

# Modeling Hydrogen Isotope Separation in Three Well-Characterized MOFs

by

Esme Rubinstein

Submitted to the Department of Physics and Astronomy

OBERLIN COLLEGE

May 2021

© Oberlin College 2021. All rights reserved.



# Modeling Hydrogen Isotope Separation in Three Well-Characterized MOFs

by

Esme Rubinstein

## Abstract

This thesis explores the possibility of using differences in the quantum zero-point energy (ZPE) to separate out rare hydrogen isotopes. In particular, we use a theoretical approach to model the difference in energy between  $H_2$  and  $D_2$  in three representative metal organic frameworks (MOFs). For each MOF, we used parametrized interaction models from the literature to generate the potential energy of the adsorbed isotope and solve the Schrodinger equation. The results are comparable to experimental values indicating that a theoretical approach might be useful in pre-screening candidate MOFs for hydrogen isotope separation.

Thesis Supervisor: Stephen FitzGerald

Title: Professor, Chair of Department of Physics and Astronomy



## Acknowledgments

Thank you to Professor FitzGerald for advising and mentoring me throughout this project. I am also grateful for the help and support of my family: Zack Rubinstein, Alice Julier, and Zoe Rubinstein. Thank you to my roommates and friends for keeping me motivated and also helping me relax. Finally, thank you to the Oberlin Physics Department for making this possible.



# Contents

<b>1</b>	<b>Introduction</b>	<b>13</b>
<b>2</b>	<b>MOFs Explored</b>	<b>15</b>
2.1	MOF-5 . . . . .	16
2.2	MOF-74 . . . . .	18
2.3	CuMFU-4l . . . . .	19
<b>3</b>	<b>Theory</b>	<b>21</b>
3.1	Zero Point Energy . . . . .	21
3.2	Potential Energy Models . . . . .	22
3.2.1	MOF-5 and MOF-74 Potential Energy Model . . . . .	22
3.2.2	MOF-5 and MOF-74 Potential Energy Model Assumptions . . . . .	23
3.2.3	Cu-MFU-4l Potential Energy Model . . . . .	24
3.3	Solving the Schrodinger Equation . . . . .	25
<b>4</b>	<b>Results and Analysis</b>	<b>27</b>
4.1	MOF-5 . . . . .	27
4.2	MOF-74 . . . . .	30
4.3	CuMFU-4l . . . . .	30
4.3.1	B3LYP-D2/6-31++G**(6-31G*) Model . . . . .	33
4.4	Difference in ZPE Comparison . . . . .	35
<b>5</b>	<b>Conclusion and Future Work</b>	<b>37</b>
5.1	Feynmann-Hibbs Equation . . . . .	37

5.2	Concluding Statements . . . . .	39
<b>6</b>	<b>Appendix</b>	<b>41</b>
6.1	Code Used to Calculate Energy States . . . . .	41
6.1.1	point.py . . . . .	41
6.1.2	atom.py . . . . .	43
6.1.3	utilities.py . . . . .	45
6.1.4	realisticPotentialModel.py . . . . .	46
6.1.5	realisticPotentialGraph.py . . . . .	52
6.1.6	CuUtilities.py . . . . .	58
6.1.7	SE.py . . . . .	60
6.2	generateWavefunction.py . . . . .	62



# List of Figures

2-1	Depicted are stacked unit cells for MOF-5. Blue is $Zn$ , red is $O$ , dark grey is $C$ , and light grey is $H$ . The orange and yellow spheres depict the alternating pore sizes in the MOF. Taken from [15]. . . . .	16
2-2	MOF-5 primary binding site. Numbers refer to the atom labels in Table 2.1. Grey is $Zn$ , red is $O$ , blue is $C$ , and white is $H$ . Image taken from [2]. . . . .	17
2-3	$2 \times 2 \times 2$ unit cell system of Mg-MOF-74. Cyan is $C$ , white is $H$ , red is $O$ , and grey is $Mg$ . Taken from [13]. . . . .	18
2-4	Mg-MOF-74 primary binding site. Numbers refer to the atom labels in Table 2.2. Cyan is $C$ , white is $H$ , red is $O$ , and grey is $Mg$ . Taken from [13]. . . . .	18
2-5	Portion of CuMFU-4l determined by neutron diffraction data. Cyan is $Zn$ , brown is $Cu$ , green is $Cl$ , grey is $C$ , and white is $H$ . Taken from [1].	20
2-6	Binding sites in CuMFU-4l. Our work uses symmetry axis going through Site I and I*. Cyan is $Zn$ , brown is $Cu$ , green is $Cl$ , grey is $C$ , and white is $H$ . Taken from [1]. . . . .	20
4-1	Top graph shows the potential energy model for MOF-5 with ground state and first excited state energy for $H_2$ and $D_2$ . Bottom graph shows the probability densities. . . . .	29

