

# Project 3: Molecular Mechanics

March 8, 2021

One of the major problems in quantum chemistry is calculating the electronic energy for a given nuclear configuration, a task becoming increasingly demanding for larger and larger molecules. Force field methods bypass this step by calculating the electronic energy directly as a parametrized function of the nuclear coordinates, i.e. without explicitly considering electrons as individual particles. Molecular structures correspond to the minima of this energy function.

Goal of this exercise is writing a code for the parametrization of such an energy function from a given molecular structure and its subsequent minimization.

## I. Theory

The energy function can be expressed as the sum of certain contributions describing the energy required to distort a molecular structure in a specific form. A simple expression would be

$$E_{FF} = E_{str} + E_{bend} + E_{tors} + E_{nb} , \quad (1)$$

where the first term on the *r.h.s* of (1) describes the energy for *stretching* a bond, the second one the energy for *bending* an angle, the third one represents the *torsional* energy for rotating around a bond and finally the last term represents all *non-bonded* interactions between the atoms.

### I.1. The Stretch Energy

The stretch energy is a function of all pairs of atoms in a molecule. For each pair it can be obtained as a Taylor expansion around the equilibrium bond length which in the harmonic approximation is terminated after the quadratic term,

$$E_{str}^{AB} = E_{str} \left( R^{AB} - R_0^{AB} \right) = E(0) + \frac{1}{2} \frac{d^2 E}{dR^2} \left( R^{AB} - R_0^{AB} \right)^2 , \quad (2)$$

where the linear term is zero around the equilibrium value and  $E(0)$  is just an additive constant which is usually set to zero. Thus, we might write

$$E_{str}^{AB} = k^{AB} \left( R^{AB} - R_0^{AB} \right)^2 , \quad (3)$$

where  $k^{\text{AB}}$  is the *force* constant. It is well known, that the harmonic approximation is not very accurate for large displacements from the equilibrium geometry. However, it has the advantage, that for each bond type, only two parameters  $k^{\text{AB}}$  and  $R_0^{\text{AB}}$  needs to be determined.

## I.2. The Bending Energy

The bending energy term is a function of all triples of atoms and might in the harmonic approximation be expressed as

$$E_{\text{bend}}^{\text{ABC}} = k^{\text{ABC}} \left( \theta^{\text{ABC}} - \theta_0^{\text{ABC}} \right)^2 . \quad (4)$$

For a arrangement of three atoms  $A, B, C$ ,  $\theta_0^{\text{ABC}}$  can be calculated from the formula

$$\theta_0^{\text{ABC}} = \frac{\left( R_0^{\text{AB}} \right)^2 - \left( R_0^{\text{AC}} \right)^2 - \left( R_0^{\text{BC}} \right)^2}{2 R_0^{\text{AC}} R_0^{\text{BC}}} . \quad (5)$$

Notice, that also this ansatz is only valid near the equilibrium geometry.

## I.3. The Torsional Energy

A torsional angle is a function of four atoms  $A, B, C, D$  and might be describe as the angle between the  $A - B$ - and  $C - D$  bond, looking down the  $B - C$ -bond. Contrary to the bending energy, the torsional energy must be a periodic function, as after a rotation around  $360^\circ$  the energy must be unchanged. To this end, the usual ansatz for  $E_{\text{tors}}$  is a Fourier series,

$$E_{\text{tors}}(\omega) = \sum_{n=1} V_n \cos(n\omega) , \quad (6)$$

where for each term,  $V_n$  describes the size of the barrier for a rotation around the  $B - C$ -bond. Usually, the zero point of the potential is shifted by adding a factor of 1 to each term and often truncating the Fourier series after a few terms already gives good results. In your implementation you will employ the simpler ansatz

$$E_{\text{tors}}(\omega^{\text{ABCD}}) = \frac{1}{2} V_1^{\text{ABCD}} \left[ 1 + \cos \left( n\omega^{\text{ABCD}} - \gamma \right) \right] . \quad (7)$$

