# Global Optimization on Complex Systems

Universidad de La Laguna

Universidad de La Laguna

Roberto Díaz Pérez
Tutor: Javier Hernández Rojas

03/03/2017

# Índice general

# Capítulo 1

# Abstract

## 1.1. Abstract

Nuestro objetivo en este trabajo es encontrar el mínimo global del potencial de Lennard-Jones (LJ) y del potencial de Lennard-Jones mejorado (ILJ) para un clúster de N partículas. Obtener el mínimo global es importante porque la estructura natural del sistema suele ser aquella que minimiza la energía potencial.

El potencial LJ ha sido utilizado para modelar el comportamiento de gases nobles, átomos y moléculas neutrales. El potencial ILJ es una corrección del LJ que corrige el comportamiento tanto a rangos cortos como a rangos largos.

Tres algoritmos de optimización global son comparados en este texto:

- Simulated annealing (SA), o templado simulado, está inspirado en el templado de un vidrio. Este método realiza una exploración de la superficie de energía potencial (PES), mediante el desplazamiento aleatorio de las coordenadas de cada uno de los átomos. Este desplazamiento es aceptado de acuerdo a la probabilidad de Boltzmann, cuya temperatura va disminuyendo en cada iteración, de forma similar al templado del vidrio.

- Genetic algorithms (GA), o algoritmos genéticos, es un método de optimización global basado en los principios evolutivos que contiene operadores como reproducción, mutación o selección natural. Este método es aplicable a cualquier problema en el que las variables se puedan poner como una cadena o cromosoma, siendo cada variable un alelo. La idoneidad de un cromosoma es una propiedad que nos indica lo bueno que es dicho cromosoma. Al ser nuestro problema de optimización, el cromosoma con más idoneidad será aquel con menor energía. En cada iteración sobreviven un número de hijos que son seleccionados de acuerdo su idoneidad. Este algoritmo explora varias zonas de la función simultáneamente, como un árbol familiar.

- Basin-Hopping (BH), o salto de pozos, es el método de optimización utilizado en este trabajo, SA y GA son tratados de forma bibliográfica. En este algoritmo la PES es transformada al mínimo local más cercano. Esta PES transformada es explorada de acuerdo a la probabilidad de Boltzmann, con una temperatura constante.

Para obtener los resultados se programó el método Basin-Hopping y se calcularon los potenciales LJ y ILJ junto con sus gradientes, necesarios para el algoritmo de

minimización local utilizado. Además fue necesario optimizar la implementación para reducir el tiempo de computación, lo que incluyó cambios en el método de minimización local.

Se obtuvieron los resultados tanto para LJ como para ILJ hasta N = 49. Por un lado, los resultados del potencial LJ fueron comprobados con los obtenidos en otros trabajos [10], verificando su validez. Por otro lado, este es el primer trabajo en obtener los mínimos globales del potencial ILJ. Además, se realiza un análisis de las estructuras geométricas encontradas, basadas en el icosaedro de Mackay excepto por el caso especial de N = 38, comprobando que las estructuras no cambian entre un potencial y otro.

# Capítulo 2

# Global Optimization

## 2.1. Introduction

Optimization has always been quite important in mathematics, economics, physics and many other fields. From Fermat's theorem [1] to Lagrange's multipliers [4] mathematicians have developed multiple ways to find the minimum of a function. Optimization has been used for solving multiple problems, from really simple ones, like finding the best shape for a milk bottle (optimizing volume, cost and ease of transportation), to more complex ones, for example, the design of microprocessor circuitry.

In physics and chemistry, the discovery of the lowest energy structure for a given system is often very useful, as it is likely that the natural structure of the system is the one that minimizes the potential energy. Our final goal is to obtain the global minimum of the potential energy surface (PES) of a cluster of N atoms affected by both the Lennard-Jones potential and the improved Lennard-Jones potential, using, and understanding, global optimization methods such as Basin-Hopping.

In essence, the optimization problem can be represented as:

Given a function f: $A \to \mathbb{R}$ from some set A to the real numbers, we seek $x_0$ in A such that $f(x_0) \leq f(x) \forall x \in A$.

Our problem is not a simple one, given that, for the Lennard-Jones potential, $A \in \mathbb{R}^{3N}$, which means that we are going to be probing a 3N dimensional function searching for the global minimum, with N being any number from 2 to several hundreds, and with an exponential number of local minima as N grows, for example 1328 for N=13 [11]. It should be clear that the use of local minimzation algorithms does not suffice, requiring the use of global optimization algorithms.

The traditional approach to finding the global minima is to perform several transformations starting from a random initial configuration and going to the nearest local minima using an algorithm. This method has been deemed ineffective as N increases, because the number of local minima increases exponentially with N. Thus, mathematicians developed other optimization techniques, such as Simulated Annealing [9], Genetic Algorithms [2] and Basin-Hopping [10].

