

# CDL Quantum 2020 Cohort Project

July 22, 2020

## 1 Introduction

The 2020 Cohort Project is an online collaboration bringing together founders in a series of open-source challenges. During the four-week bootcamp, founders will form teams to compete and collaborate on a set of challenges on a general topic or theme relevant for the Quantum Stream. Four separate challenges will be issued; one per week. New teams will be formed by the CDL Quantum team weekly.

Challenges consist of scientific, computational, and business tasks, devised to be tackled in diverse teams of founders with complementary skill sets. Each year the theme of the Cohort Project will change. In 2020 it is *quantum chemistry*. Begin your reading with the suggested review.<sup>1</sup>

## 2 Quantum Chemistry

The field of quantum chemistry is a multi-disciplinary effort spanning physics, chemistry, computer science, software engineering, and high-performance computing, all working together to assist in the design and discovery of new molecules, materials, and drugs. It is a *native quantum* field, meaning that the underlying foundation of its technology is quantum mechanics. It offers a rich opportunity to develop science and technology based companies around classical software, machine learning, quantum-inspired solutions, and quantum computing.

One of the central problems of theoretical chemistry is to compute the lowest energy state of a molecular Hamiltonian. The behavior of protons and electrons in atoms is governed by the Schrodinger equation  $i\hbar\frac{\partial}{\partial t}|\Psi\rangle = \hat{H}|\Psi\rangle$ , where  $\hat{H}$  is the Hamiltonian, and  $\Psi$  is the quantum state or *wavefunction*. Quite generally, this equation cannot be solved exactly for more than a few atoms. For large molecules this problem is intractable even on today's largest supercomputers.

The standard approach to computational chemistry begins with the Born-Oppenheimer approximation, which assumes that nuclei are fixed classical point charges. These charges create the potential that determines the electronic Hamiltonian, and therefore the electronic wavefunction. The low-energy electronic states is where all the action is; solving this given a molecular Hamiltonian  $\hat{H}$  is called the *electronic structure* problem. Except for simple few-body atoms and molecules, the solution to electronic structure problems can require significant computational resources. In this cohort project we will largely (but not exclusively) consider diatomic molecules composed of two bonded atoms, where reasonable computational calculations can be done.

---

<sup>1</sup><https://arxiv.org/abs/1812.09976>

## 3 Weekly Challenges

Each week your team will be presented with a set of problems in the form of scientific tasks and challenges. In order to complete these, you will *fork* this repository to a GitHub account managed by your team. After completing your work on the project at the end of the week, you will issue a *pull request* back to the upstream repository. CDL will choose one team's work to merge into the main upstream repository, where it will become part of the legacy of the 2020 Boot Camp. For more information see the GitHub documentation.<sup>2</sup>

### 3.1 Week 1: Machine Learning

Significant speedups in simulating quantum mechanical systems, such as electrons in molecules, can in principle be achieved with quantum computers. As stated in the previous section, determining the behavior of protons and electrons in atoms comprises of computing the lowest energy state of the entity that governs their behavior: the Hamiltonian,  $\hat{H}$ . For most of the problems in this Project, you will be asked to find a wavefunction given a Hamiltonian. However for Week 1 we will assume that a quantum computer is doing that for us. In order for this to happen, one must first translate the problem of dealing with electrons in an atom to the language of quantum hardware, which deals with quantum bits (qubits), not electrons. There exists several mathematical frameworks wherein one can view Hamiltonians that describe electrons as Hamiltonians that describe qubits<sup>3</sup>. This Hamiltonian can then be encoded in a quantum device that (hopefully) finds the lowest energy state. Next week, you will learn in detail about how molecular energies can be calculated in quantum hardware using the Variational Quantum Eigensolver (VQE).

This week we will approach the electronic structure problem from a data-driven perspective. You will be given the *output* of a quantum computer - the individual qubit measurements of a molecular qubit Hamiltonian that was prepared on a (real or simulated) quantum computer. From this data, you will be asked to reconstruct the quantum state  $\Psi$  in a stochastic neural network (a restricted Boltzmann machine) using unsupervised machine learning techniques. This reconstructed state will be a faithful representation of the quantum wavefunction from which you can calculate physical quantities – for example the electronic energy  $E = \langle H \rangle$ .

