A NON-QUANTUM MONTE CARLO METHOD FOR FINDING MOLECULAR SHAPES

ANAS MAMDOUH ELWASIFI

1. Introduction

Molecular shapes heavily influence both biological and chemical activities. In 1894, Fischer's ”Lock-and-key” model proposed that certain molecules have specific shapes which allow them to complement other molecules, activating biological processes [GC12]. In chemistry, reaction mechanisms depend on the shapes of molecules. Reactions proceed step-by-step depending on the collision of the right faces of both reactant molecules, causing certain particles to interact.

As a result, finding the exact shapes of molecules and bond angles has always been a hotspot for research. Despite this, it wasn't until 1912 that William and Lawrence Bragg applied Max von Laue's discovery of X-ray diffraction in creating 3D models of molecules [Unk]. Although Bragg's techniques accurately found the shapes of heavy molecules, it couldn't find the shapes of molecules containing hydrogen as hydrogen diffracts X-rays very weakly. In 1957, VSEPR (Valence Shell Electron Pair Repulsion) theory was introduced by Ron Nyholm and Ronald Gillespie in an article [Gil07]. It provided a simple method to estimate the shapes of molecules by considering lone electron pairs. This overcame previous barriers, where it estimated the angles and shapes of all molecules; specifically, those containing hydrogen. However, VSEPR's simplicity was at the cost of its precision to determine bond angles accurately (for example, H-O-H bonds in water were found to be 109.5° instead of the modern 104.5°).

While many methods have come after VSEPR, most utilize quantum mechanics making them computationally intensive. Thus, methods with near-quantum accuracy that balance both computational simplicity and accuracy of determining molecular shapes need to be developed.

The paper proposes to do this by using a non-quantum algorithm based on Monte Carlo, specifically the Metropolis Markov chain Monte Carlo (MCMC) method with annealing. Monte Carlo simulations are known for their brute-force methods to solve complex problems through random sampling, while annealing gradually decreases the randomness of the process. By applying this approach, the configuration with the lowest electrostatic potential energy will be found, which corresponds to the optimal molecule shape. The approach will work on any type of molecule, as it doesn't rely on X-ray diffraction nor simplified ideas as in the case of VSEPR.

To do this, the algorithm first makes 1000 random throws of the molecule's protons and electrons' positions, saving the distribution with the least potential energy to perform annealing on. The annealing process is done by choosing a particle and performing a random walk, where the potential energy of the resultant electron-proton distribution is recalculated. If the step decreases the system's potential energy without stepping over its boundaries, the step is accepted. Otherwise, it'll be accepted with a probability related to the temperature of the system and the energy increase. The algorithm is run until the temperature of the system reaches 5.

2. Methodology

The goal is to find the shape of molecules using only their number of electrons, lone pair electrons, and protons. By minimizing the potential energy of these systems, we can find the optimal configuration of the molecule's constituent particles, and thus an approximation of its real shape. For simplicity, 2 main assumptions are made:

(1) The potential energy of the system is modeled using Coulomb's law of electrostatic energy shown in equation 1

$$U = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \frac{q_i * q_j}{4\pi\epsilon_0 \, r_{ij}} \tag{1}$$

where q_i and q_j are the charges of particles i and j, ϵ_0 is the vacuum permittivity, and r_{ij} is the distance between particles i and j. The potential energy is calculated by summing the potential energy of all pairs of particles in the system.

(2) Lone pair electrons, which consist of two electron charges bound together, and protons are simplified as singular particles with equivalent charges. Thus, lone pair electrons are considered only one particle with a double negative charge ($-2e$), and protons as one particle charged positively with the number of protons of the atom.

The algorithm consists of 2 essential phases. In phase one, the goal is to get a random particle distribution that we can optimize later on. This is done by generating 1000 distributions using random throws in a cube of length 1×10^{-11}, where only the molecule's central atom is fixed at the origin point. The configuration with the lowest potential energy is saved for phase 2. An example of a resulting distribution is shown in Figure 1.

In the second phase, particles are chosen with a probability directly related to their charge (higher charged particles are more likely to be chosen and vice versa). A chosen particle is moved by small movements to try and find more optimal shapes for the molecule. While doing this by only accepting moves that decrease the potential energy is possible, the system will start falling into local minima, specifically as the system's boundaries are sufficiently larger than the particles themselves.