## 2.2. Lennard-Jones Potential

The potential we are going to be working with, the Lennard-Jones (LJ) potential, is defined as:

$$V_{LJ} = 4\epsilon \sum_{j=1}^{N} \sum_{i<j} \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^{6} \right] \tag{2.1}$$

The value of the constants changes for each system, a table of values can be found in [7].

To simplify, from now on, unless otherwise is stated, we will be using the reduced Lennard-Jones units, assigning the values $\epsilon = 1$ and $\sigma = 1$, giving:

$$V^*_{LJ} = 4 \sum_{j=1}^{N} \sum_{i<j} \left[ \left( \frac{1}{r_{ij}} \right)^{12} - \left( \frac{1}{r_{ij}} \right)^{6} \right] \tag{2.2}$$

As we can see in figure 2.1, this potential is repulsive at short distances and attractive at long distances, having a minimum at $V = -\epsilon$. Trivially, we can obtain a lower bound for the total energy, $V = -\frac{N(N-1)\epsilon}{2}$, assuming each pair is at their equilibrium separation. For N=2,3,4 this lower bound can be achieved, but from N=5 onward it's impossible to place each atom at the minimum of the potential energy of every other one.

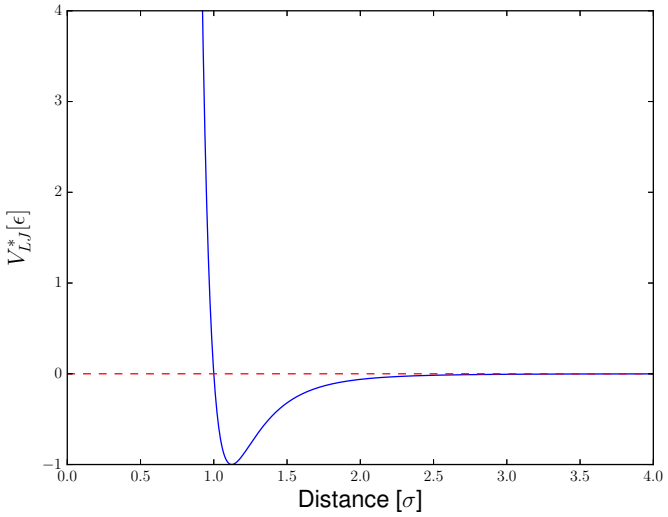



Figura 2.1: Lennard-Jones potential for N = 2.

Due to its computational simplicity, the Lennard-Jones potential is often used to describe the properties of gases. It's specially accurate for noble gases and neutral atoms and simple molecules.

### 2.2.1. Improved Lennard-Jones potential

The improved Lennard-Jones potential [7] is a potential introduced to fix some inaccuracies of the Lennard-Jones potential both at short and long range and better representation of non neutral atoms and molecules.

This potential is defined as follows:

$$V(r) = \epsilon \left( \frac{m}{n(r) - m} \left( \frac{r_m}{r} \right)^{n(r)} - \frac{n(r)}{n(r) - m} \left( \frac{r_m}{r} \right)^{m} \right) \tag{2.3}$$

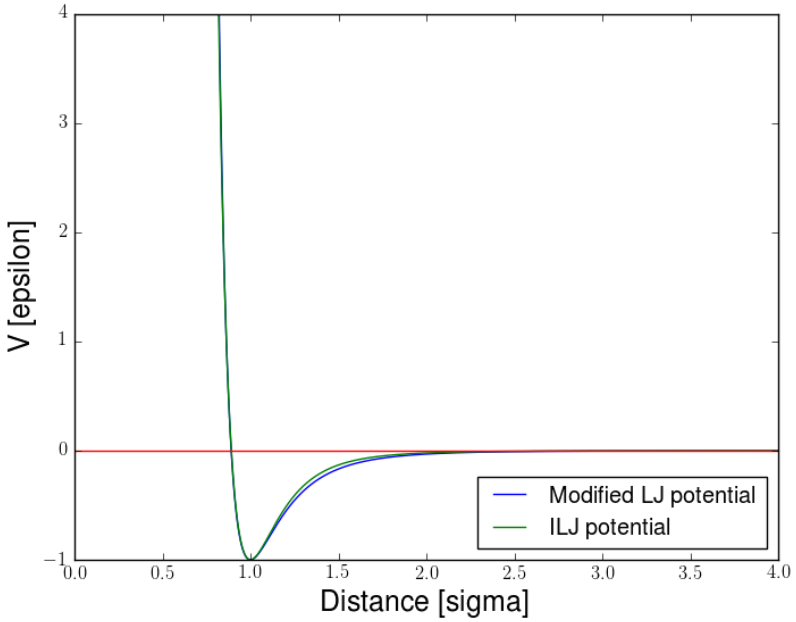



Figura 2.2: Comparison of ILJ vs LJ potentials for N=2.

