BACHELOR PROJECT SCIENTIFIC COMPUTING

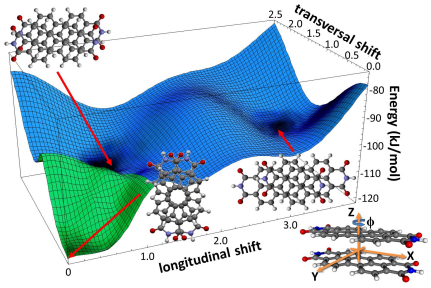
Molecular Geometry Optimizations

Problem Description

A key step in many applications of material science modeling on a molecular level (for, e.g., molecular electronics, drug design, catalysis, etc.) is the optimization of molecular structures, i.e., the structural arrangement of the atoms forming them. Achieving this requires stable, efficient, and therefore, elaborate optimization algorithms.

The optimal geometry of a molecule is the one that minimizes the total energy E of the system. If the molecule consists of $i = 1, \ldots, N$ atoms, each described by its three Cartesian coordinates $\mathbf{r}_i = (x_i, y_i, z_i)$, E is a scalar function of all atomic coordinates $E = E(\mathbf{r}_1, \ldots, \mathbf{r}_N)$. It describes a $3N$-dimensional hypersurface, also often referred to as *Potential Energy Surface* (PES), and can exhibit several distinct minima and/or saddle points. The geometry optimization problem is equivalent to finding the minima of the PES, in other words

$$\{x_i, y_i, z_i\} = \arg\min\{E = E(\mathbf{r}_1, \ldots, \mathbf{r}_N)\}$$



Example of a PES with several minima (from: www.quantenchemie.tu-berlin.de.

Very often, gradient optimization methods are used which are based on quadratic approximations to the PES at a point with energy E_0 in the form

$$E^* = E_0 + \mathbf{g}^T\mathbf{s} + \frac{1}{2}\mathbf{s}^T\underline{\underline{\mathbf{H}}}\mathbf{s}$$

where E^* is the predicted energy at step $\mathbf{s}$ away from the current point, and $\mathbf{g}$ and $\underline{\underline{\mathbf{H}}}$ are the gradient and Hessian at the current point, respectively. In many cases the exact calculation of in particular second derivatives of the total energy from accurate quantum-mechanical techniques is computationally intractable. Sometimes even no closed form expressions for energy gradients exist and one has to resort to costly numerical differentiation techniques. It is therefore extremely important to have optimization algorithms that find a minimum with high accuracy in as few inexpensive steps as possible. Typical algorithms include, e.g., gradient descent, conjugate gradients, Newton-Raphson methods, quasi-Newton methods, rational function optimizations, and trust radius models.

Tasks

- Study literature (provided by the supervisor).
- Implement several methods in Matlab or Python and apply them to simple but representative problems.
- Analyze the convergence behavior and accuracy of the different methods.

Requirements

- Linear algebra
- Basic programming skills in Matlab or Python

Supervisor

Björn Baumeier—MF 7.142—040-2472205—b.baumeier@tue.nl