

An Exploration of Computational Empirical and Quantum Chemical Methods

Masood Malekghassemi

Abstract

Computational chemistry is no new subject however, the idea of taking the process from including a human being in the loop between ab initio methods and general rule making for molecular mechanics calculations to excluding that same being via AI seems to be unimplemented. This project revolves around the idea that molecular mechanics is an incredibly fast and viable way of calculating the geometry of molecules and that as such, given the proper parameterizations and generalities, can be as accurate as or more accurate than ab initio methods. The final product is a program capable of performing both and transferring information between the two classes of methods. The journey to the final product should include much self-taught knowledge regarding the inner workings of the quantum mechanical model of chemicals.

Background

Computational chemistry is a form of chemistry that takes physical rules and applies them to virtual models in a virtual space. The chemist enters a geometry and queries the computer with regards to its various conformational energies and electron densities to try to understand how it is that the molecule behaves.

There are three different categories of computational chemistry separated based on how far removed they are from the fundamental quantum mechanical equations. The 'ab-initio' methods start from first principles and calculate the inner workings of a system. The 'semi-empirical' methods still work from first principles, but take in extra information from empirical studies and let go of certain calculations to make computation easier (such as scrapping all electrons but those in the valence shell from consideration). The 'empirical' methods work directly from empirical data, working off of a so called 'ball and spring' model (each bond is a spring parameterized by experiment in a set of parameterizations called the 'force-field').

Relevant Quantum Chemical Equations

Time-Independent Schrodinger Equation

$$H\psi = E\psi$$

Time-Independent Energy

$$E = \frac{\langle \psi | H | \psi \rangle}{\langle \psi | \psi \rangle}$$

Electronic Hamiltonian

$$H = -\sum_i \nabla^2 R_i - \sum_i \nabla^2 r_i - \sum_i \sum_j \frac{Z_i Z_j}{R_i - R_j} - \sum_i \sum_{j>i} \frac{1}{r_i - r_j} - \sum_i \sum_{j>i} \frac{Z_i Z_j}{R_i - R_j}$$

