

# Molecular Modeling

Steve Rotenberg  
CSE291: Physics Simulation  
UCSD  
Spring 2019

# Molecular Modeling

- Molecular modeling is a fairly mature field having evolved over more than fifty years
- Modern techniques are able to model complex interactions between mega-molecules such as proteins
- They can also predict properties of novel molecules and macroscopic properties of novel materials
- They are also used as a tool in modern medical engineering, genetics, and pharmaceutical design

# Molecular Modeling

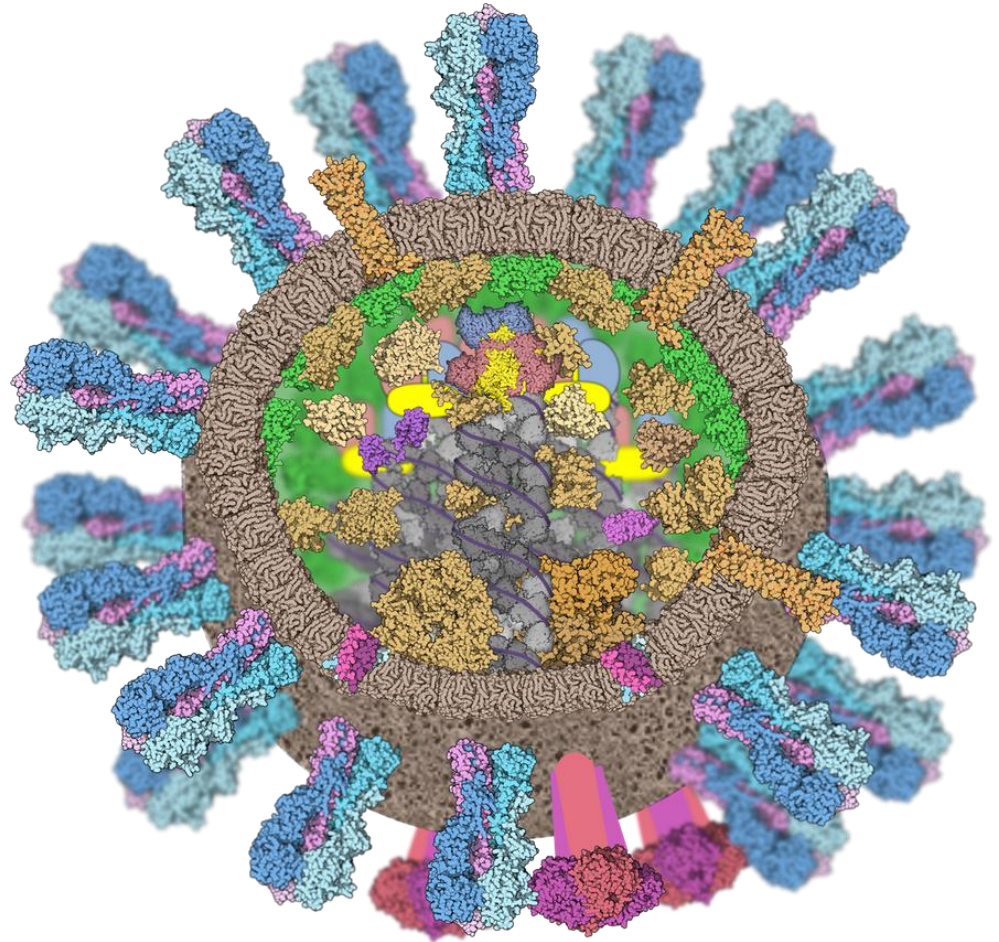
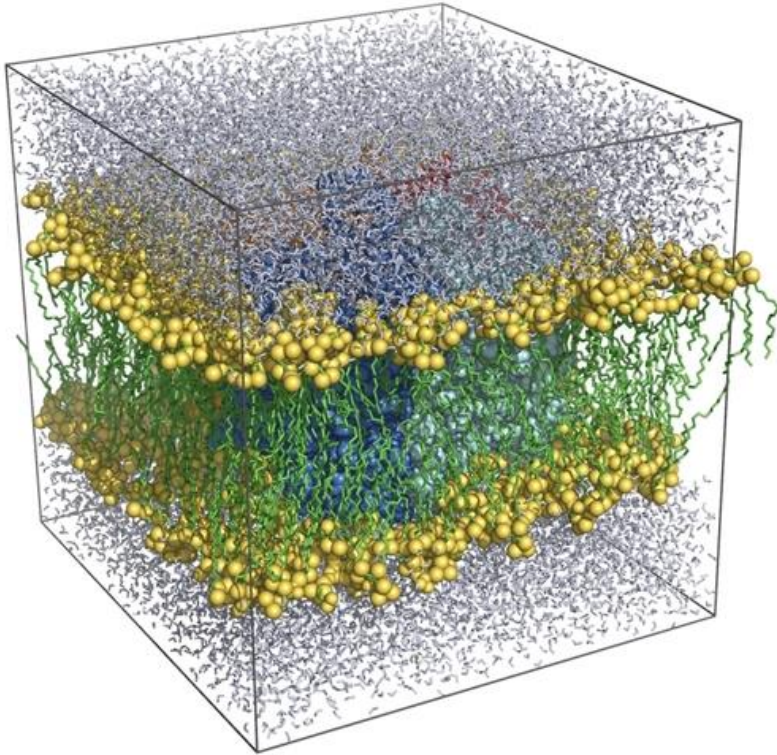
- Broadly speaking, molecular modeling can be divided into two main approaches:
  - Molecular Dynamics (MD): each atom is treated as a Newtonian particle and interatomic forces are based on various models
  - Quantum Modeling: electron wave equations are computed based on Schrödinger equation and used to predict properties of molecules
- The MD method is used for larger scale simulations of molecules interacting over time, and typically don't involve any real chemical interactions or bond forming/breaking
- Quantum methods are very slow and complex but much more general and able to predict a wide variety of chemical properties
- The two methods can also be hybridized- for example, a drug interacting with a protein might be primarily simulated with MD but use quantum mechanics to model the up close interactions at the reaction site