#### 3.1.1 Neural networks as quantum wavefunctions

Machine learning (ML) tasks are often broken down as *supervised* or *unsupervised*. A general goal of unsupervised learning is to learn probability distributions and make new predictions from limited input in the form of data.<sup>4</sup> Consider for instance the task of learning the probability distribution underlying a biased coin (i.e. a coin that is not 50/50 heads or tails). A friend of yours flips this coin many times and sends you the results as a data set. To answer the question “what is the probability of getting heads or tails?”, you could just build a *frequency distribution* and say that

$$P_{\text{heads}} \approx \frac{\# \text{ of heads}}{\# \text{ of heads} + \# \text{ of tails}}$$

<sup>2</sup> <https://help.github.com/en/github/collaborating-with-issues-and-pull-requests/about-forks>

<sup>3</sup>For those interested: The electronic structure Hamiltonian can be expressed in terms of fermionic operators. After a Bravyi-Kitaev transformation, the same Hamiltonian now describes qubits. See e.g. quant-ph/0108146 and quant-ph/0003137.

<sup>4</sup>A question to ponder: what is the relationship between a probability distribution  $P$  and a quantum wavefunction  $\Psi$ ?

and  $P_{\text{tails}} = 1 - P_{\text{heads}}$ . However, in some cases (what cases?) frequency distributions are insufficient. Then, ML algorithms called *generative models* can be used. Generative models comprise of a bunch of tunable parameters that, given the algorithm architecture, describe a general probability distribution. The model is trained by tuning its parameters (call them  $\lambda$ ) in such a way that its probability distribution closely matches the underlying one from the data your friend gives you. For Project 1, we will use a restricted Boltzmann machine (RBM) as the generative model, where the parameters are weights and biases.<sup>5</sup> Let's now break down what the training looks like.

### 3.1.2 Training

ML algorithms generally associate how “well” they are doing in learning through a cost function; for example the *Negative Log-Likelihood*.<sup>6</sup>

$$NLL = - \sum_v p_{\text{true}}(v) \ln(p_{\text{ML alg.}}(v)) \quad (1)$$

In the case where a quantum computer provides data, what is  $p_{\text{true}}$  and  $p_{\text{ML alg.}}$ ? How do these relate to  $\Psi$ ? ML algorithms want to minimize the cost function: find the set of parameters in the model such that the cost function is smallest. The way in which ML algorithms do this is via *gradient decent*<sup>7</sup>: update parameters,  $\lambda$ , incrementally at every training step with the following rule.

$$\lambda \leftarrow \lambda - \eta \nabla_{\lambda} NLL \quad (2)$$

- $\eta$ : The *learning rate*. It is just a real number that scales the gradient.
- $\nabla_{\lambda}$ : Symbol to denote the gradient operator with respect to all of the parameters in the model,  $\lambda$ .

Generally, the gradient of the cost function may not be analytic (i.e. it may not have a “nice” form that we can write down and we have to resort to other differentiation methods). For the RBM however, the gradients of the cost function are analytic. We are simply required to sample our machine learning algorithm (ask it to generate new instances of data) in order to calculate  $\nabla_{\lambda} NLL$ . In summary, for our purposes, training the RBM boils down to the following.

1. Generate data from the RBM
2. Calculate  $\nabla_{\lambda} NLL$
3. Update the RBM parameters:  $\lambda \leftarrow \lambda - \eta \nabla_{\lambda} NLL$

Start with reading the documentation for Project 1.<sup>8</sup> You will explore using the RBM to learn the wavefunctions of the diatomic  $H_2$  molecule and a chain of Rydberg atoms from the provided data (Task 1 and Task 2). Once you've mastered these, you're encouraged to continue with one of the *Challenges* provided (or better yet, make up your own)!

<sup>5</sup>For more reading see <https://arxiv.org/abs/1905.04312>

<sup>6</sup>This is related to the *Kullback Leibler Divergence*:  $\sum_v p_{\text{true}}(v) \ln\left(\frac{p_{\text{true}}(v)}{p_{\text{ML alg.}}(v)}\right)$

<sup>7</sup>Really, there are many different algorithms for minimizing the cost function. Pytorch has a list of different optimization methods that you can find here: <https://pytorch.org/docs/stable/optim.html>

<sup>8</sup><https://github.com/CDL-Quantum/>

## 3.2 Week 2: Variational Quantum Eigensolver: Constructing Potential Energy Surfaces for Small Molecules

Solving the time-independent Schrodinger equation,

$$\hat{H}|\psi\rangle = E|\psi\rangle \quad (3)$$

is, arguably, the central goal of physics and chemistry. Essentially, a model for something in nature is captured in the Hamiltonian,  $\hat{H}$ , and we wish to extract properties,  $E$  and  $|\psi\rangle$ , from this model.<sup>9</sup> More often than not, solving this via brute force is intractable for modern hardware. We therefore settle for numerical techniques to approximately/exactly solve this problem. One very common way to solve the Schrodinger equation is via the *variational method*.