- 4-2 Top graph shows the potential energy from the B3LYP-D2/def2-TZVPPD (def2-SVP) model from [1] with blue markers. Interpolated potential is shown by the black line. Bottom graph shows the probability densities for the ground state and first excited state for  $H_2$  and  $D_2$ . . . . . 32
- 4-3 Top graph shows the potential energy from the B3LYP-D2/6-31++G\*\*(6-31G\*) model from [1] with blue markers. Interpolated potential is shown by the black line. Bottom graph shows the probability densities for the ground state and first excited state for  $H_2$  and  $D_2$ . . . . . 34
- 5-1 Slice of the effective potential energy for  $H_2$  using Feynmann Hibbs equation at  $z = 3.04 \text{ \AA}$ .  $z = 3.04 \text{ \AA}$  was chosen by finding distance along the z axis where our effective potential energy was at a minimum. The red box shows the effective potential energy minimum in K and the coordinate values for that minimum in  $\text{\AA}$ . . . . . 38

# List of Tables

2.1	$\sigma$ and $\epsilon$ are the Lennard Jones parameters. $q(e^-)$ refers to the partial charge of the atoms in the primary binding site of MOF-5. Label refers to the numbers in Figure 2-2. Taken from [2]. . . . .	17
2.2	Lennard Jones parameters ( $\sigma$ , $\epsilon$ ) and partial charges ( $q(e^-)$ ) for atoms in the primary binding site of Mg-MOF-74. Label refers to the numbers in Figure 2-4. Taken from [13]. . . . .	19
4.1	Energy state solutions for MOF-5. All values are in kJ/mol. . . . .	28
4.2	Energy state solutions for Cu-MFU-4l. All values are in kJ/mol. B3LYP-D2/6-31++G** (6-31G*), B3LYP-D2/def2-TZVPPD (def2-SVP), and $\omega$ B97M-V refer to the three functionals used to create the potential energy models from [1]. . . . .	31



# Chapter 1

## Introduction

In the last century, Earth's average temperature has risen around 2 degrees Fahrenheit, a change largely due to greenhouse gas emissions [3]. Therefore, the need for alternative energy sources grows. One option for sustainable energy comes from hydrogen fusion reactions. Fusion occurs when hydrogen nuclei collide to form a helium atom, releasing a large amount of energy. This reaction is most easily done using hydrogen isotopes deuterium and tritium [10]. Unlike a typical hydrogen atom (protium), which contains no neutrons, deuterium has one neutron and tritium has two neutrons. All three hydrogen isotopes also contain an electron and a proton.

One of the many practical challenges to fusion energy generation is developing cost effective methods of hydrogen isotope separation. The natural abundance of deuterium is only 0.016% of hydrogen found on earth, and tritium is found in trace amounts [8]. Furthermore, the separation of deuterium and tritium from protium is especially difficult since they are chemically the same except for their masses. Current industrial processes for performing this separation, most notably the Girdler sulfide process and cryogenic distillation, are very energy consuming [7]. An alternative to these processes is using porous structures called metal organic frameworks (MOFs) to adsorb and release hydrogen isotopes as conditions are changed. When hydrogen isotopes interact with adsorption sites on these structures, the isotope with the heavier mass is preferentially adsorbed.

Quantum mechanically, the lowest possible energy state of a system is not the po-

tential energy minimum but slightly above. This difference between potential energy minimum and ground state energy is known as zero point energy (ZPE). A heavier object has a lower ZPE since quantum mechanical effects become less apparent as mass increases. Thus, inside a MOF, hydrogen isotopes will have different ZPEs and can be separated based on this difference, with a larger difference indicating more feasible separation.