# Molecular Dynamics

# Molecular Dynamics

- *Molecular dynamics* (MD) treats individual atoms as Newtonian point masses and interactions between atoms as various force models
- At a minimum, the forces include
  - Covalent bonds
  - Angular bonds
  - Torsional bonds
  - Van-der-Waals forces
  - Coulomb forces
- In general, MD does not model any chemical interactions, so no bonds are formed or broken
- Also, gravity is usually ignored, as it is very weak at the time scales involved

# Molecular Dynamics



*Image: Ed Hutchinson/University of Glasgow*

# Atoms

- Individual atoms are treated as point masses
- An atom has an integer atomic number, which is the number of protons, and serves to identify the type (hydrogen, carbon, gold, etc.)
- It also has an integer number of neutrons, which can vary for different isotopes of the atom
- It will also have an atomic mass, based on the number of protons and neutrons
- The masses and isotope information can be found in a good periodic table or chemistry reference

# Force Computation

- As with other simulation methods, we just sum up the forces on the particles
- In some cases, forces are defined in terms of the total potential energy function

$$U(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n)$$

- Where the forces on each particle are defined as the negative spatial gradient of the potential function

$$\mathbf{f}_i = -\frac{\partial U}{\partial \mathbf{r}_i}$$



# Covalent Bonds

- A covalent bond between two atoms (whether a single, double, or triple bond) is modeled essentially as a spring between the two atoms
- At a minimum, it would have a rest length (average bond length) and a spring constant
- More complex models can use nonlinear functions for the force-distance relationship
- The values of the constants in these will vary based on the type of the two atoms being bonded (H-H will have different constants from H-C, etc.) and these are usually just specified in lookup tables

# Angular Bonds

- The term 'angular bond' refers to a force that acts on the angle between two bonded atoms
- For example, consider a water molecule  $\text{H}_2\text{O}$  where the average angle between the two hydrogen atoms is 104.5 degrees
- We use an angular bond to enforce this

# Torsional Bonds

- A 'torsional bond' is a hypothetical force that acts against twisting along an axis connecting two atoms
- It operates on the axis in the middle of a chain of 4 molecules

