

# Week 3

???

## 3.1 Computational Chemistry

- 9/17:
- Lecture 3 recap.
    - Huckel theory: A fast way to draw the MOs of conjugated  $\pi$ -systems.
      - If the conjugated  $\pi$ -system in question is cyclic, use a Frost circle.
    - Aromaticity.
      - Huckel's definition:  $4n + 2$ .
      - Möbius's definition:  $4n$ .
      - Leads to properties like stabilization, quadrupoles, and ring current.
    - Cyclopropane:  $sp^2$ -like banana bonds (the only thing we need to remember from that discussion).
    - Wavefunctions: Solutions to the Schrödinger equation.
  - Today: Computational chemistry (an overview).
    - Computational chemistry is typically an entire class!
  - Lecture outline.
    - Methods of computational chemistry.
    - Molecular mechanics.
    - Semi-empirical methods.
    - Ab initio methods.
    - Hartree-Fock.
    - Density functional theory (DFT).
    - Best practices for calculations.
    - Properties that are especially easy (or hard) to calculate.
  - Why do we do computational chemistry?
    - If we could fully solve the Schrödinger equation, we could know the properties of all of our electrons!
    - However, the Schrödinger equation can only be fully solved (practically) for the simplest systems.
      - For now, at least: People are working on this.
    - As such, we *approximate* solutions instead.

- **Computational chemistry:** The science of approximating solutions to the Schrödinger equation.
  - Computational chemistry can be broken up into two general strategies (**ab initio** and **empirical** methods) and one in-between strategy called **semi-empirical** methods.
- **Ab initio** (methods): Make well-defined approximations to the Schrödinger equation, and then solve the approximations mathematically. *Etymology* from Latin “from first principles.”
  - Essentially, make your math simpler.
- **Semi-empirical** (methods): Replace complicated parts of the Schrödinger equation with experimentally derived parameters, such as bond lengths, vibrational frequencies, and more that we can get from spectroscopy.
  - Essentially, shortcut the hardest parts of solving with experimentally derived features.
- **Empirical** (methods): Approximate molecules with force fields that are experimentally derived, and adjust with further experimental parameters.
  - Essentially, start with reality and derive computational things from that.
- We now look at some commonly derived methods. The following list is sorted from methods with high **accuracy** and low **speed** to methods with low accuracy and high speed.
  - Methods at the high end of accuracy and the low end of speed (ab initio).
    - **Coupled cluster.**
    - **Perturbation theory.**
    - **Density functional theory.**
    - **Hartree-Fock.**
  - Methods in the middle (semi-empirical).
    - **Semi-empirical methods.**
  - Methods at the high end of speed and the low end of accuracy (empirical).
    - **Molecular mechanics.**
- **Speed:** Ease of calculations.
- **Accuracy:** Careful and diligent.
- **Coupled cluster:** Useful for approximately 10 **heavy atoms**. *Also known as CC.*
- **Density functional theory:** Useful for approximately 80 heavy atoms, though we can use more (it just gets slower). *Also known as DFT.*
- **Hartree-Fock.** *Also known as HF.*
- **Molecular mechanics:** Useful for hundreds of heavy atoms. *Also known as MM.*
- **Heavy atom:** Any atom that's not hydrogen.
- In this course, we'll discuss further the bottom four methods in the above list of six.
- Molecular mechanics (MM).
  - Atoms are treated as balls and springs (this is a classical analogy and thus much easier to simulate).
  - We use force fields to describe electrons.
    - These force fields are derived from experimental data, i.e., choose a force field that gives us the bonds we calculate from XRD or the vibrations we see in IR.
  - Very fast; often considered “quick and dirty.”