Different MOFs will produce different spreads between the ZPEs of a protium molecule ( $H_2$ ) and a deuterium molecule ( $D_2$ ). This thesis explores this spread in ZPE between  $H_2$  and  $D_2$  inside three well characterized MOFs. The three MOFs explored, MOF-5, MOF-74, and CuMFU-4l, have experimentally measured  $H_2$  binding energies of 5, 10, and 33 kJ/mol respectively [2][13][1]. In each MOF, we took parametrized interaction models from the literature to generate the potential energy function along a particular symmetry axis. We then solve the Schrodinger equation for these potentials to determine the ground state energies for  $H_2$  and  $D_2$ . Ultimately, these results can be used help experimentalists choose candidate MOFs for isotope separation from the vast array of existing structures, put some bounds on what might be reasonable expected with MOFs for isotope separation, and help in the development of new MOFs for separation.

# Chapter 2

## MOFs Explored

Metal organic frameworks are crystalline materials constructed of metal ions and organic ligands. They have a 3D framework with pores that can adsorb molecules such as  $H_2$  and then release as conditions are changed [17]. Additionally, they have a high surface area per gram for adsorption. MOFs are also highly crystalline, meaning they are well characterized and can be easily reproduced [16]. There are numerous possible structures for MOFs, and the different permutations can be manipulated to achieve an optimal MOF for hydrogen isotope separation. This process of altering the possibilities is also known as tunability.

In a practical separation setup, a mixture of hydrogen isotopes is dosed into a MOF cooled to 77K. Then, the isotopes are extracted from the MOF using pressure or by heating up the MOF. The lighter isotope is released first, and this process is done repeatedly using some sort of feedback loop. The three MOFs studied in this thesis were chosen because they are well characterized MOFs that garnered much attention in the field of  $H_2$  storage and have been studied with neutron diffraction [12][1] [19]. There are a limited number of neutron scattering facilities and standard x-ray diffraction is insensitive to hydrogen isotopes due to its low atomic number [19]. Neutron diffraction reveals the crystalline location of the adsorbed  $D_2$  (neutron diffraction can only be done with  $D_2$  due to technical reasons but the two isotopes will be in the same location [20]), which is utilized in calculating the parameters and coordinates for our potential energy models. The structure and primary binding sites

of MOF-5, MOF-74, and Cu-MFU-4l are outlined below.

## 2.1 MOF-5

MOF-5 has cubic unit cells as shown in Figure 2-1. It has  $Zn_4O(CO_2)_6$  at its corners and edges made of  $C_6H_4$  ligands. While there are multiple binding sites within MOF-5, the primary site, located near the  $Zn_4O(CO_2)_6$  and depicted in Figure 2-2, will generally fill first [4]. The experimentally measured binding energy of  $H_2$  at the primary site is approximately 5 kJ/mol [4].

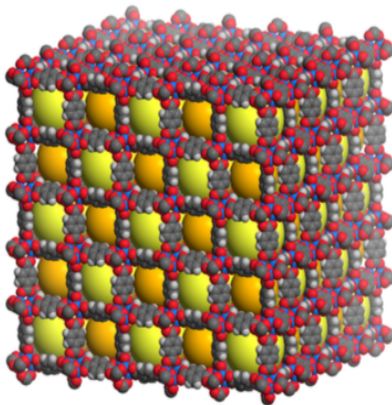


Figure 2-1: Depicted are stacked unit cells for MOF-5. Blue is  $Zn$ , red is  $O$ , dark grey is  $C$ , and light grey is  $H$ . The orange and yellow spheres depict the alternating pore sizes in the MOF. Taken from [15].

We chose MOF-5 because it has one of the highest hydrogen storage capacities by mass and was the first MOF to be posited as a  $H_2$  storage material [9]. It has been studied in great detail and has well established  $H_2$  binding sites. It also has readily available Lennard-Jones parameters and partial charges for each atom in the primary binding site. These parameters are taken from [2] and listed in Table 1, where  $\sigma$  and  $\epsilon$  are Lennard Jones parameters and  $q(e^-)$  refers to the partial charge of the atom. These are used in the potential energy model described in Section 3 where we parametrize how each atom would interact with a hydrogen molecule.



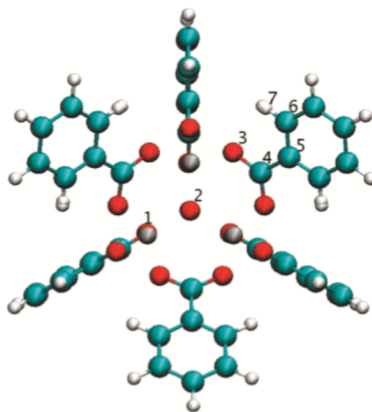


Figure 2-2: MOF-5 primary binding site. Numbers refer to the atom labels in Table 2.1. Grey is  $Zn$ , red is  $O$ , blue is  $C$ , and white is  $H$ . Image taken from [2].