Let's suppose that we do not know how to specifically solve the Schrodinger equation, but we know what the solution  $|\psi\rangle$  should “look” like (i.e. its functional form). So, we come up with a trial solution  $|\psi_{\text{trial}}(\Theta)\rangle$  that has the desired features we want from a solution, but also can be tuned via changing parameters  $\Theta$  that define it. How do we change these parameters? The variational method boils down to an optimization problem. The entity that we wish to optimize is the trial energy,

$$E_{\text{trial}}(\Theta) = \langle \psi_{\text{trial}}(\Theta) | \hat{H} | \psi_{\text{trial}}(\Theta) \rangle. \quad (4)$$

We give the task of finding parameters  $\Theta$  such that  $E_{\text{trial}}(\Theta)$  is minimized to a machine and we're left with the optimal trial solution.

However, in the process of describing the variational method we've glanced over a very important problem: finding  $|\psi_{\text{trial}}(\Theta)\rangle$  to begin with. This is where this week's topic of the Variational Quantum Eigensolver comes in handy.

### 3.2.1 The Variational Quantum Eigensolver [1]

The Variational Quantum Eigensolver (VQE) provides two answers to our goal of solving the Schrodinger equation using the variational method: a versatile/expressive trial solution and computational efficiency. Briefly, our trial solution is embedded in a quantum circuit through parameterized *gates*.<sup>10</sup> Given an output from the circuit, we can then ask the quantum device to efficiently calculate  $E_{\text{trial}}(\Theta)$ . To optimize the circuit parameters, we then rely on a classical algorithm to adjust the parameterized gates in the circuit. See Figure 3.2.1. Continue with reading the documentation for Project 2.<sup>11</sup>

<sup>9</sup>This is a linear algebra problem; we wish to find the eigenspectrum of  $\hat{H}$ :  $E$  and  $|\psi\rangle$

<sup>10</sup>Unitary operations that alter the input to the quantum circuit.

<sup>11</sup><https://github.com/CDL-Quantum/>

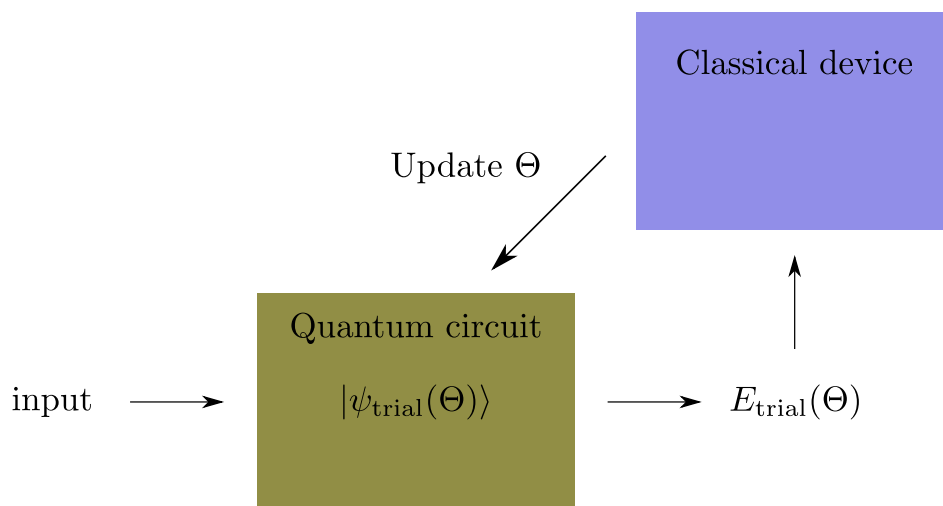


Figure 1: A flowchart of VQE.