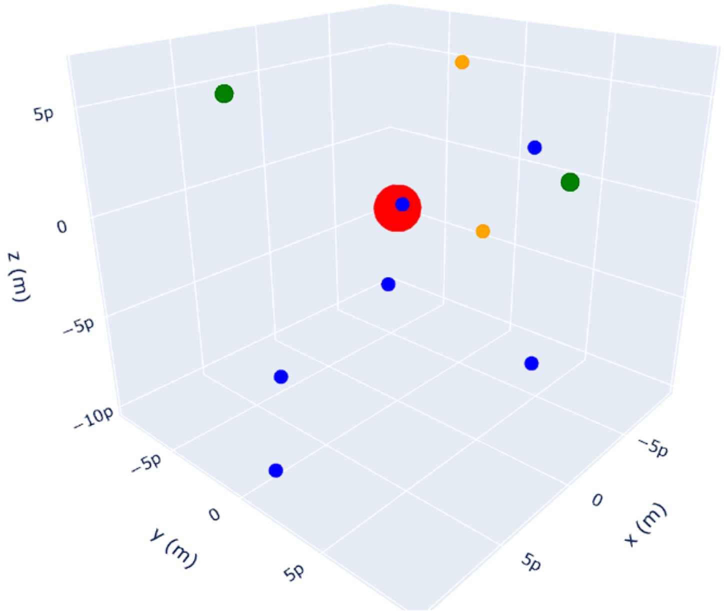

Figure 1. *An example of a random distribution of particles in a molecule.*

The easiest way to escape these local minima is to use simulated annealing, essentially allowing the system to act as a molten metal that begins chaotically with its particles moving around. Then, as it cools off over time, the metal particles lock into their regions and move only in them until it becomes solid. In our system's terminology, the particles will have a high probability of accepting "up-hill," potential-increasing movements at the start of the simulation, while also accepting any "down-hill," potential-decreasing movements. This allows the system to escape any local minima it might be inside in search of a global minima. As the system cools off, the up-hill movement probability decreases, allowing the particles to lock into their places.

The up-hill movement probability is calculated by representing both the energy state our system is in and the one it'll go to by using Boltzmann distributions shown in equation 2:

$$P(E_{a_i},T)=\frac{e^{\left(\frac{-E_{a_i}}{k_BT}\right)}}{Z(T)}, \quad P(E_{a_f},T)=\frac{e^{\left(\frac{-E_{a_f}}{k_BT}\right)}}{Z(T)} \tag{2}$$

where both E_{a_i} and E_{a_f} represent the initial and final energy states of the system, k_b is the boltzmann constant equivalent to 1.38×10^{-23}, and T is the system's temperature. As we don't have the partition function $Z(T)$, we divide both the Boltzmann distribution equations to get a relative probability, as in equation 3, that tells us how likely the system is to transfer from the initial energy state to the final.

$$p = \frac{\frac{e^{\left(\frac{-E_{a_i}}{k_B T}\right)}}{Z(T)}}{\frac{e^{\left(\frac{-E_{a_f}}{k_B T}\right)}}{Z(T)}} = e^{\left(\frac{-\Delta E}{k_B T}\right)} \tag{3}$$

If the system's initial temperature is too high, it'll over explore and likely cool off in a physically impossible distribution; for example, nuclei having angles less than 1 degree. Low initial temperatures just mean that it won't explore enough and thus get stuck in a local minima. Thus, depending on each molecule's complexity, the initial temperature of the systems was determined by trial and error.

As soon as any particle neither goes up-hill nor down-hill for 4 consecutive moves, another particle is chosen to undergo the same process of random movements. After the algorithm goes through a number of particles equal to the system's total number of particles, it'll cool off by multiplying the temperature with a constant 0.9 as shown in equation 4.

$$T_{\text{new}} = T_{\text{old}} \times 0.9 \tag{4}$$

While it may seem useful to increase the number of consecutive moves before choosing another particle, doing that risks an electron getting chosen first, which then surrounds and effectively traps the nucleus in its spot. Making each particle move a small number of moves is better as it gives all particles the opportunity to explore before coming together.

A move's amplitude is determined by generating a random number between 1×10^{-14} and 1×10^{-11}. Thus, relative to the system's maximum boundaries of $\pm 5 \times 10^{-11}$, a move is between five thousandth and a fifth of the boundaries. All amplitudes are further multiplied by a ratio of the system's current temperature and initial temperature to allow for finer movement later as the particles get closer to each other. The maximum distance any two particles whether electrons or nuclei can get to each other is 1×10^{-13}, a tenth of a picometer.