$\epsilon$ and $r_m$ are, respectively, the depth of the potential well and its location. The exponent m assumes the value m=6 for neutral-neutral pairs, m=4 for ion-neutral pairs and m=1 for ion-ion pairs.

$n(r)$ is defined as follows:

$$n(r) = \beta + 4 \left( \frac{r}{r_m} \right)^2 \tag{2.4}$$

$\beta$ represents the hardness of the two interacting partners and, in our case, is fixed to 9 in order to obtain good results while studying noble gases.

In our case, we have a neutral-neutral system so m is fixed to 6, and using Lennard-Jones units, $r_m$ and $\epsilon$ are assigned the value of 1. The normalized potential becomes:

$$V^*_{ILJ}(r) = \frac{6r^{-9-4r^2} - (9 + 4r^2)r^{-6}}{3 + 4r^2} \tag{2.5}$$

In order to compare this potential versus the LJ potential, some tweaking must be done. The LJ potential for N=2 has its minimum at $r = 2^{1/6}$, but the ILJ potential has its minimum at r=1. We adjust the LJ potential to move its minimum to 1, whilst maintaining the energy value:

$$V^{adjusted}_{LJ}(r) = r^{-12} - 2r^{-6} \tag{2.6}$$

This is done by instead of using changing the value of $\sigma$ from 1 to $\frac{1}{2}^{\frac{1}{6}}$. This does not affect the results energy wise as the energy is given in $\epsilon$ units, which remains unchanged.

It is correct to compare the results obtained by the LJ potential and the ILJ potential, even if the position of the minimum differs, because the potential energy and the structure of the molecule will remain the same, just stretched bigger.

We compare the LJ and the ILJ potentials in Figure 2.2.

## 2.3. Simulated Annealing

Simulated Annealing (SA) is a global optimization method that uses a combination of quenches and annealings at various cooling rates to obtain the global minimum. It has been used at obtaining the configuration of minimum energy of atomic clusters held together by Lennard-Jones interactions (2.2) by L. T. Wille [9].

SA is based on the analogy of a perfect crystal obtained from the melt by a slow cooling process called annealing, following this analogy, a steep descent into a local minima would be a quench.

Using the montecarlo method, SA is based on a biased random walk on the potential energy surface using the Metropolis algorithm (A.1). A move is proposed by perturbing the coordinates from the current energy, $V_{old}$, to a new one, $V_{new}$. This move is accepted according to the Boltzmann probability (2.7).

$$\rho = e^{-\frac{V_{new}-V_{old}}{k_B T}} \tag{2.7}$$

With $k_B$ being the Boltzmann constant and T the temperature.

The SA is usually started at a high temperature, so the cluster is a liquid state, then the temperature is decreased logarithmically ($T \to aT, 0 < a < 1$) until the solid state is reached, and once we are deep in the solid state, a quench may be done, as we are likely in a basin.

A disadvantage of this method is the fact that the simulation is very likely to be trapped in a zone of the PES not corresponding to the global minima as shown in [9], with a slow cooling rate this phenomena can be reduced, but it is always present.

## 2.4. Genetic Algorithms

Genetic Algorithm (GA) [2] is a global optimization method based on the principles of evolution. It uses operators, such as mating, mutation and natural selection, found in natural election. GA can be applied to any problem where the variables to be optimized (genes) must be encoded to form a string (chromosome). As in biology, the value of each variable are known as alleles. This relation is explained schematically in Fig.2.3.

The initial set of individuals who are to be evolved by GA are usually chosen randomly, though by prior knowledge it might be useful to use a custom initial population.

The fitness of a string is a measure of its quality, in a minimization problem the best string would be the one which, when passed to the function being optimized returns the lowest value. If the upper and lower limits of the function are unknown,

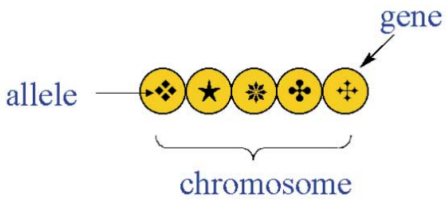



Figura 2.3: Schematic representation of a chromosome in GA optimization. Image extracted from [2].

a dynamic fitness scaling can be adopted, relating the fitness of each member to the best and worst members of the population.

To select a member for subsequent crossover there are two methods: roulette wheel selection and tournament selection. In tournament selection a group of members are chosen at random and the two with the best fitness are chosen. In roulette wheel selection a random member is chosen according to a probability associated with its fitness value. This is easy to visualize as a weighted roulette, the size of the slots increasing with the fitness, as shown in Figure 2.4.