# Bond Geometry

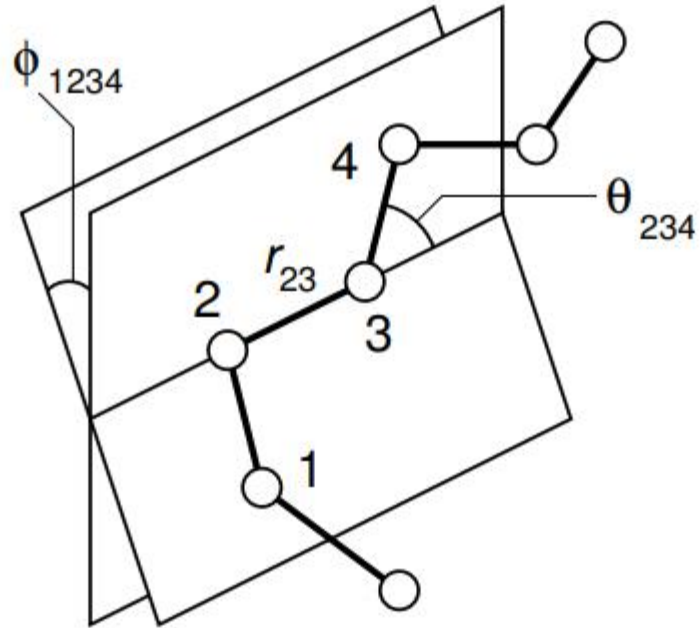


Figure 3. Geometry of a simple chain molecule, illustrating the definition of interatomic distance  $r_{23}$ , bend angle  $\theta_{234}$ , and torsion angle  $\phi_{1234}$ .

*Image: "Introduction to Molecular Dynamics Simulation", Michael P. Allen*

# Bond Force Model

- A basic bond force potential model that includes covalent, angular, and torsional components is:

$$U_{bond} = \frac{1}{2} \sum_{bonds} k_{r_{ij}} (r_{ij} - r_{eq})^2$$
$$+ \frac{1}{2} \sum_{\substack{bend \\ angles}} k_{\theta_{ijk}} (\theta_{ijk} - \theta_{eq})^2$$
$$+ \frac{1}{2} \sum_{\substack{torsion \\ angles}} \sum_m k_{\varphi_m_{ijkl}} (1 + \cos(m\varphi_{ijkl} - \gamma_m))$$

- This requires lots of constants that have to be fit to experimental data

# Coulomb Force

- The *Coulomb force* is the force due to the electrostatic attraction/repulsion between two atoms

$$\mathbf{f}_{coulomb} = \frac{1}{4\pi\epsilon_0} \frac{q_1 q_2}{r^2} \mathbf{e}$$

- Where  $q_1$  and  $q_2$  are the electrostatic charges,  $r$  is the distance between the atoms, and  $\mathbf{e}$  is a unit vector from atom 1 to 2
- Most atoms in molecules are roughly neutral, but some will have a slight charge due to ionization or irregular distribution of electrons
- The Coulomb force acts between every pair of atoms, but it usually ignored for bonded pairs

# Van der Waals Forces

- The van der Waals force is an interaction between pairs of nearby atoms
- When atoms are nearby, the force causes a slight attraction
- When they get very close, the force switches to a strong repulsion and prevents the atoms from getting too close
- The full force is quite complex, and is often simplified with a model along the lines of:

$$\mathbf{f}_{vdw} = \left( \frac{C_1}{r^{13}} - \frac{C_2}{r^7} \right) \mathbf{e}$$

- Where  $C_1$  and  $C_2$  can be fit to experimental data

# Periodic Domains

- Most molecular dynamics interactions of interest happen within gasses, liquids, or solids
- To account for the properties of large scale fluids and solids, most molecular dynamics simulations take place in a *periodic domain*
- Typically, the simulation is limited to a rectangular box that is meant to repeat effectively infinitely in each direction
- When an individual atom passes the +x boundary, for example, it wraps around to the -x boundary
- Molecular interaction forces have to take the tiling space into consideration. For example, if an H<sub>2</sub> molecule is sitting on the x boundary, one of the hydrogen atoms will be on the far right of the domain and the other will be on the far left. The bond analysis code has to normalize them into the same space



# Force Octrees

- Covalent bond forces are explicitly identified at the start of the simulation and operate between specific pairs (or triplets/quads) of atoms
- Coulomb and van der Waals forces however, operate between all pairs of atoms which leads to  $n^2$  computational time
- The octree methods we used to optimize galactic simulations can also be adapted to optimize these forces and can also be adapted to the tiling space

# Integration

- It is common to use the variation of forward Euler that we previously looked at, using the new velocity to advance the time instead of the previous velocity

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \mathbf{a}_i \Delta t$$
$$\mathbf{r}_{i+1} = \mathbf{r}_i + \mathbf{v}_{i+1} \Delta t$$

- This goes by many different names in literature including semi-implicit Euler, Euler-Cromer, symplectic Euler, Størmer-Verlet, and leapfrog, depending on the context
- It is simple and reasonably good at preserving energy in this type of problem
- The time step is usually around 1 fs ( $1 \times 10^{-15}$  s)

# Temperature Control

- The integration method will guarantee conservation of momentum, but will only approximately conserve energy
- In order to enforce conservation of energy, we can use an artificial ‘thermostat’, which allows us to set the average kinetic energy (temperature) to any value we want by just scaling all particle velocities slightly up or down until we reach the temperature we want
- This temperature control also allows us to simulate the effects of external heating or cooling by raising or lowering the thermostat