Table 2.1:  $\sigma$  and  $\epsilon$  are the Lennard Jones parameters.  $q(e^-)$  refers to the partial charge of the atoms in the primary binding site of MOF-5. Label refers to the numbers in Figure 2-2. Taken from [2].

Atom	Label	$\sigma(\text{\AA})$	$\epsilon(\text{K})$	$q(e^-)$
Zn	1	2.46	62.4	1.85
O	2	3.12	30.2	-2.26
O	3	3.12	30.2	-1.01
C	4	3.43	52.8	1.10
C	5	3.43	52.8	-0.14
C	6	3.43	52.8	-0.05
H	7	2.57	22.1	0.15

The potential model in this thesis is focused on the primary binding site. We explored this potential along a single symmetry axis. We place the origin at a corner oxygen site with the z-axis pointing along the body diagonal that contains the primary  $H_2$  site.

## 2.2 MOF-74

MOF-74 was one of the first MOFs with an open metal site. These MOFs have a honeycomb-like structure, shown in Figure 2-3, and the open metal site lies at the intersection of the honeycomb [13]. The open metal site provides a highly favorable sorption site for the isotopes and, therefore, is at the center of the primary binding site shown in Figure 2-4. These MOFs produce  $H_2$  binding energies over 10 kJ/mol [13].

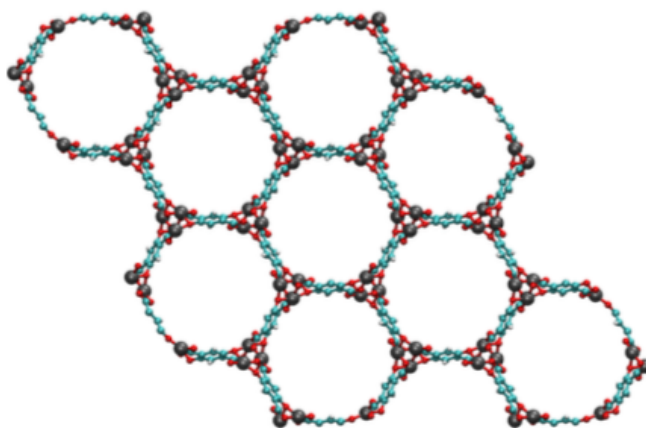


Figure 2-3:  $2 \times 2 \times 2$  unit cell system of Mg-MOF-74. Cyan is  $C$ , white is  $H$ , red is  $O$ , and grey is  $Mg$ . Taken from [13].

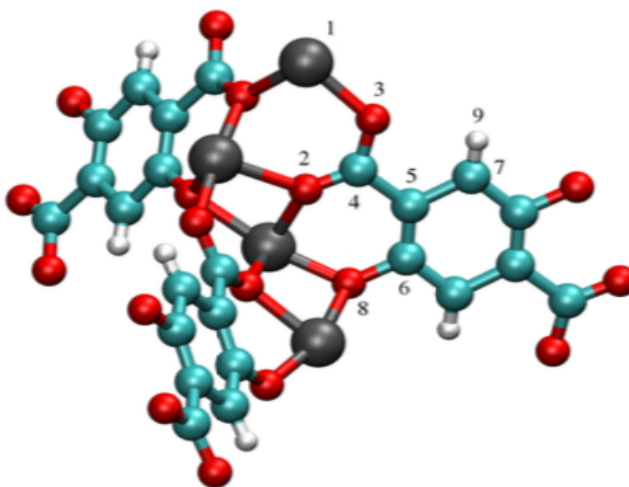


Figure 2-4: Mg-MOF-74 primary binding site. Numbers refer to the atom labels in Table 2.2. Cyan is  $C$ , white is  $H$ , red is  $O$ , and grey is  $Mg$ . Taken from [13].

Table 2.2: Lennard Jones parameters ( $\sigma$ ,  $\epsilon$ ) and partial charges ( $q(e^-)$ ) for atoms in the primary binding site of Mg-MOF-74. Label refers to the numbers in Figure 2-4. Taken from [13].

