemas de interés que requieren una mayor cantidad de cálculos. Si se analizan las implementaciones que se utilizan actualmente, se puede ver que la mayor parte del costo computacional del método se mantiene asociado al cálculo de las interacciones de no-union (entre partículas que no forman enlaces químicos entre sí), el cual está representado principalmente por el potencial de L-J. Este cálculo tiene una complejidad de orden N^2 sobre el número de partículas del sistema.... En el capítulo 1 se describió una aproximación muy utilizada para intentar reducir la cantidad de cálculos necesarios, la cual consiste en despreciar las interacciones entre partículas que se encuentren a mayor distancia que cierto valor de cutoff. Esta aproximación está basada en las propiedades de la función que describe el potencial de L-J. Incluso ajustando el valor de cutoff a valores suficientemente chicos, el costo que implica este paso representa la mayor parte de los requerimientos..... Por esta razón, una gran parte de los esfuerzos se centran en optimizar esta etapa.

Como se puede apreciar a partir de la descripción de este potencial en el capítulo 1, la forma funcional que lo describe se caracteriza por tener un rango de distancias donde el resultado crece (o decrece) exponencialmente, y un punto a partir del cual converge y el valor asociado a la interacción se hace despreciable. Basándose en estas características es razonable pensar que, utilizando una tabla conteniendo los resultados para un rango de valores definido y, aun sin tener ésta un tamaño excesivo, se podría obtener una buena aproximación del resultado para cualquier distancia.

Para implementar este mecanismo hay varias propiedades del sistema de memoria que se puede usar y que se deben analizar.....

Se debe tener en cuenta que, aún cuando sea posible realizar una buena aproximación utilizando tablas, el algoritmo de dinámica molecular implica utilizar este valor para obtener las fuerzas y movimientos resultantes de la interacción, lo cual afecta la posición (y por lo tanto el valor del potencial asociado) en la próxima iteración. De esta forma, los errores resultantes de la aproximación por utilizar tablas de valores serán propagados a lo largo de una simulación que involucra miles de pasos.....

Así, en este trabajo se estudiarán distintas formas de utilizar el sistema de memoria de la GPU para realizar el cálculo del potencial de Lennard Jones, analizando su utilización en el contexto del método de dinámica molecular. Dado que el cálculo de este potencial representa la mayor parte del costo asociado a la simulación, se espera obtener una mejora considerable en el tiempo de ejecución con respecto al algoritmo estándar sobre GPU, el cual resuelve la ecuación de L-J para cada interacción a evaluar.

3.1. Algoritmo base sobre GPU (o Esquema general del algoritmo)

El algoritmo inicial sigue la especificación que se detalla en el capítulo 1.... Dado que el aporte al campo de fuerzas que nos interesa es el correspondiente al potencial de L-J (potencial de no unión), este es el único aporte que se va a considerar. De esta forma, el sistema evoluciona en el tiempo a partir de las velocidades iniciales de las partículas y las fuerzas que se

derivan de este potencial, unicamente. Si bien este modelo (considerando solo la interaccion de L-J) no es de utilidad para simular sistemas biologicos de interes (donde existen diversos enlaces entre los elementos), en este caso

Si bien el codigo aqui implementado ya está implementado como parte de distintos paquetes de software de dinamica molecular, el objetivo de realizar un algoritmo independiente, donde la evolucion temporal ocurra unicamente en base al potencial de L-J

Dado que el trabajo se centra puntualmente en modificar la forma en que se calcula el potencial de L-J, el primer paso consiste en implementar un algoritmo de dinamica molecular que utilice exclusivamente un

Dentro de las multiples versiones del algoritmo que poseen implementaciones sobre GPU, gran parte implementa optimizaciones orientadas a mejorar el rendimiento basadas en las propiedades del sistema a simular . Por ej. variaciones en la forma de representar las interacciones con el solvente, Dado que en este trabajo apuntamos a optimizar el calculo del potencial de interaccion de L-J, en primer lugar se realiza una implementacion propia del algoritmo de Dinamica molecular sobre GPU que utiliza exclusivamente un potencial de interaccion de L-J.

El objetivo de esto es, en primer lugar, facilitar la implementacion de esta modificacion/optimizacion en el algoritmo, y en segundo lugar probar las propiedades de esta variante de forma independiente a cualquier otra implementacion existente sobre GPU. Utilizando una implementacion propia se puede evaluar, de forma aislada, las ventajas y desventajas del uso de una tabla numerica. De esta forma se puede verificar facilmente que las variaciones en el tiempo y en la precision de los resultados son producto exclusivo de las modificaciones implementadas en cada paso.

Idealmente, las ventajas resultantes de este trabajo se sumaran a las caracteristicas de cualquier implementacion existente.

3.2. Calculo usando tablas de valores del potencial

3.3. Calculo usando tablas de valores de la derivada

Capítulo 4

Implementaciones

- 4.1. Esquema general de la implementación
- 4.2. Implementacion sobre CPU
- 4.3. Implementación usando tabla de valores potenciales
- 4.4. Implementación usando tabla de derivadas

Capítulo 5

Resultados

5.1. Sistemas de partículas utilizados

5.2. Performance

5.2.1. Evaluación de las implementaciones

5.2.2. Efectos del tamaño de bloque

5.3. Calidad numérica

Capítulo 6

Conclusiones