### 3.3 Week 3: Franck Condon Factors

You’ve probably wondered how scientists in labs are able to determine extremely small quantities like the distance between atoms that are chemically bonded. Spectroscopy is the field of study that has provided techniques to measure such quantities with ease. Not only is its importance in academic settings unprecedented, industry relies on spectroscopic theory and techniques for countless applications. This week, you’ll learn:

1. introductory physics and chemistry behind spectroscopy
2. what a Franck Condon Factor is
3. how to calculate Franck Condon Factors for simple molecules

For further reading, you may consult the following references among many others: Ref. [2, 3, 4].

#### 3.3.1 What is spectroscopy?

Spectroscopy is the study of how light and matter (atoms and molecules) interact. Light can be absorbed by matter (*absorption*) or matter can emit light (*emission*). However, it turns out that atoms and molecules can only emit / absorb certain wavelengths of light and not all wavelengths. The wavelengths of light that atoms and molecules can emit / absorb is dictated by the *energy levels* of a molecule. Think of a molecule’s energy levels as a set of stairs, where each stair represents one energy level and the vertical distance separating stairs is the difference in energy between stairs. For a molecule to absorb or emit light, the light must have *exactly* the amount of energy between two stairs. If stairs # 1 and # 2 have energy  $E_1$  and  $E_2$  respectively, then the light’s energy,  $E_{\text{light}}$ , must satisfy

$$E_2 - E_1 = E_{\text{light}}. \quad (5)$$

Therefore, a molecule can only absorb or emit light that has a specific energy, and therefore wavelength since the energy of light is inversely proportional to its wavelength ( $\lambda$ ).

$$E_{\text{light}} \propto \frac{1}{\lambda} \quad (6)$$

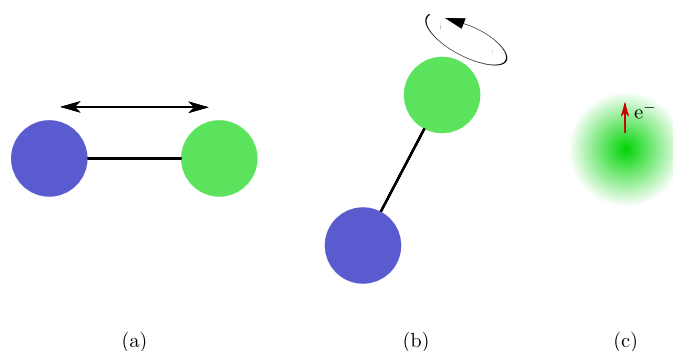


Figure 2: (a) Molecular vibration, (b) rotation, and (c) exciting an electron in an atom / molecule.

It is from the pattern of light that a molecule can emit / absorb – its *spectrum* – that we can determine all sorts of properties (e.g. bond strengths, positions of atoms in molecules, etc.)

### 3.3.2 Molecular energy levels

So, the energy levels of molecules give us the key to its structural properties. What are these energy levels? Energy in a molecule can be stored in a few ways: vibration, rotation and the energy of electrons ( $e^-$ ) surrounding the molecule (see Fig. 2). For now, let's only worry about harnessing vibrational energy and energy from exciting electrons within a molecule.

The chemical bond is what forces atoms in a molecule to stay close enough together. If we were to try and take two atoms that are chemically bonded and rip them apart, we would be met with incredible resistance from the retaliation of the bond. On the flip side, there is a repulsive force between atoms that are bonded that ensures that the atoms don't get too close to each other. This balance between the repulsive and attractive forces determines the distance that separates atoms in a chemical bond (called *bond length*). If we were to plot the potential energy stored in a molecule,  $E_{bond}$ , as a function of the two atoms' separation,  $r$ , it would look something like the blue or red curve in Fig. 3. It's at the very minimum point on the curves in Fig. 3 where the attractive and repulsive forces are in equilibrium with each other.

Vibrations affect the distance between the atoms,  $r$ , in the molecule. It turns out that the laws of quantum mechanics only allow for *discrete* vibrational energies, and that the lowest vibrational energy that can be obtained (dubbed the “ground” vibrational state) is actually slightly above the minimum point on the blue/red curve. The allowed vibrational energy levels of the molecule are the horizontal lines within the blue curve in Fig. 3.

We can also think about exciting the molecule's electrons. If we were to do this, this can drastically change the interplay between the attractive and repulsive forces. Therefore,  $E_{bond}$  as a function of  $r$  changes significantly. The red and blue curves in Fig. 3 are meant to represent the molecule in an excited electronic state and in its ground (lowest energy) electronic state, respectively. Note that the excited electronic state of the molecule also vibrates and therefore has its own discrete vibrational energy levels.