Atom	Label	$\sigma(\text{\AA})$	$\epsilon(\text{K})$	$q(e^-)$
Mg	1	2.69	55.9	1.68
O	2	3.12	30.2	-0.87
O	3	3.12	30.2	-0.77
C	4	3.43	52.8	0.93
C	5	3.43	52.8	-0.48
C	6	3.43	52.8	0.47
C	7	2.57	22.1	-0.38
O	8	3.12	30.2	-0.81
H	9	2.57	22.1	0.23

Mg-MOF-74 was studied in this thesis because it has readily available Lennard-Jones parameters listed in Table 2.2 and taken from Reference [13]. These parameters can be used to produce the potential model at Mg-MOF-74's primary binding site. The symmetry axis where the potential is explored starts at the Mg and expands out in the direction of the adsorbed isotope.

## 2.3 CuMFU-4l

The recently synthesized Cu-MFU has the highest hydrogen binding energy of any MOF (32 kJ/mol) [1]. It is made up of pentanuclear zinc nodes bridged by ligands that are then connected to an open metal-site [1]. A portion of Cu-MFU-4l determined by neutron diffraction data is shown in Figure 2-5. According to Reference [1], there are three known binding sites inside of CuMFU-4l, as shown in Figure 2-6.

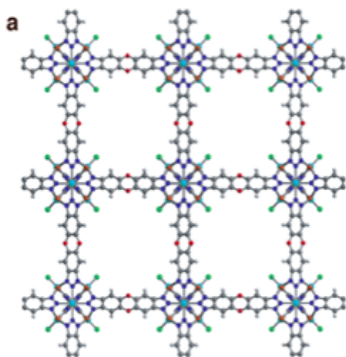


Figure 2-5: Portion of CuMFU-4l determined by neutron diffraction data. Cyan is *Zn*, brown is *Cu*, green is *Cl*, grey is *C*, and white is *H*. Taken from [1].

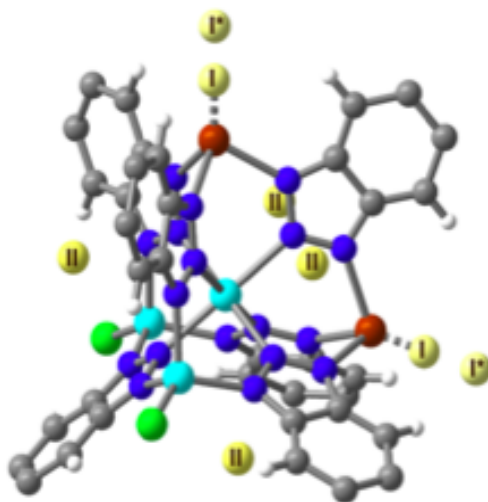


Figure 2-6: Binding sites in CuMFU-4l. Our work uses symmetry axis going through Site I and I\*. Cyan is *Zn*, brown is *Cu*, green is *Cl*, grey is *C*, and white is *H*. Taken from [1].

The primary binding site is located at the trigonal, pyramidal  $Cu^+$  sites and it is labeled site 1 in Figure 2-6 [1]. Site II is located at the windows of the pentanuclear tetrahedral nodes. Lastly, site I\* is located directly above site I. Our potential energy model explored the potential along the symmetry axis going from the  $Cu^+$  towards adsorption site I and I\*. Therefore, site II is not explored in this model. In MFU-4l, site II yields a binding energy of around 5 kJ/mol, which is around 6 times smaller than the 33 kJ/mol binding energy of CuMFU-4l [1].

# Chapter 3

## Theory

### 3.1 Zero Point Energy

Inside a MOF,  $H_2$  and  $D_2$  will have different zero point energies (ZPEs) and can be separated based on this difference, with a larger difference indicating more feasible separation. The ZPE of a system is the difference between the potential energy minimum and the lowest possible energy state for that system. It is defined as

$$E_0 - V_{min} \tag{3.1}$$

where  $E_0$  is the ground state energy of a system and  $V_{min}$  is the potential energy minimum. This difference can be understood in terms of the Heisenberg Uncertainty Principle. For any object dominated by quantum mechanical behavior, the position and the momentum are not determined beyond the quantum limit

$$\Delta x \Delta p \geq \frac{\hbar}{2} \tag{3.2}$$

where  $\Delta x$  is the uncertainty in position,  $\Delta p$  is the uncertainty in momentum, and  $\hbar$  is the reduced Planck's constant. An object cannot have both zero momentum and be in a specific rest position, so there still exists an energy at the "zero" point of a system. A heavier object has a lower zero point energy since quantum mechanical

effects become less apparent as mass increases. Thus, inside a MOF, heavier hydrogen isotopes will have lower ZPEs.