With each move, the potential energy of the system is updated by removing the particle's contribution to the potential before the movement then adding it after the movement, rather than recalculating the entire system. This makes the code faster. Subtractive cancelation is negated by recalculating the entire system's energy ground-up before any new particle is chosen for movements.

When the system's temperature reaches 5 kelvin, the algorithm stops and the final distribution is saved. Due to the many local minima, the previous process is repeated 20 to 30 times and the lowest potential energy distribution is saved.

3. Results and Discussion

Some molecules with different shapes were retrieved from PHeT [col] and Libretexts [Lib23] and simulated using the algorithm to see if they aligned with the expected, real angles. The results are as shown in Table 1. All the distributions can be accessed here in 3D interactive form.

Type	Molecule	Expected angle/s	Initial temperature	Optimal angle/s (least potential energy)	Second closest run
Linear	BeH_2	180°	200	140.0°	163.0°
Linear	CO_2	180°	700	117.1°	161.0°
Trigonal Planar	BF_3	120°, 120°, 120°	900	50.1°, 76.5°, 126.4°	111.0°, 168.5°, 80.5°
Trigonal Pyramidal	NH_3	107.8°, 107.8°, 107.8°	400	141.1°, 47.2°, 98.5°	102.9°, 100.2°, 106.2°
Bent	H_2O	104.5°	400	119.8°	115.0°
T-shaped	ClF_3	87.5°, 87.5°, 180°	900	107.5°, 103.4°, 121.1°	82.8°, 66.0°, 111.0°

Table 1. *Results of the algorithm on different molecules. The expected angles are compared to the optimal angles found by the algorithm, which are the angles of the distribution with the lowest potential energy. The second closest run is the distribution with the second lowest potential energy.*

One thing that's prevalent throughout all the particles is that the optimal distribution is almost always less accurate than other less optimal distributions. This is expressed the most in NH_3. Figure 2 shows both the optimal distribution and second optimal distribution. While it's tough to 100% confirm this, it's suspected that this happens because the optimal distributions would not be physically possible. On the other hand, the other results the algorithm outputs do follow expected shapes and angles.

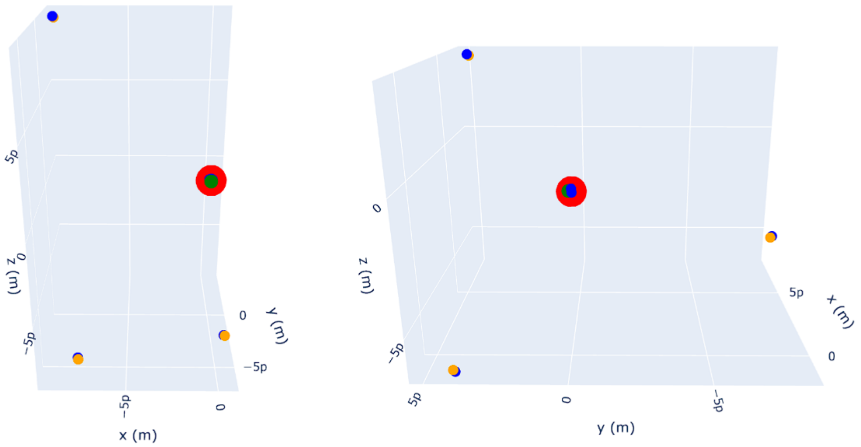

Figure 2. *Red particles demonstrate Nitrogen, yellow hydrogen, and green electrons. Left: Optimal distribution with a potential energy of* -8.886×10^{-14} *J, where the nitrogen atom is outside the shape formed by the hydrogen atoms. Right: Second optimal distribution with a potential energy of* -8.878×10^{-14} *J, where the nitrogen atom is centrally positioned with hydrogen atoms distributed around it, yet the potential energy is higher.*

This behavior also happens to the T-shaped ClF_3, but in a more extreme manner as the most optimal distribution shows no resemblance to the expected shape. Figure 3 shows the optimal distribution with potential energy $\approx -8.146 \times 10^{-13}$ along with another distribution with potential energy $\approx -7.784 \times 10^-13$. With only a difference in potential energy of -8×10^-17, it's expected that both runs will be similar. However, The optimal distribution isn't T-shaped but its hydrogen is instead uniformly around the central particle. Surprisingly, the other distribution is 8th optimally yet perfectly captures the right shape of the molecule.