# Water Solutions

- Many molecular interactions of interest take place in water
- Water is quite complex, as the hydrogen bonding has a significant effect on the behavior
- There are two common options for modeling the surrounding water solution
  - Explicitly model it with a bunch of water molecules (slower, more accurate)
  - Implicit model that applies the average effects of the water solution to boundaries of other molecules (faster, less accurate)

# Force Field Definitions

- There are many standard force field definitions that contain sets of force constants for numerous types of chemical bonds
- Some of the main ones are AMBER, CHARMM, GAFF, MMFF, and UFF, and each of these has multiple variants tuned for specific molecules and specific applications
- There are also many commercial packages that implement one or more of these force fields along with various other tools and user interfaces

# Quantum Modeling

# Quantum Mechanics

- If we want to do serious molecular modeling, we can simulate molecules at the quantum level
- Quantum simulations compute the wave functions of the electrons in a molecule, allowing for computation of more fundamental properties
- For example, one can do things like:
  - Predict the shape of a molecule from its atomic formula (including distances between atoms and bond angles)
  - Compute the electrostatic charge distribution of a molecule
  - Compute the energy levels of base and excited states
  - Estimate the color (absorption spectrum) and other optical scattering properties
  - Predict chemical interactions
  - Compute constants used in MD simulations (rest lengths, bond stiffness, angles...)

# Finite Dimensional Real Vector Spaces

- We've done lots of work with real valued finite dimensional vector spaces, such as the space of 3D vectors
- A vector in  $n$ -dimensional space is written as:

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = [a_1 \quad a_2 \quad \cdots \quad a_n]^T$$

- And the dot product of vectors  $\mathbf{a}$  and  $\mathbf{b}$  is:

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^n a_i b_i = [a_1 \quad a_2 \quad \cdots \quad a_n] \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \mathbf{a}^T \mathbf{b}$$



# Complex Vector Spaces

- If we extend this to allow the components of the vector to be complex numbers, we have a *complex vector space*
- If we also extend this to allow  $n$  to be infinity, we have an *infinite dimensional vector space*
- We can introduce the *Dirac notation* (also called *bra-ket notation*) for this
- A vector in this notation is written as:

$$|a\rangle = \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} a_{1r} + ia_{1i} \\ a_{2r} + ia_{2i} \\ \vdots \\ a_{nr} + ia_{ni} \end{bmatrix}$$

- Where each value has a real and imaginary component and  $n$  is allowed to be infinity

# Infinite Dimensional Spaces

- Infinite dimensional spaces are actually not that weird
- Consider some function  $f(x)$
- We can sample it at 3 points and represent an approximation of the function as a 3D vector
- We could increase the number of samples to 100 and sample a wider range of the function at a higher density and represent it with a 100D vector
- We could just keep going and sample the function at an infinite number of places and represent it as an infinite dimensional vector
- Instead of representing the individual dimensions with a positive integer 1, 2, 3, ...,  $\infty$ , we instead represent them with a real number  $x$   $[-\infty, \infty]$ . Either way, we have an infinite number of values to work with
- An infinite dimensional vector is just another way of thinking of a function. In fact, we don't have to limit it to a 1D function. Any  $n$ -dimensional function could be represented as an  $\infty$ D vector

$$|a\rangle \equiv a(x)$$

# Vector Adjoint

- If we have a complex number  $c = c_r + ic_i$ , we define the *complex conjugate*  $c^*$  as:

$$c^* = c_r - ic_i$$

- We can define the *vector conjugate* as the vector of conjugates, and the *adjoint*  $\dagger$  as the transpose of the vector conjugate:

$$\langle a | = |a\rangle^\dagger = [a_1^* \quad a_2^* \quad \cdots \quad a_n^*]$$

# Dot Product

- In Dirac notation, the dot product equivalent is defined as:

$$\langle a|b\rangle = \langle a| \cdot |b\rangle = \mathbf{a}^\dagger \mathbf{b} = \sum_{i=1}^n a_i^* b_i$$