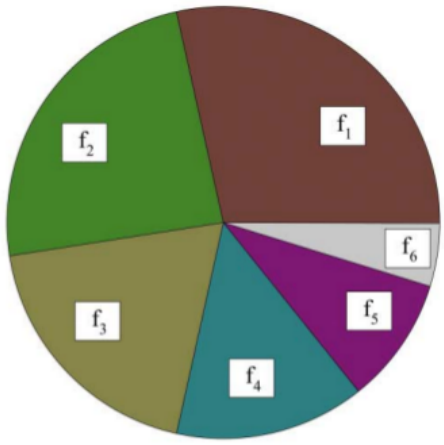



Figura 2.4: Roulette selection. Image extracted from [2].

Then, the crossover is done, mixing the genes of the parents. To introduce new material, helping to avoid stagnation, the mutation is done by randomly changing one or more chosen genes in an individual.

Natural selection, whilst not so natural in our case, is done by choosing which of the childs survive according to their fitness.

GA investigates multiple regions at the same time expanding like a family tree, if a pattern emerges that gives individuals with great fitness, the algorithm recognizes this and propagates it in the population.

## 2.5. Resumen

El problema a resolver en este trabajo es la obtención del mínimo global del potencial de Lennard-Jones (LJ) y del potencial Lennard-Jones mejorado (ILJ), para N partículas. Los métodos tradicionales de minimización no son adecuados para resolver este problema debido al gran número de mínimos locales presentes,

por ejemplo, 1328 mínimos para N=13 [11]. Este problema se hace más evidente según aumentamos N, ya que el número de mínimos aumenta exponencialmente. Necesitaremos usar métodos de optimización global.

Trataremos 3 métodos de optimización global: Simulated annealing y Genetic Algorithms, que son tratados de forma bibliográfica, y Basin-Hopping que es el utilizado de forma práctica para obtener los mínimos globales.

El método simulated annealing, o templado simulado, emula el proceso de templado de un vidrio. Partiendo de una posición inicial aleatoria, se aplican transformaciones en la posición, y, mediante la probabilidad de Boltzmann (2.7), se decide si se acepta o no este cambio de posición. La temperatura se va disminuyendo de forma logarítmica, simulando el enfriamiento del vidrio. Una vez que el sistema se quede atrapado en un mínimimo local, o pozo, se disminuye la temperatura de forma drástica.

El método de algoritmos genéticos, esta basado en los principios evolutivos. Usa operadores tales como apareamiento, mutación, selección natural y cromosomas. La mayor diferencia con otros métodos de optimización es que este algoritmo explora la función muchas regiones, expandiendose como un arbol familiar.

El método de Basin-Hopping o salto entre pozos, será explicado en el siguiente capítulo.

# Capítulo 3

# Basin-Hopping

## 3.1. Introduction

Basin-hopping is an optimization technique in which the potential energy surface (PES) is transformed associating each point to the local minimum, simplifying the problem considerably by getting rid of transition states entirely.

## 3.2. Method

We consider the transformed energy defined as:

$$\tilde{E}(\vec{X}) = min(E(\vec{X})) \tag{3.1}$$

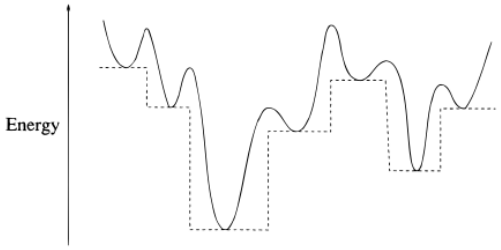



Figura 3.1: Schematic diagram of the transformation of the PES in the Basin-Hopping algorithm. Image extracted from [10].

$\vec{X}$ is the 3N-dimensional vector of nuclear coordinates and min signifies that a local minimization is performed starting at $E(\vec{X})$. In the transformed energy, a number of interconnected staircases are formed, creating basins in the joint points, giving name to the algorithm.

In this work, the local minimization is done using the Limited-Memory Broyden-Fletcher-Goldfarb-Shanno method (L-BFGS) (A.3) as it is a really efficient algorithm that was already implemented in Scipy, a library for python [8].

Then the transformed PES is explored displacing randomly the coordinates, following:

$$x_i^{new} = x_i^{old} + s \times 2(\gamma - 0{,}5) \tag{3.2}$$

With $\gamma$ being a random number, chosen from an uniform distribution between 0 and 1, and $s$ being the maximum step size, which is adjusted dynamically every few steps to get an acceptance rate of 0.5.

The condition to accept the change is defined as the Boltzmann probability:

$$\rho = e^{-\Delta V/kT} > random[0,1] \tag{3.3}$$

This probability increases as $\Delta V = (V_{new} - V_{old})$ lowers and decreases as $\Delta V$ increases, favoring acceptance of lower energies.

Multiple optimizations can be added to this method, for example, restricting the atoms to a sphere of a designated radius.