### 3.3.3 Wavefunctions

We need knowledge of one more thing before we can accomplish our goal of calculating Franck Condon Factors: wavefunctions. The wavefunction of a system, often denoted by the Greek letter  $\psi$ , is essentially the key to being able to calculate anything and everything about said system. In our

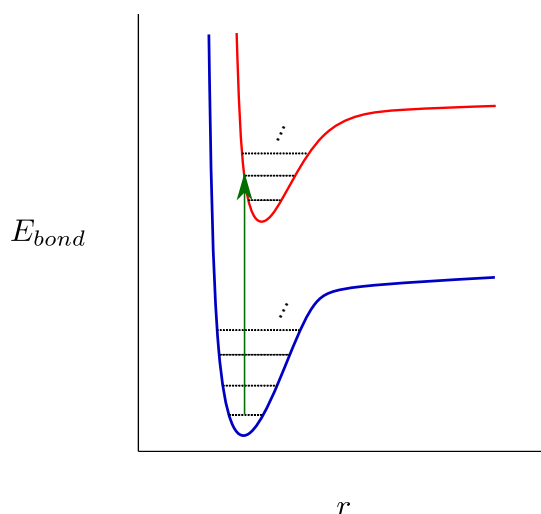


Figure 3: The general form of a molecule’s potential energy curve as a function of bond length. The blue curve represents a molecule in its ground electronic state and the red curve represents the same molecule in an excited electronic state.

spectroscopy language, the wavefunction of a molecule that we are interested in can be extremely complicated to fully determine. However, for our purposes we can “separate” the wavefunction of our molecule into parts describing its vibration ( $v$ ), rotation ( $R$ ) and electrons ( $e$ ).

$$\psi_{\text{molecule}} = \psi_e \psi_R \psi_v \quad (7)$$

### 3.3.4 Franck Condon Factors

Armed with this conceptual knowledge of molecular vibrational energy levels and wavefunctions, we can now learn about Franck Condon Factors (FCFs). Picture the following scenario. We have a molecule in its ground state (both vibrationally and electronically), and we excite it electronically (i.e. move from the blue curve to the red curve in Fig. 3). To a very good approximation, it turns out that electronic transitions are so fast that the distance separating atoms in a molecule does not have time to adjust (this is the Franck Condon Principle). Diagrammatically, we are looking at the vertical green line in Fig. 3 (it is vertical because the atoms in the molecule stays at the same distance apart during the transition). This specific “vertical” transition from the ground electronic and vibrational state to the excited electronic state and its second vibrational state will have a *transition intensity* that is related to the vibrational part of the wavefunctions describing the system before and after the transition. More specifically, the transition intensity is proportional to the square *overlap* (think of a vector dot product) between the two vibrational wavefunctions.

$$\text{Transition intensity} \propto \text{overlap}(\psi_{v,\text{before}}, \psi_{v,\text{after}})^2 \quad (8)$$

The square of this overlap is precisely the Franck Condon Factor!

$$\text{FCF} = \text{overlap}(\psi_{v,\text{before}}, \psi_{v,\text{after}})^2 \quad (9)$$

### 3.3.5 A simplified model for vibrations of diatomic molecules (2-atom molecules)

We’ve now developed the idea of a Franck Condon Factor (FCF). How do we formally calculate these quantities? As evident in Eq. (9), we require knowledge of the *vibrational wavefunction* before



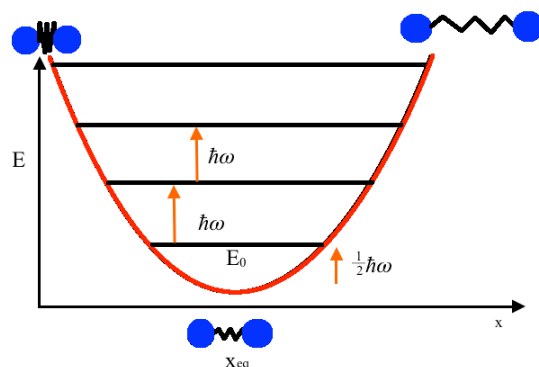


Figure 4: A representation of a diatomic and its vibrational energy levels in the harmonic oscillator approximation. The properties of this diatomic molecule are its equilibrium geometry,  $x_{eq}$ , its frequency,  $\omega$ , and its reduced mass,  $\mu$ . The ground state energy ( $E_0$ ) is  $\frac{1}{2}\hbar\omega$  above the potential minimum and vibrational energy differences between adjacent levels is  $\hbar\omega$ .

and after the transition. One of the simplest ways to represent vibrations of diatomic molecules is to consider that they are connected by a spring. This “spring system” is usually referred to in the scientific community as a *harmonic oscillator*. Let’s now look into this description of molecular vibrations.