- Gives us a general picture of what we're thinking about.
- Common application: Very large and flexible systems.
  - Think proteins, polymers, etc.
  - Things that have a lot of degrees of freedom.
  - Very useful for chembio, polymer chemistry, etc.
- Subset application: **Molecular dynamics** (MD).
  - Simulating movement; uses MM as a basis.
- If you're going to use this method, know that it is (in general) only appropriate for approximating the ground states of molecules (not their transition states).
  - However, MM can be a good starting point for higher-level calculations (i.e., more accurate methods).
  - In Orgo, it's mainly used for first approximations to be refined later (and for heavier stuff).
  - All the same, it is a super useful tool with tons of applications, and its simplicity should not lead us to discount it.
- Running MM.
  - If we have a PC, try clicking the MM2 button in Chem3D (which is part of our ChemDraw package).
  - This may not work on Macs; figure this out!!
  - PerkinElmer (who developed ChemDraw) initially developed their stuff for Macs; Masha's not quite sure where they dropped the ball.
- Semi-empirical quantum mechanical (SQM) methods.
  - Use empirical parameters to simplify *ab initio* calculations.
    - Tries to deliver the best of both worlds (speed and accuracy).
  - We can add corrections for missing phenomena and underestimated features.
  - Theoreticians (developers) will draw the line on accuracy somewhere, and then organic chemists will say, "this model fails here."
    - Once that feedback gets into the literature, theoreticians redefine their line.
    - They might need to account for *d*-orbitals, London dispersion forces (LDFs), flexibility, solvent, or more.
      - Methods of accounting for solvent effects are continuously being optimized.
    - It's important to be on top of the literature here, since things are always getting better!
  - Modern implementations (these are getting fast enough to be usable and really good!).
    - Density function based tight binding (DFTB): Approximate DFT.
    - eXtended Tight Binding (xTB)
      - Developed primarily by the Grimme lab.
      - Basically just adding more parameters.
  - LDFs are becoming increasingly important for selective catalysis, so there's a lot of work to approximate them.
    - Catalysis is not about partial positive and negative charges so much as it is about electrons flopping around to achieve incredible selectivities in next-gen catalysts.
  - Very fast (seconds) and pretty accurate. Increasingly used, especially for ML and data science.
    - Nowadays, if you want to do ML, you need these hundreds of experimental data points.
- Ab initio methods.
  - Background theories (neither is technically true, but it is helpful for speed).

- **Born-Oppenheimer approximation.**
- **Independent electron theory.**
- **Born-Oppenheimer approximation:** Nuclei are way bigger than electrons (have over 1000 times more mass), so they are basically fixed in space relative to the electrons.
  - This means that you can treat the nuclei separately; you can use one approach for the nuclei and an entirely different approach for the electrons.
- **Independent electron theory:** Electron movements are not correlated to each other; all electrons whiz around independently.
  - Making this approximation will cause some issues.
- Hartree-Fock (HF).
  - Treat electrons as a delocalized cloud with independent electron movement.
    - Remember the plum pudding model of the atom? This is not that dissimilar from that.
  - This approximation ignores Coulombic interactions (like LDFs).
    - This becomes very problematic for transition states.
  - HF methods are largely historical today.
    - There are applications where they're still used today, but not in Orgo and not without an understanding of their shortcomings.
- We can run any and all of these computations throughout grad school as MIT students, and we should! They're in our toolbelt now, and we should try them out!!
- Density functional theory (DFT).
  - Instead of calculating wavefunctions, we're going to calculate electron density.
    - We're going to do this using **functionals**.
    - We can include functionals for things like Coulombic interactions, etc.<sup>[1]</sup>
  - This is a good workhorse method in organic chemistry.
    - DFT is appropriate for reaction coordinate mapping, transition states, etc.
    - We'll often work with collaborators that can tailor a model to our needs.
  - There are many specific functionals and basis sets.
    - You have to choose the functional (choose what to include), and then choose the basis sets (how much detail do I need for this calculation, e.g., treating polarization, charge, unpaired electrons more accurately).
    - It is best to find a basis set and functional appropriate for our context.
  - Basis sets don't describe all types of elements.
    - Some describe elements 1-30, others do 1-86.
    - Don't be that person who has to redo their entire calculation because they forgot that tin is one of their reagents!
    - We often use **split basis sets** (esp. for transition metals), i.e., certain atoms (i.e., metals) get more functionals.
      - Carbon, hydrogen, and oxygen (CHO) don't need the craziest level of theory to approximate, but that palladium center will!
    - Think about what level of theory you need for each atom.

---

<sup>1</sup>Maybe what I can be known for in research is custom building computational tools for specific organic problems, and turning that into a workflow that people do. Maybe that's what ML already is.