This method may appear similar to Simmulated Annealing, but it has an important advantage, that being the fact that it won't be trapped forever in a basin as keeping the temperature constant and adjusting the stepsize will guarantee the exit from a basin.

The Basin-Hopping algorithm has been used successfully to find the minima up to N=110 [10] although I have only tested it up to N=49 due to time and computational constraints.

## 3.3. Resumen

El método de optimización Basin-Hopping o salto de pozos, se basa en transformar la superficie de energía potencial (PES) en el mínimo local más cercano a cada punto mediante un algoritmo de minimización local.

Esta superficie transformada es explorada desplazando aleatoriamente las coordenadas y cada desplazamiento es aceptado de acuerdo a su probabilidad de Boltzmann (3.3). Una ventaja importante frente al método Simmulated Annealing es el hecho de que Basin-Hopping no se queda atrapado indefinidamente en un pozo correspondiente a un mínimo local.

# Capítulo 4

# Simulation

## 4.1. Introduction

In this work, we have analyzed both the Lennard-Jones (LJ) and the Improved Lennard-Jones (ILJ) potentials to find the global minima up to N=49 using the Basin-Hopping algorithm and compared it with the results previously obtained in [10].

## 4.2. Objectives

We seek to find the global minima of both LJ and ILJ potential for N up to 49 using the Basin-Hopping method.

## 4.3. Programming

All the results have been obtained using python 3.5.2, using the modules contained in the scipy.optimize Scipy [8] library, specifically, the *scipy.optimize.minimize(method='L-BFGS-B')* function. Used to find the local minima of a function.

The first task was to write both the LJ and the ILJ potential energies and their gradients in python. While the use of the gradient is not needed, using it allowed us to reduce the computational time of calculating the local minimum by a large margin, as most methods need the gradient and would resort to a numerical aproximation if not provided.

Once the potentials were correctly implemented, it was time to find the minima using the Basin-Hopping algorithm, which has to be written. Even though scipy has a basin hopping implementation, *scipy.optimize.basinghopping*, I wanted to implement it myself to be sure that I fully understood the ins and outs of the algorithm. After some tinkering, my implementation worked flawlessly but was really slow. Various local minimization methods were tried, most based in the Newton method (A.2), trying to reduce computational time, such as:

1. Conjugated Gradient Algorithm (CG)
2. Broyden-Fletcher-Goldfarb-Shanno Algorithm(BFGS)
3. Newton Conjugated Gradient Algorithm (Newton-CG)

4. Limited memory Broyden-Fletcher-Goldfarb-Shanno Algorithm (L-BFGS)

5. Truncated Newton Algorithm (TNC)

Then, various temperatures and starting step sizes were tested.

This testing was done by doing 10 optimizations of somewhat simple clusters, N = (7, 13, 17, 19) were chosen as they were simple enough that the testing would take a reasonable time but not simple enough that finding the minima would be trivial. The step sizes and temperatures tested in both cases were: (0.4, 0.6, 0.8).

The optimal parameters found were:

1. $T = 0{,}8$

2. $s = 0{,}4$

3. $method = L - BFGS$

The step size was the argument that affected less the time as it was later adjusted to obtain an acceptance rate of 0.5, and the method was the factor that affected time the most as it is a crucial and time consuming part of the process. It is important to point out that the TNC algorithm was almost as efficient as the L-BFGS algorithm.

The initial value of $\vec{X}$ was randomly chosen with the particles restricted to a sphere of fixed radius. This restriction was only to chose the starting configuration, the position of the particles was not restricted during the optimization.

Once we had implemented both algorithms and tested the most efficient ways to optimize all that was left was obtaining the results and checking them with the results already obtained by previous papers [10].

## 4.4. Results

In this section we are going to show the results and compare them with already known results from The Cambridge Energy Landscape Database [5].

### 4.4.1. Lennard-Jones

In figure 4.1 we can visualize the minimum potential energies per particle obtained. This results coincided with the ones obtained in [10], serving as confirmation that our algorithm was working correctly.

To obtain this data, the algorithm would run a number of steps, in the beginning we set that number to 1000, but, when analyzing the data some minima was higher than those found in [10], this meant that the global minimum was not found, instead the algorithm just stopped at a local minimum. To fix this problem, the number of steps was increased to 10000.

The number of iterations to find the minimum is shown in figure 4.2. The maximum is situated at N=38 which is an exceptional case as, while almost all of the clusters are based in a Mackay icosahedron (Figure 4.3) at the core covered by a an overlay, the N=(38,75,75,77,98,102,103,104) structures are not, with N=38 based in a truncated octahedron (Figure 4.4b). The algorithm is very likely to be trapped in a local minimum based in the Mackay icosahedron for a significant number of iterations (as shown in Figure 4.2), inflating the time taken to find the actual global minimum.