In section 3.3.2, you learned that vibrational energy levels are discrete. Fig. 4 shows how the vibrational energy levels are spaced in the harmonic oscillator picture. If a molecule is in its ground vibrational state, it will “stretch” and “contract” about its natural (or equilibrium) bond length. As it gains energy and moves into excited vibrational states, it can stretch and contract further and further from its equilibrium bond length. The frequency of the vibration – how many times in one second that the molecule’s spring fully extends, then fully contracts, and comes back again – is directly related to how strong the bond is between the atoms (i.e. how “stiff” the spring is).

The harmonic oscillator is an extremely simplified model, since it ignores all repulsive effects of the 2 atoms within the molecule (the steep wall at small  $r$  denoted by a more realistic potential energy curve in Fig. 3). On top of this, the harmonic oscillator potential has no dissociation (or bond-breaking) as is captured by the flat portion at large  $r$  in the potential energy curve in Fig. 3. In our harmonic oscillator picture, this means that at higher excited states the atoms will continue to stretch further and further without the “spring snapping.”

However, the harmonic oscillator has a useful feature in that the energy levels are equally spaced and the vibrational wavefunction (what we need to calculate FCFs) takes a “nice” analytic form, allowing for simple calculations. The  $n^{th}$  state of the harmonic oscillator has the following vibrational wavefunction.

$$\psi_{v,n}(x) = \frac{1}{\sqrt{2^n n!}} \left( \frac{\mu\omega}{\pi\hbar} \right) \cdot \exp\left(-\frac{\mu\omega(x - x_{eq})^2}{2\hbar}\right) \cdot H_n\left(\sqrt{\frac{\mu\omega}{\hbar}}(x - x_{eq})\right) \quad n = 0, 1, 2, \dots \quad (10)$$

$\mu$  is the reduced mass of the molecule,  $\omega$  is its fundamental vibrational frequency,  $x$  is the bond length,  $x_{eq}$  is the equilibrium bond length,  $n$  is the vibrational state of the system, and  $H_n$  is called a *Hermite polynomial*.<sup>12</sup>

With this information, we can begin actually calculating FCFs! Start with reading the documentation for Project 3.<sup>13</sup> You will explore techniques to calculate FCFs. Once you’ve mastered these

<sup>12</sup>Check this Wikipedia article out to read more on Hermite polynomials: [wikipedia.org](https://en.wikipedia.org/wiki/Hermite_polynomials).

<sup>13</sup><https://github.com/CDL-Quantum/>



techniques and have finished the tasks, you're encouraged to continue with one of the *Challenges* provided (or better yet, make up your own)!

### 3.4 Week 4: Ising Annealer

Numerical methods are often developed with little consideration for application in the sciences. Early Monte Carlo methods were no exception. Initially, Monte Carlo methods were designed to approximate very difficult integrals (i.e. functions that behave unideally and very high dimensional integrals). We encourage you to visit the Wikipedia article<sup>14</sup> on MC integration methods. There are some interesting examples to work through! For now, let's focus on how we will be employing MC methods to solve your week 4 tasks.

#### 3.4.1 Monte Carlo Simulations (Metropolis-Hastings)

In statistical physics at finite temperature, the partition function,

$$Z = \sum_{\sigma} e^{-\beta E(\sigma)}, \quad (11)$$

is what we wish to calculate in order to fully determine the equilibrium properties of a system of interest. Here,  $\sigma$  describes one of many possible microscopic configurations that a system can be in,  $E(\sigma)$  is the corresponding energy of the configuration  $\sigma$ , and  $\beta \propto \frac{1}{T}$  where  $T$  is the temperature of the system. It turns out the the probability of the system being in a configuration  $\sigma$  is<sup>15</sup>

$$p(\sigma) = \frac{e^{-\beta E(\sigma)}}{Z}. \quad (12)$$

Our goal will be to sample configurations  $\sigma$  from this probability distribution. However, this can rarely be done directly because the partition function, Eq. (11), requires resources exponential in system size to compute. We are once again met with the *curse of dimensionality*!