- If we assume  $n$  is infinite, we can replace the summation with an integral:

$$\langle a|b\rangle = \int a^*(x)b(x)dx$$

# Unit Vectors and Normalized Functions

- Just as we can compute the squared magnitude of a vector as  $\mathbf{a} \cdot \mathbf{a}$ , we can compute the squared magnitude of infinite dimensional vector as:

$$|a|^2 = \langle a|a \rangle = \int a^*(x)a(x)dx$$

- If  $a(x)$  is normalized, then  $\langle a|a \rangle = 1$
- This gives us a method for normalizing a  $\infty$ D vector:

$$\text{normalize}(a) = \frac{|a\rangle}{\sqrt{\langle a|a \rangle}}$$

# Orthonormal Bases

- We can construct an orthonormal basis in 3D space by defining 3 vectors  $\mathbf{e}_1$ ,  $\mathbf{e}_2$ , and  $\mathbf{e}_3$  such that:

$$\mathbf{e}_i \cdot \mathbf{e}_i = 1$$

$$\mathbf{e}_i \cdot \mathbf{e}_j = 0, \text{ for } i \neq j$$

- We can also write this as

$$\mathbf{e}_i \cdot \mathbf{e}_j = \delta_{ij}$$

- Where  $\delta_{ij}$  is the *Kronecker delta*:

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j \end{cases}$$

# Orthonormal Bases

- We can also have an orthonormal basis in infinite dimensional complex vector spaces
- We can define an set of basis vectors  $\mathbf{e}_1 \dots \mathbf{e}_n$  where  $n$  can be a finite or infinite

$$\langle e_i | e_j \rangle = \delta_{ij}$$

# Operations

- We can take a vector and multiply it by a matrix to produce a new vector

$$\mathbf{b} = \mathbf{M}\mathbf{a}$$

- In a sense, we are performing an operation on the vector, and in particular, it is a linear operation
- We can generalize this into infinite dimensional space and define an operator  $\mathcal{O}$  on a vector producing a new vector as:

$$|b\rangle = \mathcal{O}|a\rangle$$



# Eigenfunctions

- With finite dimensional vectors, we have the familiar eigenvalue equation:

$$\mathbf{M}\mathbf{x} = \lambda\mathbf{x}$$

- Given matrix  $\mathbf{M}$ , we wish to find orthonormal eigenvectors  $\mathbf{x}_i$  that when transformed by  $\mathbf{M}$ , result in a vector lining up with the original vector, but scaled by eigenvalue  $\lambda_i$
- With infinite dimensional vectors, we have a similar equation:

$$\mathcal{O}|a\rangle = k|a\rangle$$

- Given an operator  $\mathcal{O}$ , we wish to find the orthonormal eigenfunctions  $|a_i\rangle$  as well as their associated eigenvalues  $k_i$

# Schrödinger Equation

- The time-independent form of the Schrödinger equation is:

$$\mathcal{H}|\psi\rangle = \mathcal{E}|\psi\rangle$$

- Where  $\psi$  is a wavefunction,  $\mathcal{H}$  is the Hamiltonian operator, and  $\mathcal{E}$  is the (scalar) energy
- It is an eigenfunction equation that describes the properties of quantum wavefunctions

# Wavefunctions

- The function  $|\psi\rangle$  refers to the spatial wavefunction of the particles concerned
- The wavefunction is a single infinite dimensional vector, but for a particle, it can be thought of as a density function in 3D space
- Technically, it's a probability function, but we can really just think of the particle as being smeared out by this function
- If the wavefunction represents a single electron, the negative electric charge is not concentrated at a single point, but is distributed spatially by the wavefunction
- If two particles are interacting, we can use a single 6D wavefunction, where the first 3 dimensions are used to represent the first particle and the next 3 are for the second particle
- In this way, we can have  $n$  particles interacting all represented by a single wavefunction

# Born-Oppenheimer

- The mass of a single proton is 1837 times the mass of an electron (neutrons are about the same as protons)
- All of the protons and neutrons of an atom pack into a tiny space in the nucleus
- For these reasons, we can treat the entire nucleus of each atom as a 0-dimensional point with a charge equal to the number of protons
- Quantum molecular modeling treats the electrons as wave functions, but simplifies the protons and neutrons together as point charges
- This is known as the Born-Oppenheimer approximation
- This leads to a small quantifiable error on the order of 1% in most cases

# Hydrogen Hamiltonian

- In the  $\mathcal{H}|\psi\rangle = \mathcal{E}|\psi\rangle$  we have the Hamiltonian operator  $\mathcal{H}$ , the wavefunction  $|\psi\rangle$ , and the scalar energy  $\mathcal{E}$
- If we are modeling hydrogen and using the Born-Oppenheimer approximation for the proton, then the wavefunction only needs to describe the single electron
- The Hamiltonian defines the inherent behavior of the electron combined with the attraction to the proton
- For hydrogen, we have a single electron interacting with a single proton (fixed at the origin) resulting in the Hamiltonian:

$$\mathcal{H} = -\frac{1}{2}\nabla^2 - \frac{1}{r}$$

- Note: In atomic units, most of the constants (speed of light, electric constant, elementary charge) are 1.0, so this equation looks clean