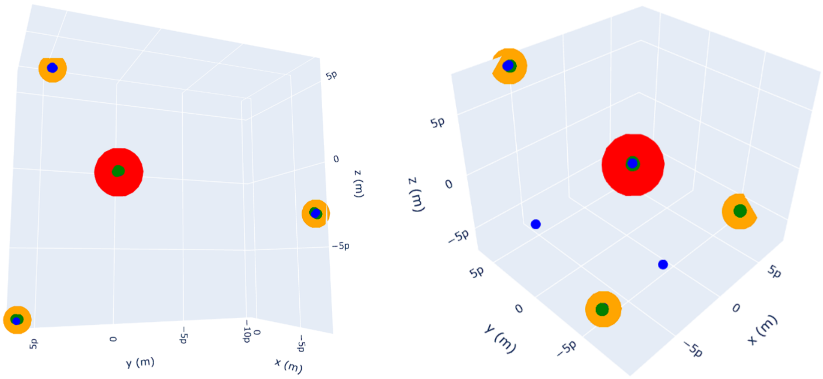

Figure 3. *Red particles demonstrate chlorine, yellow fluorine, green lone pair electrons, and blue electrons. Left: Optimal ClF_3 distribution with the chlorine atom behind the fluorine and in the middle. Right: 8th optimal distribution with both the chlorine and fluorine almost perfectly creating a T-shape*

4. Recommendations

It's clear from the results that while using potential energy to find the optimal shapes of molecules didn't strictly output the correct, final distributions, it was still capable of getting them. Thus, the current algorithm needs to become more precise and accurate. This can be done by implementing the following changes:

- **Metrics to distinguish between possible and impossible distributions:**
  By retaining the same idea of using potential energy minimization for solution exploration and adding other metrics such as nuclei angles, it'll be possible to make the results more accurate and robust at the same time, while also allowing for the achievement of similar or better results in lower runtime.

- **Adaptive cooling schedule based on system state:**
  Switching to an adaptive cooling schedule that changes based on the averaged distance of electrons from nuclei could also help make the code both faster and decrease the probability of electrons escaping systems randomly. It would also make sure that the algorithm doesn't stop prematurely.

- **Replacing the current random throw method with roughly estimated distributions:**
  Trying to get rough starting positions of all or some of the system's nuclei as a function of the system's energy would help by decreasing the number of distributions and local minima can fall into, thus allowing for more accuracy and faster runtime.

5. Conclusion

In conclusion, the algorithm demonstrated that it could approximate molecular shapes across many different geometries, including linear, trigonal planar, trigonal pyramidal, bent, and T-shaped molecules. While it successfully identified plausible configurations, the most optimal distributions did not always correspond to the expected angles, indicating limitations in the current model. Thus, while it does provide a solid foundation and starting point for non-quantum computational methods in molecular geometry prediction, it still needs further improvement to enhance its accuracy, reliability, and runtime.

References

[col] colorado.edu. *Molecule Shapes — PhET*. URL: https://phet.colorado.edu/sims/html/molecule-shapes/latest/molecule-shapes_all.html.

[GC12] Joseph Gal and Pedro Cintas. "Early history of the recognition of molecular biochirality". In: *Topics in current chemistry* (Jan. 2012), pp. 1–40. DOI: 10.1007/128\{_}2012\{_}406. URL: https://pubmed.ncbi.nlm.nih.gov/23274573/.

[Gil07] R.J. Gillespie. "Fifty years of the VSEPR model". In: *Coordination Chemistry Reviews* 252.12-14 (July 2007), pp. 1315–1327. DOI: 10.1016/j.ccr.2007.07.007. URL: https://www.sciencedirect.com/science/article/abs/pii/S0010854507001476.

[Lib23] Libretexts. *1.4: Development of chemical bonding theory*. Oct. 2023. URL: https://chem.libretexts.org/Courses/Athabasca_University/Chemistry_350%3A_Organic_Chemistry_I/01%3A_Structure_and_Bonding/1.04%3A_Development_of_Chemical_Bonding_Theory.

[Unk] Unknown. *X-ray crystallography: Revealing our molecular world — Science Museum*. URL: https://www.sciencemuseum.org.uk/objects-and-stories/chemistry/x-ray-crystallography-revealing-our-molecular-world.