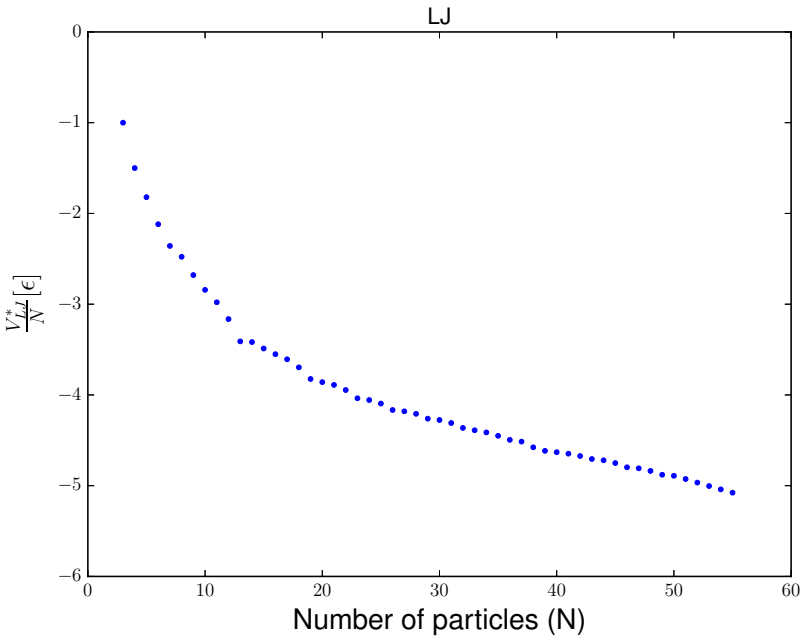



Figura 4.1: Potential energy per particle VS number of particles under the LJ potential

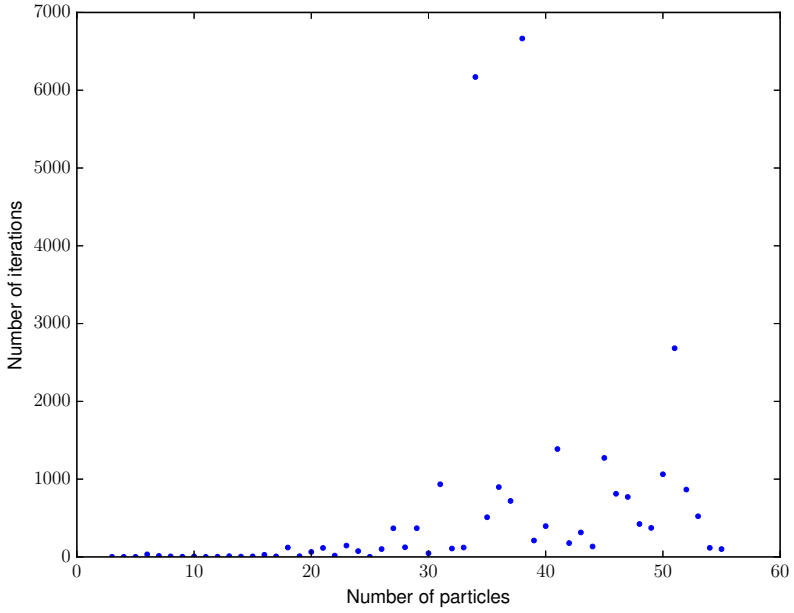



Figura 4.2: Number of iterations needed VS number of particles for the LJ algorithm.

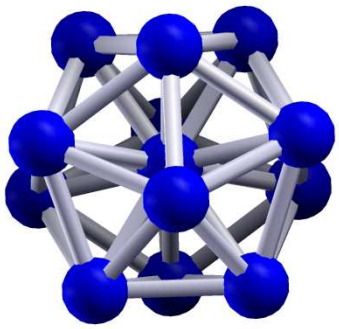
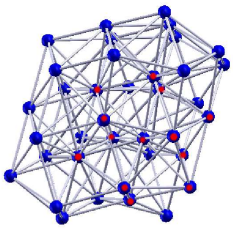
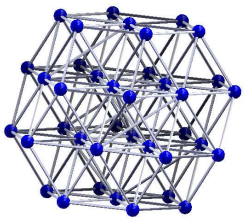


Figura 4.3: Basic Mackay Icosahedron for N = 13. This figures are obtained using XCrySDen[3]

.



(a) Mackay Icosahedron structure for N = 37 with the Mackay Icosahedron denoted by the red dots.



(b) Truncated octahedron structure N = 38.

## 4.5. Improved Lennard-Jones

When trying to optimize this potential we found a new problem, the algorithm fails to calculate the global minimum in the alloted 10000 steps, which is solved by increasing the number of steps to *80000*. This potential was also more time consuming to optimize.