# Hydrogen Atom

$$\mathcal{H}|\psi\rangle = \mathcal{E}|\psi\rangle$$

$$\mathcal{H} = -\frac{1}{2}\nabla^2 - \frac{1}{r}$$

- If we write the wavefunction  $|\psi\rangle$  as a function of in 3D space  $\psi(\mathbf{x})$ , we can re-write this as:

$$-\frac{1}{2}\nabla^2\psi(\mathbf{x}) - \frac{\psi(\mathbf{x})}{r} = \mathcal{E}\psi(\mathbf{x})$$

- Which is a messy spatial differential equation that relates a function  $\psi(\mathbf{x})$  with its Laplacian  $\nabla^2\psi(\mathbf{x})$
- Remarkably, there is an exact solution, but it is not simple

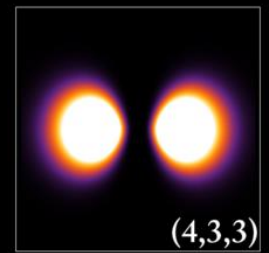
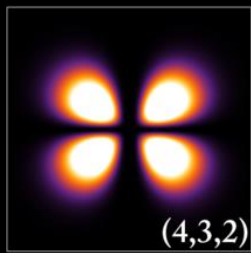
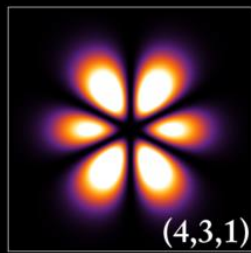
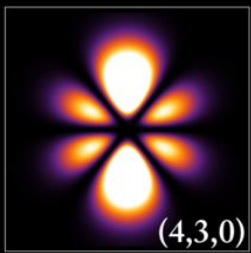
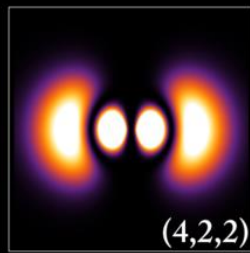
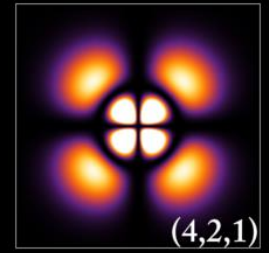
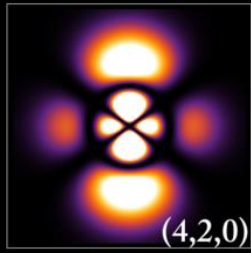
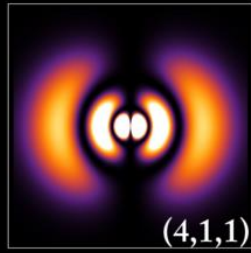
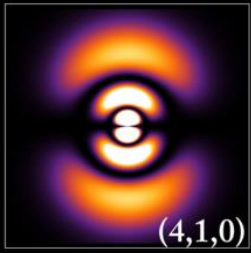
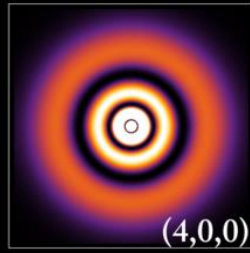
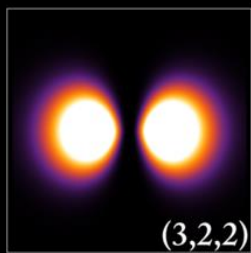
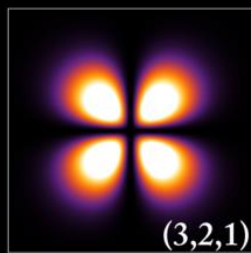
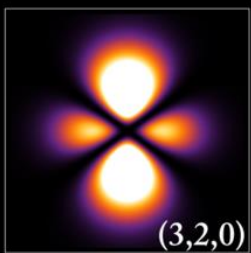
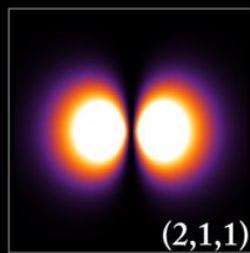
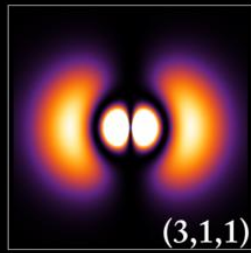
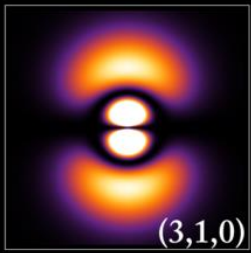
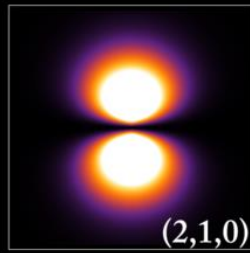
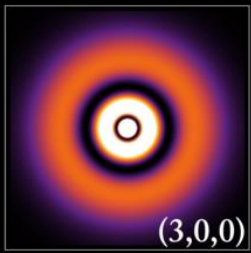
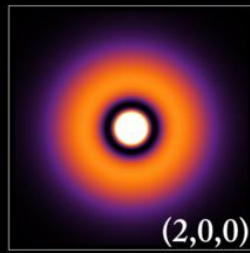
# Hydrogen Atom