- **Functional:** A function of functions.<sup>[2]</sup> Also known as **higher-order function**.
- Best practices for running calculations for our own things.
  - This part of the lecture is *critical*; it tells us what we need to know to use computational chemistry.
    - If we want to learn the theory for all of these things, we should read a textbook or take Heather Kulik's class.
  - Use the appropriate level of theory for your needs and capabilities.
  - Questions to ask yourself to assess your needs and capabilities.
    - Do you have a supercomputer? How much time on the supercomputer do you have?
    - When do you need this result by? Is your PI breathing down your neck?
    - What am I trying to model?
    - Is this a thought experiment or something serious?
  - Additional things to consider wrt your needs and capabilities.
    - Consider speed vs. accuracy.
      - You can always start at a lower level theory and then ramp it up if you need more accuracy. This is a great general approach.
    - Consider size and flexibility (no HF on proteins, or MM on methane).
    - Consider “weirdness”: If you've got something that's all inverted and Möbius like, you're gonna need something more tailor-made.
    - Find a *reliable* literature precedent for a similar system.
      - If you want to model a cationic cyclization, use a precedent paper's level of theory.
      - How do I model an iridium catalyst? Find an iridium catalyst paper and go from that!
    - Know how your level of theory works.
      - Does it account for polarizability? Charge? Solvent? *d*-orbitals?
      - It is our responsibility as an experimentalist to know this if we're going to publish it; our PI probably won't be as deep into the nitty-gritty as us.
  - Don't blindly trust calculations.
    - Calculations always give you an answer (unless they fail or don't converge). However, just because you get a number doesn't mean that that number is accurate!
    - Benchmark your calculation with experiments whenever we can. Examples: X-ray structure, ratio of products (we can back-calculate from temperature the activation energy barrier, and from the transition states what ratio of products we expect).
    - Redo a couple of calculations at a higher level of theory to see if you get the same answer.
    - Chris Cramer (a founding father of computational chemistry): “There is no particular virtue to the speed at which a wrong answer can be obtained.”
  - Example of doing calculations wrong: Doing an S<sub>N</sub>1 reaction without solvent. These reactions are so solvent-dependent, and there's no gas-phase cation that will replicate this solution-phase reaction.
- What's easy to calculate?
  - Spectra: IR, Raman, NMR.
    - Masha likes to predict the NMR spectra of wacky intermediates.
    - ChemDraw does this for free.
      - The default solvent is THF; make sure you change it to CDCl<sub>3</sub>!
    - MNova's function is better; it's ML-based, but it also costs money to run?? I think Masha has this wrong for MIT students.

---

<sup>2</sup>This is the computer science definition; it is largely unrelated to the mathematical definition that is equivalent to linear forms and duals.

- Geometries, conformers, and ground state structures.
  - “Geometry optimization” or “energy minimization” is very common.
  - Draw a 3D structure, give it to our program, move atoms, calculate  $E$ , repeat (let the program perturb the atom’s positions a bit) until we reach a *local* minimum.
  - This is what we’ll do on the problem set.
  - If we want to get the *global* minimum, we have to look for lower energy structures (manually, automatically, or a combination of both).
  - We often start at a low level of theory and then refine. Start with a search of the chemical space to find some stable conformers, and then pop that into DFT.
- Frequencies, well-defined transition states, and **single point calculations**.
  - Important because transition states are saddle points on the potential energy surface with 1 imaginary frequency corresponding to the bond-making or -breaking event.
  - If you have a structure that you think is a ground state, you have to prove this.
- **Single point calculation:** Calculating the energy of a structure without any other atoms around.
- Note: Nucleophiles typically come in at a  $120^\circ$  angle (the **Bürgi-Dunitz angle**), because that’s where it’s easiest to donate into the  $\pi^*$ -lobe.
- What’s “hard” to calculate?
  - Caveat: Do your research!!
    - Many applications require specialized approaches.
    - There’s an army of computational chemists who are trying to develop niche methods for our little problem; find them, connect with them, collaborate with them, etc.
    - It is our responsibility to know what part of a certain experiment is difficult.
      - Example: In photophysics, you need to know the limitations of certain parts of our model.
      - The system will not say, “I’m bad at predicting excited states;” you have to know that.
  - Things that require more finessing to study.
    - Open-shell species, e.g., radicals.
    - Transition metals: Heather researches how to model TMs with SQM, etc. This is really important and really hard.
    - “Unusual structures,” e.g., gas-phase plasma nonsense.
  - Thermochemistry.
    - E.g., thermodynamical parameters, calorimetry, etc.
    - There are specific packages that work for this, if we need to look into them.
    - Masha doesn’t know anything about any of this, but recommends that we can learn them!!
  - Solvent effects, LDFs, etc.
    - For these topics, the methods are getting better all the time (which is code for, “the programs don’t work great yet”).
    - **Implicit solvation** vs. **explicit solvation**.
- **Implicit solvation:** Treat the solvent as a continuous medium.
- **Explicit solvation:** Draw solvent molecules and add them to the calculation.
  - If you think the solvent is stabilizing the transition state in an  $S_N1$ , you need to draw a little THF donating its lone pair to the carbocation.
  - This gets complicated in proteins, where we need to identify how many waters we need to draw to accurately represent water; recent papers suggest that effects matter up to 44  $H_2O$  molecules away!