## I.4. Nonbonded Interactions

Nonbonded interactions also play an important role in determining the structure of individual molecules. They might be divided into Van-der Walls' interactions and electrostatic interactions. In the simplest model, the latter one might be modelled using partial atomic charges and employing Coulomb's law,

$$E_{\text{nb,elec}} = \sum_{j>i}^N \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} , \quad (8)$$

where  $r_{ij}$  denotes the distance between two of the  $N$  atoms. The other important part of non-bonded interactions are Van der Waals forces interactions which are typically most difficult to model. They might be described as

$$E_{vdw} = \sum_{i < j} \left[ \frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^6} \right] , \quad (9)$$

### 1.5. Minimizing the Energy

After the energy function is set up, it can be used to minimize it and find the most stable arrangement of all atoms in a molecule. Most minimization algorithm works with (either numerical or analytical) gradients, as described in project 4. In this project, you will resort to the so-called *Metropolis* algorithm which has the big advantage that no gradients need to be evaluated. It can be formulated as follows:

1. start from some position on the energy function  $\mathbf{x}_i$ .
2. Chose a new vector according to

$$\mathbf{y} = \mathbf{x}_i + r\mathbf{q} ,$$

where  $r$  is some constant search radius and  $\mathbf{q}$  is a vector of random number between  $-1$  and  $1$ .

3. Calculate the energy difference

$$\Delta E = E(\mathbf{y}) - E(\mathbf{x}_i) .$$

Accept the new position with the probability

$$p_A = \min \left( 1, \exp \left( \frac{-\Delta E}{k_B T} \right) \right) .$$

This means, if  $\Delta E < 0$  always accept the new position,  $\mathbf{x}_{i+1} = \mathbf{y}$ . if not, chose a random number  $q$  between 0 and 1. If  $q < p_A$  then  $\mathbf{x}_{i+1} = \mathbf{y}$ , otherwise  $\mathbf{x}_{i+1} = \mathbf{x}_i$ .

The radius  $r$  can be varied. Note that a too small  $r$  will lead to slow convergence while a too big  $r$  makes the algorithm numerically unstable (it can happen with a certain probability that the algorithm jumps out of a already found minimum). Obviously, you need to provide some temperature  $T$  for this algorithm to work. A lower temperature will lead to a lower possibility of leaving a given minimum as the energy for passing a barrier must be higher.

Of course your algorithm must be terminated at some point. Think about some good termination criterion.

## II. Implementation

Due to the rapid growth of the number of parameters needed for a force field capable of describing molecules consisting of many types of atoms, in this projects we will implement force field parameters for molecules containing only carbon and hydrogen atoms and only single bonds (i.e. linear or cyclic alkanes). The corresponding parameters can be found in the paper by Kollman and co-workers for all contributions to the energy function. How you store this data is up to you. Options are

- Declaring all parameters in a **DATA** block,
- writing them to file and reading them in any time you start your code.

You will be provided with input files containing the number of atoms, the involved type of atoms and their Cartesian coordinates. After reading in these information from the file you need to store them. To this end, it might be worthwhile to create a type **atom**, containing the type of atom (Carbon or Hydrogen) and its coordinates. Next, you need to set up your energy function. It is part of this project to think about how this can be done in an efficient way. However, some hints are given in the following.

- Identify all bonds in the system and bond types (CC,CH). You might loop over all atoms and calculate their distances. A pair of atoms is then considered as bonded when their distance is (within some tolerance) equal to a typical (CC or CH) bond length.
- To the bending and torsional energy, only bonded triples (quadruples) of atoms which are bonded contribute. Think about a way how you can avoid to loop over all atoms (triples - three nested loops, quadruples - four nested loops) in the system *after* you figured out which atoms are bonded.
- keep in mind that your energy function must always be a function of  $3N - 6$  coordinates, where  $N$  is the number of atoms.

After you set up you energy function you can set up the Metropolis algorithm in a separate module. You might vary  $r$  and check the convergence properties of your algorithm.