Since we will once again be working in the language of spin-1/2 particles as in Projects 1 and 2, we may think of configurations  $\sigma$  as being spin configurations. For a system of  $N$  particles, a spin configuration is simply  $N$  bits (0's and 1's). Suppose that we start with a spin configuration that may or may not be a very probable spin configuration that the system could be in. Our goal will be to modify said configuration in such a way that after several calculated modifications we will have essentially sampled the probability distribution given by Eq. (12). How do we modify the initial configuration and how do we know that the modification we made is a good one?

Systems lie in states that minimize their energy. Therefore, if we propose modifications to the initial configuration such that the new configuration has a lower energy, we should always accept that modification. On the flip side, if the new configuration we propose has a higher energy than the previous one we can accept it probabilistically. This is, essentially, the Metropolis-Hastings algorithm. Here it is in a little more detail.

1. Choose a random spin configuration  $\sigma$  to start.
2. Randomly choose a spin in the configuration and flip it (i.e.  $0 \rightarrow 1$  or  $1 \rightarrow 0$ ). Call this new configuration  $\sigma'$ .

<sup>14</sup>[wikipedia.org/wiki/Monte\\_Carlo\\_integration](http://wikipedia.org/wiki/Monte_Carlo_integration)

<sup>15</sup>We are disregarding any degeneracies here.

3. Accept this new configuration with probability

$$P_{\text{accept}} = \min \left( 1, \frac{e^{-\beta E(\sigma')}}{e^{-\beta E(\sigma)}} \right) \quad (13)$$

4. If you accepted the update, let  $\sigma = \sigma'$ . Otherwise, discard  $\sigma'$ .
5. Repeat step 2.

The number of times that we repeat step 2 in the above list is typically called the *number of MC steps*.

Of course, we've intentionally kept this introduction to Monte Carlo methods very brief and we've left out many important details. However, this should be enough to get you started. If you want to learn more check out Ref. [5].

### 3.4.2 Annealing

MC simulations that use the single-flip MH algorithm as outlined previously work well in a lot of cases. However, sometimes these simulations fail at lower and lower temperatures. Consider for moment that we performed a MC simulation at a higher temperature  $T_1$  and we've obtained good samples from the system of interest at temperature  $T_1$ . If we were to look at the system at a slightly lower temperature  $T_1 - \varepsilon$ , where  $\varepsilon \ll 1$ , we should expect that the samples we obtained from the simulation at  $T_1$  would be very similarly distributed to samples obtained from a good simulation at  $T_1 - \varepsilon$ . So, if we use the samples from the  $T_1$  simulation as our starting configurations for a simulation at  $T_1 - \varepsilon$  we should have a much easier time with completing a simulation at  $T_1 - \varepsilon$ .

Suppose now that our end goal is to have samples from a system at a temperature  $T_2$  and that performing a MC simulation directly at  $T_2$  is difficult. However, a MC simulation at  $T_1$  is relatively easy. A procedure to get to an efficient simulation at  $T_2$  is if we do exactly what was outlined above but many, many more times until we reach  $T_2$ . This is called an *annealing procedure*.

With this information, you can start reading the documentation for Project 4.<sup>16</sup> You will get a chance to work with a MC simulation that uses the MH algorithm and execute an annealing procedure to simulate a low-temperature system. Once you've mastered the tasks, continue on with one of the Challenges provided or make up your own!

---

<sup>16</sup><https://github.com/CDL-Quantum/>

## References

- [1] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Aln Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1), Jul 2014.
- [2] Nicolás Quesada. Franck-Condon factors by counting perfect matchings of graphs with loops. *J. Chem. Phys.*, 150(16):164113, April 2019.
- [3] John C. Wright and Theresa Julia Zielinski. Franck-Condon Factors and Their Use in Undergraduate Quantum Mechanics. *J. Chem. Educ.*, 76(10):1367, October 1999.
- [4] U. Fantz and D. Wunderlich. Franck-Condon factors, transition probabilities, and radiative lifetimes for hydrogen molecules and their isotopomers. *Atomic Data and Nuclear Data Tables*, 92(6):853–973, November 2006.
- [5] M. E. J. Newman and G.T. Barkema. *Monte Carlo Methods in Statistical Physics*. Clarendon Press, 1998.