As we can see in figure 4.5 the ILJ potential always has a higher energy per particle than the LJ potential.

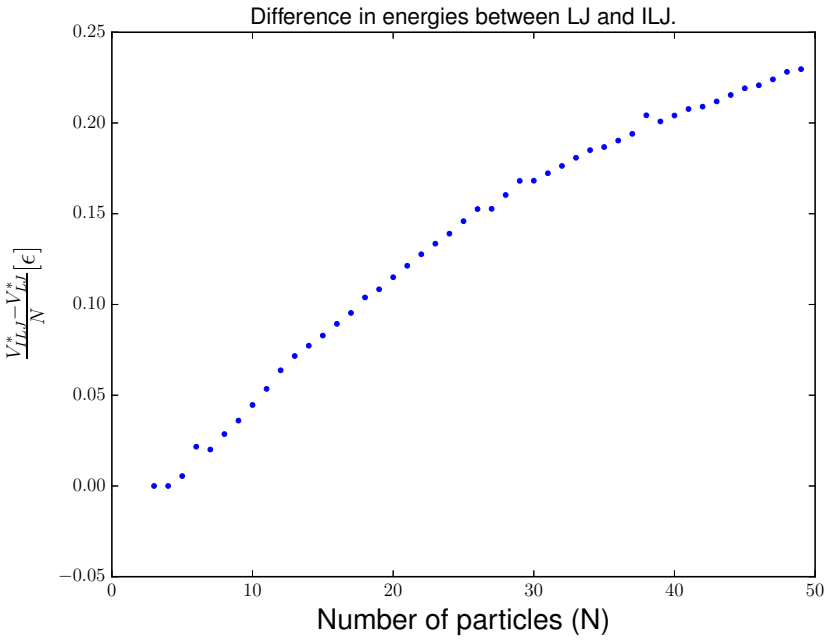



Figura 4.5: Difference between ILJ and LJ.

The second energy difference (equation 4.1) was also estudied, as shown in figure 4.6. This magnitude gives us an estimation of the stability of the cluster, having a maximum in N =13, as it is the full Mackay icosahedron, an extremely stable structure. The similarity of the second energy difference of the LJ and ILJ potentials, lets us state that there are no major structural changes between both potentials. However, one anomaly is found at N = 38, where there is a notable discrepancy between LJ and ILJ second energy difference, which may point to the Basin-Hopping method getting stuck in the ILJ potential on a local minimum, corresponding to a Mackay icosahedron structure, instead of finding the global minimum, associated to a truncated icosahedron. However, manually checking the results shows that the structure found is a truncated icosahedron, signaling that the true global minimum was found.

$$\Delta_2 V(N) = V(N+1) + V(N-1) - 2V(N) \tag{4.1}$$

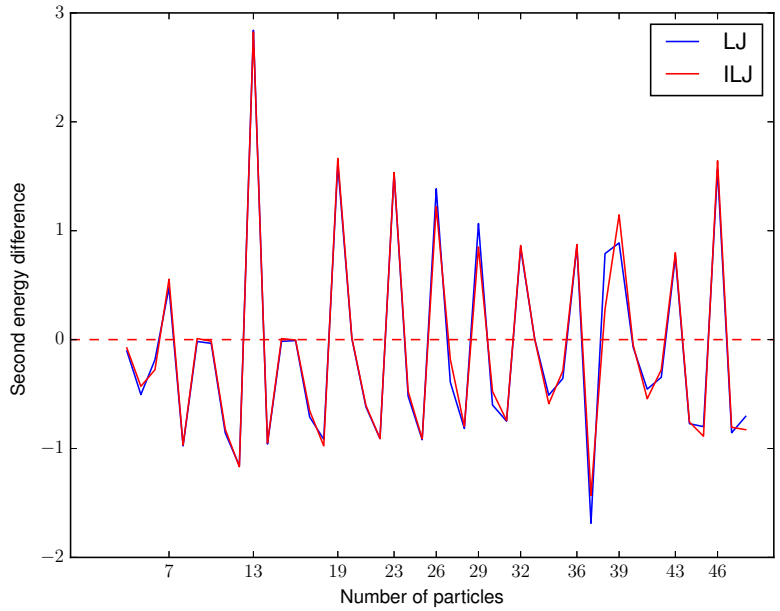



Figura 4.6: Second energy differences of LJ and ILJ potentials. The values on the x-axis are those corresponding to those with high stabilities.

## 4.6. Resumen

En este capítulo, obtenemos los mínimos globales tanto de el potencial LJ como del potencial ILJ (potencial Lennard-Jones mejorado).

Implementamos el algoritmo de minimización Basin-Hopping en python y resolvemos una serie de problemas que surgen, además optimizamos el algoritmo usando un método de optimización local muy eficiente.