- If we use the time independent form of the Schrödinger equation and we treat the proton at the center as a fixed point, we can calculate an exact solution to the eigenfunction equation
- This gives us a set of electron wavefunctions corresponding to quantum energy levels (eigenvalues)
- For hydrogen, we can compute these analytically with a combination of factorials, Laguerre polynomials, and Legendre polynomials
- These lead to the 1s, 2s, 2p, 3s, 3p, 3d... orbitals that are typically taught in chemistry classes

# Hydrogen Wave Function

Probability density plots.

$$\psi_{nlm}(r, \vartheta, \varphi) = \sqrt{\left(\frac{2}{na_0}\right)^3 \frac{(n-l-1)!}{2n[(n+l)!]}} e^{-\rho/2} \rho^l L_{n-l-1}^{2l+1}(\rho) \cdot Y_{lm}(\vartheta, \varphi)$$





# Hartree-Fock Method

# Molecular Hamiltonians

- For multiple electrons interacting around multiple atomic nuclei:

$$\mathcal{H} = - \sum_{i=1}^n \frac{1}{2} \nabla_i^2 - \sum_{i=1}^n \sum_{A=1}^m \frac{Z_A}{r_{iA}} + \sum_{i=1}^n \sum_{j>i}^n \frac{1}{r_{ij}}$$

- The first term is the Laplacian term, which is fundamental to the physics of particles
- The second term is the Coulomb repulsion between electron  $i$  and atomic nucleus  $A$  with charge  $Z_A$  and at distance  $r_{iA}$
- The third term is the Coulomb repulsion between electron  $i$  and electron  $j$

# Pauli Exclusion Principle

- In larger atoms or molecules, more than one electron is involved
- The *Pauli exclusion principle* prevents multiple particles from having the same state means that the electrons will occupy different energy levels
- In the rest state of a molecule with  $n$  electrons, the lowest  $n$  levels will be occupied
- As they are eigenfunctions, the electrons will be orthogonal to each other, i.e.,  $\langle a|b \rangle = 0$

# Hartree-Fock Equation

- The *Hartree-Fock equation* is a complex integral-differential equation that can be used to compute approximations to solutions of the molecular Schrödinger equation
- The quantum modeling method that uses this equation is known as the *Hartree-Fock method* (HF)
- The method dates back to the early 1930's
- It can be used to calculate properties of very basic molecules (like H<sub>2</sub>) on paper, but the calculations grow quite substantial for anything more complex, requiring computers

# Hartree-Fock Method

- The only input required for the HF method is a set of atomic nuclei, with given integer charges and 3D vector locations
- The masses are assumed infinite relative to the electrons, so aren't used in the calculations (Born-Oppenheimer approximation)
- The output is a set of approximate electron wavefunctions as well as the energy levels associated with each electron

# Hartree-Fock Method

- A single run of the HF method takes atomic nuclei as input and computes electrons as output
- This information can be used to compute various other data such as ionization energies, electrostatic potentials, electron densities, bond strengths, absorption spectra, and more
- By iteratively running the HF method, one can compute energy minimizations used to predict geometric properties of molecules, such as bond angles and lengths

# Hartree-Fock Method

- As the calculations are performed in atomic units, most of the relevant constants are 1.0
- The method does not require any additional constants or tuning parameters
- In other words, it can hypothetically compute the properties of the entire periodic table and of any molecule completely from scratch without any magic numbers
- In practice, however, it has its limits due to the required computational performance and the limitations of the way it approximates wavefunctions with a set of basis functions

