Dive into computational physical chemistry

Lecture 9: Intro to density functional theory and cheminformatics

Glen Hocky
November 22, 2022
**Reminder: Electronic structure in Born Oppenheimer approximation**

\[ \hat{H} \Psi = E \Psi \]

\[ \hat{H} = -\frac{\hbar^2}{2m_e} \sum_i \nabla_i^2 - \frac{\hbar^2}{2} \sum_A \frac{\nabla_A^2}{2M_A} - \sum_{A,i} \frac{Z_A e^2}{4\pi\epsilon_0 r_{Ai}} + \sum_{A>B} \frac{Z_A Z_B e^2}{4\pi\epsilon_0 r_{AB}} + \sum_{i>j} \frac{e^2}{4\pi\epsilon_0 r_{ij}} \]

Exact: Evaluate matrix elements of \( \hat{H} \) in a basis, then solve the eigenvalue problem to get wave functions and energy levels.

Approximate: Hartree-Fock, Perturbation theory, coupled-cluster etc (discussed more later)

Hartree-Fock is a *mean field theory*. Only treats correlation between electrons in an average way.
Alternative approach: Density functional theory

- Hohenberg-Kohn theorems:
  - Everything about the ground state properties of a many electron system is determined by the electron density, which is a function of only 3 special coordinates $\rho(x, y, z)$
  - The ground state $\rho(x, y, z)$ is the one which minimizes the energy – and the energy is a functional of the density $E[\rho(x, y, z)]$
  - But what does this $\rho(x, y, z)$ look like and how do we find it?
Kohn-Sham DFT

- Instead of solving the problem for the fully interacting system, solve the non-interactive Schrodinger equation for a fake system of non-interacting particles that have the same density as the interacting system.

- These non-interacting particles feel an effective potential

\[ \left( -\frac{\hbar^2}{2m} \nabla^2 + v_{\text{eff}}(r) \right) \phi_i(r) = \epsilon_i \phi_i(r) \quad \text{and} \quad \rho(r) = \sum_{i=1}^{N} |\phi_i(r)|^2 \]

- (note, ignoring spin for now)
Kohn-Sham potential

- \( E[\rho] = T_s[\rho] + \int dr \, v_{\text{ext}}(r)\rho(r) + E_H[\rho] + E_{\text{xc}}[\rho] \)
- Kinetic energy, external potential and interaction with nuclei, coulomb repulsion, and “exchange correlation” (everything else!)
- \( T_s[\rho] = \sum_{i=1}^{N} \int dr \, \phi_i^*(r) \left( -\frac{\hbar^2}{2m} \nabla^2 \right) \phi_i(r) \)
- \( E_H[\rho] = \frac{e^2}{2} \int dr \int dr' \frac{\rho(r)\rho(r')}{|r-r'|} \)
- From this, functional derivative: \( v_{\text{eff}}(r) = v_{\text{ext}}(r) + e^2 \int \frac{\rho(r')}{|r-r'|} dr' + \frac{\delta E_{\text{xc}}[\rho]}{\delta \rho(r)} \)
- \( v_{\text{xc}}(r) = \frac{\delta E_{\text{xc}}[\rho]}{\delta \rho(r)} \) is the exchange correlation potential
- Hartree term and exchange correlation functional depend on density, so this has to be done iteratively/self-consistently
Approximations to the XC functional

- Local density approximation (LDA) only depends on density at a point
- Gradient corrected (GGA): depends on density and gradient of density
- Meta-GGA: density, gradient, second derivative
- Hybrid DFT: fraction of exchange from Hartree fock calculation. Includes B3LYP, PBE
SMILES strings

- Simplified molecular-input line-entry system
- Non hydrogens are represented by atomic symbol, hydrogens are assumed based on valence: "C"=methane, “N”=Ammonia, “O”= water
- Bonds are represented as ‘-’, ‘=’, and ‘#’, with single bonds implied
  - “C”=methane, “CC”=ethane, “CCC”=propane etc.
- Read some more about it in exercises
RDKit

- Very powerful and extensive python package for dealing with molecules and cheminformatics
- Has interfaces for dealing with molecules, reactions, etc
- Also good for drawing lewis structures in 2d, and also can get 3d structures and minimize them with molecular mechanics tools
- Can use this too feed structures into quantum calculations
- I also installed package psikit which is a wrapper for combining rdkit and psi4
**Pandas**

- Framework in python for making tables of data
- Very important for data science
- Good interface with RDKit and jupyter
- You will make a pandas “dataframe” with your results
**Today**

1. Pull updates on comp-lab-class github page
   
   https://github.com/hockyg/comp-lab-class/blob/main/Week11/Assignment.md

2. Do some rdkit practice

3. Build some dataframes with rdkit

4. Fill in some properties with psi4/psikit