Nuestros resultados para el potencial LJ coinciden con los obtenidos por otros investigadores [10], y observamos los distintos tipos de estructuras geométricas que se obtienen.

Realizamos el mismo estudio para el potencial ILJ y observamos que las diferencias entre ambos potenciales. Además, calculando las segundas diferencias observamos que ambos potenciales tienen las mismas estructuras.

# Apéndice A

# Appendix

## A.1. Metropolis-Hastings Algorithm

This algorithm is used to draw samples from a probability distribution (P(x)), as long as you can compute the value of f(x) proportional to the density of probability. This algorithm is mostly used to sample multi-dimensional distributions.

To initialize the algorithm, an arbitrary point, $x_0$, is chosen, and also an arbitrary probability density , $g(x|y)$, that suggest the next sample value x, given the previous one y. The function g is referred to as the proposal density or jumping distribution and must be symmetric.

For each iteration n, a candidate $x_*$ is picked from the proposal density g, and it is tested by calculating the acceptance ratio $\alpha = f(x_*)/f(x_n)$. Because f is proportional to the density of P, $\alpha = P(x_*)/P(x_n)$.

The candidate is accepted, $x_{n+1} = x_*$, with a probability of $\alpha$ if $\alpha \geq 1$ the candidate is automatically accepted. If the candidate is rejected, the next iteration is done with the previous candidate, $x_{n+1} = x_n$.

In our case, $g(x|y)$ is a linear distribution of length equal to the step size, as a the criteria for choosing the next candidate is defined by the equation (3.2).

Our f(x), is defined as:

$$f(x) = e^{-V(x)/kT} \tag{A.1}$$

This way, the acceptance rate, $\alpha$, results:

$$\alpha = e^{-(V(x_*)-V(x_n))/kT} \tag{A.2}$$

## A.2. Newton's Method

Newton's method, also known as Newton–Raphson method, is a method for finding successively better approximations of the roots of a function.

The function whose roots must be obtained, f(x), defined over the real numbers, must be derivable, an initial guess, $x_0$ is made, then a better approximation is, successively:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \tag{A.3}$$

With:

$$f^{'}(x) = \frac{df(x)}{dx} \tag{A.4}$$

To use the Newton's method to find the minimum of a function, the method is applied to find the roots of $f^{'}(x)$ instead of $f(x)$.

## A.3. L-BFGS

Trying to use the Newton's method to find the local minimum of a function falls into the problem that the second derivative of a function is needed, and in our case, it being a multi variable problem, the inverse Hessian matrix is needed.

The L-BFGS method, is a modification of the BFGS (Broyden–Fletcher–Goldfarb–Shanno) method, which belongs to the family of quasi-newtonians methods.

The original BFGS, uses an estimation to the inverse Hessian matrix, it being a dense $n \times n$ matrix. The limited (L) BFGS only stores a few vectors of the inverse Hessian, being them the last $m$ updates of both the position vector, $\vec{x}$, and the gradient, $\vec{\nabla} f(\vec{x})$.

For more information on this algorithm, consult [6]. As it is a complex algorithm and explaining it goes beyond the scope of this work.

# Bibliografía


[1] Pierre de Fermat. «Methodus ad disquirendam maximam et minimam and in De tangentibus linearum]». En: (1636).

[2] Bernd Hartke. «Global geometry optimization of clusters using genetic algorithms». En: *The Journal of Physical Chemistry* 97.39 (1993), págs. 9973-9976.

[3] Anton Kokalj. *XCrySDen*. 1999. URL: http://www.xcrysden.org/.

[4] Joseph-Louis Lagrange. «Leçons sur le calcul des fonctions». En: (1804).

[5] *Lennard-Jonnes results*. 1999. URL: http://doye.chem.ox.ac.uk/jon/structures/LJ/tables.150.html.

[6] *Numerical Optimization: Understanding L-BFGS*. 2014. URL: http://aria42.com/blog/2014/12/understanding-lbfgs/.

[7] Fernando Pirani y col. «Beyond the Lennard-Jones model: a simple and accurate potential function probed by high resolution scattering data useful for molecular dynamics simulations». En: *Physical Chemistry Chemical Physics* 10.36 (2008), págs. 5489-5503.

[8] *Scipy*. 2016. URL: http://www.scipy.org/.

[9] S. Kirkpatrick; C. D. Gelatt Jr; M. P. Vecchi. «Optimization by Simulated Annealing». En: *Science* 220 (1983), págs. 671-680. DOI: https://dx.doi.org/10.1063%2F1.1699114.

[10] David J Wales y Jonathan PK Doye. «Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms». En: *The Journal of Physical Chemistry A* 101.28 (1997), págs. 5111-5116.

[11] LT Wille. «Simulated annealing and the topology of the potential energy surface of Lennard-Jones clusters». En: *Computational materials science* 17.2 (2000), págs. 551-554.