Roothaan Equation

$$FC = SC\epsilon$$

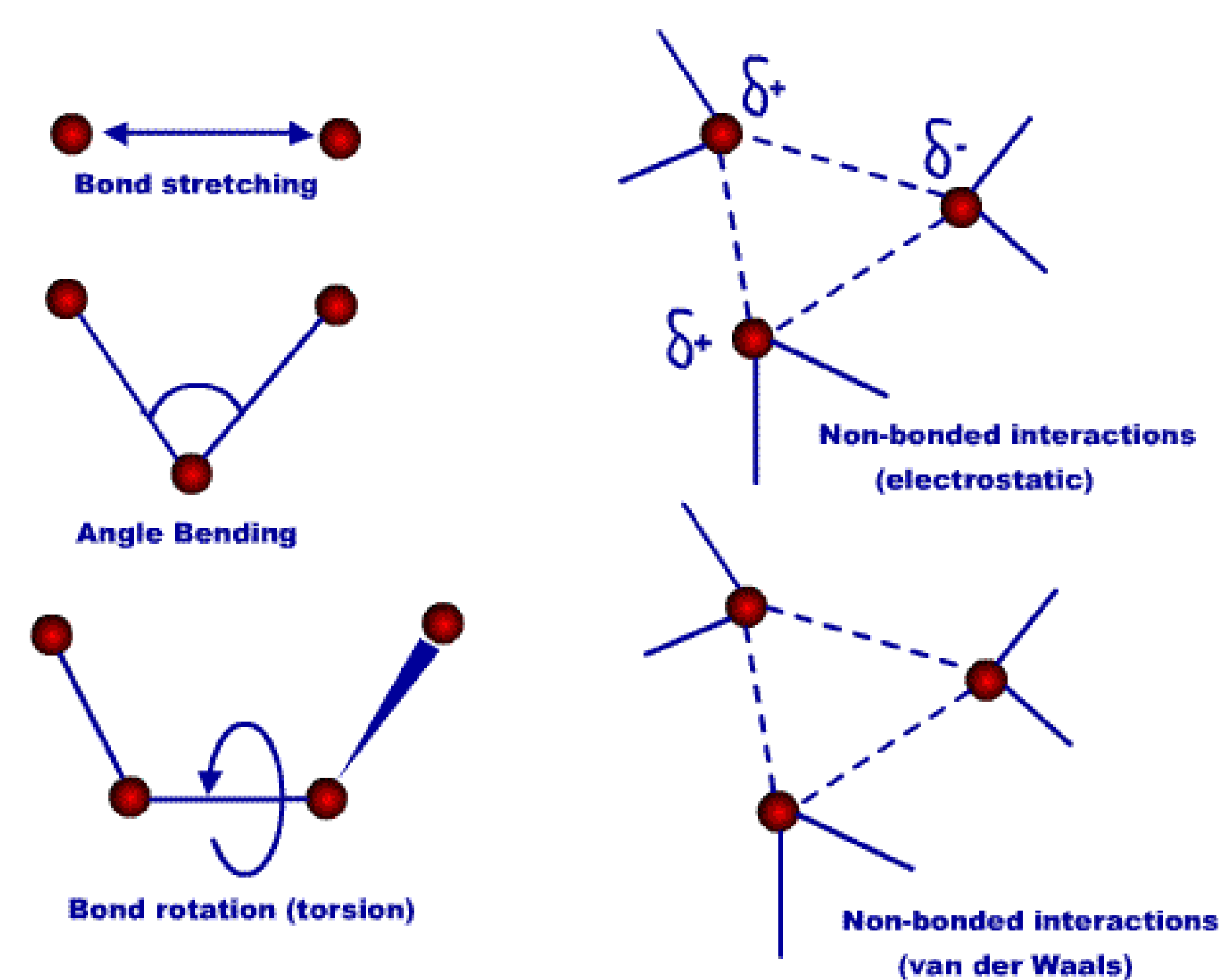


Figure 1: Molecular Mechanics Force-Field, taken from Stanford Folding@Home

The (unfinished) Project

Empirical Methods

The project aimed to have the most generic framework possible with regards to the empirical methods, such that arbitrary patterns of data interpreted by a human or, in the case of this project's final aim, the machine, may be more flexibly translated to the force-field that governs the simulation. This brings to bear the concept of a 'rule', that is, an arbitrary chunk of functionality that applies itself to the simulation and increases or decreases the energy and forces and whatever other parameters the rule can modify. This allows for such algorithms as hill-climbing or genetic algorithms to gradually find the lowest energy arrangement (conformation) of the atoms in a given well-parameterized molecule by interacting with the rules.

Ab-Initio Methods

The ab-initio methods are math intensive and, to solve them precisely, requires retaining the analytic form of the integrals brought out from the relevant set (shown in the lower left of this poster) up until the point of numeric evaluation. As an example, the overlap integral of a gaussian function multiplied by a monomial (from the matrix of overlap integrals of individual atomic electron wavefunctions 'S' in the Roothaan equation) with another atomic wavefunction of the same type centered on a different atom across all space (integrating from $-\infty$ to ∞ in all dimensions) is that of a polynomial times a gaussian function, and that in itself is a sum of products of double factorials. This is managed via analytic polynomial classes.

The actual process for calculating the energies of the molecular conformation involves finding the integrals (or approximating, as with some semi-empirical methods) of the Fock matrix (F in the Roothaan equation). This is like inserting the parts of the Hamiltonian that apply to electrons (the ones with ' r_i 's) in between the two wavefunctions of the overlap integral and then evaluating. Due to the difficulty involved with this (understanding the derivations for the analytic evaluation of the integrals to properly implement them), however, they're not currently implemented. The next step would be to solve for the energy via diagonalization of the Fock matrix. Because the Fock matrix is symmetric, one can use Jacobi diagonalization, implemented via visitor pattern (as in, Gang of Four) to the generalized matrix classes whose type traits (as in, template metaprogramming - Java programmers don't have this :-P) identify them as being symmetric.

Truncated list of Programmatic Techniques

Variadic Function Argument Iterator: takes C variadic functions and adapts the C macros to that of a C++ iterator, allowing variadic functions to pass their arguments C++ style to the vast number of iterator algorithms in both the standard library and elsewhere. Has not been tested for performance, that in itself is another research project.

Type Traits: a 'standard' yet advanced way of statically determining the capabilities of classes, allowing for a more flexible, yet somewhat more complicated, way to overload classes and functions. This is way of investing in the extensibility of the code in the future. It falls under 'C++ template metaprogramming', and is absent from other languages either due to a lack of templates or because the language does not require these kinds of facilities (ie. the type system is dynamic).