# Basis Functions

- The HF method makes certain assumptions about the electrons in order to make the problem tractable
- One main assumption it makes is that the shape of the molecular electron orbitals will be similar to the shape of atomic orbitals
- It further assumes that an electron orbital will resemble a linear combination of the basic hydrogen-like orbitals
- To simplify calculations, these hydrogen-like orbitals are themselves approximated with a summation of Gaussian functions
- These component functions that resemble hydrogen orbitals are known as *basis functions*



# Basis Functions

- The idea behind basis functions is that we can represent a 3D field as a weighted summation of these basis functions

$$f(\mathbf{x}) \approx \sum_{i=1}^n w_i B_i(\mathbf{x})$$

- Where  $f(\mathbf{x})$  is the function we wish to approximate,  $w_i$  is a set of  $n$  weights, and  $B_i(\mathbf{x})$  is a set of  $n$  basis functions
- We pre-select the basis functions based on how we expect the function to look
- We then wish to solve the set of  $w_i$  weights that provide the best fit to the function

# Basis Functions

- If we are modeling a water molecule, for example, we know that we will need 10 electrons around 3 atomic nuclei
- Each electron wavefunction may have properties of the lowest 10 energy states of hydrogen, which includes s-type and p-type orbitals
- We therefore choose a set of basis functions that include s-type and p-type shapes to work with

# Hartree-Fock Method

- The HF method iteratively computes a set of weights for the basis functions
- Ultimately, it will find the best set of weights to represent the full set of electrons in the molecule
- The quality of the solution is therefore highly dependent on the selection of the basis functions and their ability to accurately model the correct solution
- One can use a larger number of basis functions in hopes to improve the accuracy of the solution
- Unfortunately, the performance of the HF method is proportional to  $O(n^4)$  where  $n$  is the total number of basis functions in the molecule

# Basis Functions vs. Grids, etc.

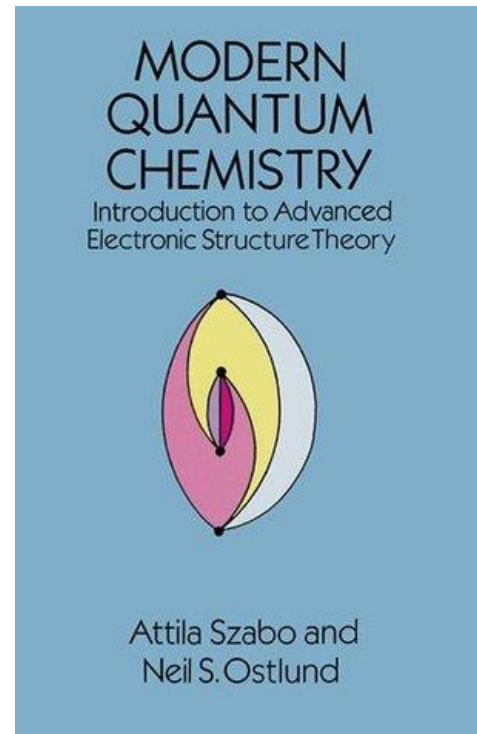
- We've seen many methods for representing fields in this course, such as grids, irregular meshes, hierarchical grids, and particles
- Basis functions are yet another method of doing this
- The grid/particle methods are great for representing arbitrary functions, but they require lots of variables
- Basis functions are much more limited and require choosing basis functions that can express the possible fields
- Their advantage is that they require much fewer variables than grid/particle methods
- Actually, all methods can be thought of as basis function methods, since particles are just small round basis functions, and grid cells are small square ones!

# Density Functional Theory

- *Density functional theory* (DFT) refers to a alternative quantum modeling method from the Hartree-Fock method
- It has advantages and disadvantages, but one particular advantage is that it handles tiling domains well
- This makes it very effective for modeling crystals and novel materials
- DFT and HF are two of the main methods used in quantum level molecular modeling today

# Reference

- “Modern Quantum Chemistry”, Szabo, Ostlund, 1982 (revised 1989)
- Excellent book on applied quantum modeling



# Multi-Scale Modeling

- We can use different physical models at different scales to perform *multi-scale simulations*
- For example, we can design new materials at the molecular scale
- We can use quantum modeling to study the molecules and compute bond and other properties that can be fed into a molecular dynamics simulation
- We can then model larger scale structures of how the molecules form a continuous material. Once we can do this, we can compute elasticity properties of the material by applying a strain to the MD simulation and computing and tabling the resulting stress. We can also model plasticity and other properties at the MD level to compute material parameters to use in a finite element simulation
- At this stage, we can do finite element simulations of macroscopic objects modeled with our new material
- This type of multi-scale modeling can also apply to visual properties