## 3.2 Potential Energy Models

Due to their complexity, the complete potential energy of MOFs has not been able to be theoretically determined. The adsorption between MOFs and hydrogen isotopes involves many different interactions such as dispersion, electrostatics, and charge transfer. Instead, there are several models based on quantum mechanical assumptions and experimental data. These models that approximate the potential energy are not always reliable. Most adsorbates are much more massive than hydrogen and considering quantum effects such as ZPE is not necessary. In our work, we used parametrized interaction models from the literature to generate the potential energy function along a particular symmetry axis. This potential energy is used to numerically calculate the energy states and ZPE for  $H_2$  and  $D_2$ . The potential energy models used are outlined below.

### 3.2.1 MOF-5 and MOF-74 Potential Energy Model

The potential energy along the single axis was approximated for MOF-5 and MOF-74 using a combination of Lennard-Jones 6-12 and electrostatic potentials for each atom in the primary binding site described in Sections 2.1 and 2.2. The Lennard-Jones equation models attractive and repulsive forces between non-bonding atoms based on their separation distance. It follows the form:

$$U_{LJ}(R) = 4 \sum_{i=1}^n \epsilon_i \left( \left( \frac{\sigma_i}{R} \right)^{12} - \left( \frac{\sigma_i}{R} \right)^6 \right) \quad (3.3)$$

where  $R$  is the separation distance,  $\epsilon$  and  $\sigma$  are the Lennard-Jones parameters in Table 2.1 and Table 2.2 in Chapter 2, and  $n$  refers to the number of atoms in the binding site. The attractive interactions are found through the  $(\frac{\sigma_i}{R})^6$  term. The  $(\frac{\sigma_i}{R})^{12}$  term comes from a fitting parameter that models repulsive interactions by using overlapping wave

functions. The Lennard Jones equation is a binary model. Equation 3.3 sums the binary interaction between each atom in the binding site and the adsorbed hydrogen molecule.

The electrostatic contribution to potential follows the equation

$$U_{pol}(R) = -k\frac{\alpha}{2}\left[\sum_{i=1}^n \frac{q_i}{R^2}\hat{r}\right]^2 \quad (3.4)$$

where  $\alpha$  is the polarization constant of  $H_2$  and  $q$  is the partial charge for each atom in the MOF. Our model uses the  $J = 0$  ground state value for  $\alpha$  of 0.675 Å [18]. The complete potential energy is found by summing these two equations for each atom in the binding site of that MOF. The Lennard-Jones component falls off with distance at  $\frac{1}{R^6}$  and the electrostatic component falls off at  $\frac{1}{R^4}$  so we limited our binding sites to only include atoms within around 6 Å of the adsorbed isotope.

The Lennard-Jones parameters and partial charges for MOF-5 and MOF-74 are found in Table 2.1 and Table 2.2 respectively. These parameters and partial charges come from Reference [2] and [13]. The partial charges were found using a charge fitting technique on different fragments of the MOF. The Lennard-Jones parameters were found using a full periodic table of force fields for molecular simulations called the Universal Force Field [14].

### 3.2.2 MOF-5 and MOF-74 Potential Energy Model Assumptions

It is accepted that this simplified potential energy model for MOF-5 and MOF-74 may not predict energies consistent with experimental data as the model makes two significant assumptions: the potential energy is dominated by electrostatic and van der Waals interactions and any disturbances the MOF makes to the hydrogen molecule are small enough that the potential can be modeled using free hydrogen parameters. Within these assumptions, the energy of an adsorbed molecule is the sum of its rotational, translational, and vibrational energies

$$E_{tot} = E_{rot} + E_{vib} + E_{trans} \quad (3.5)$$

The translational energy refers to the center of mass vibration and the vibrational energy refers to the internal vibrations of the hydrogen molecule.  $H_2$  has 3 translational coordinates as well as 2 rotational and 1 vibrational coordinates. Before adsorption, the vibrational energy can be approximated as  $\langle E_{vib} \rangle = 1/2h\nu_v$ , with  $\nu_v$  being the harmonic vibrational frequency. Thus, there is already an existing difference between  $H_2$  and  $D_2$  vibration before adsorption. After adsorption, the vibrational frequency of both will change since interactions with the MOF induces a dipole moment on the hydrogen. This new frequency can be measured using IR spectroscopy. Because the vibrational energy already has an existing difference between  $H_2$  and  $D_2$  before adsorption, the change to this difference after adsorption is minor in comparison to the translational change (there is no quantized translational motion in the gas phase).

Solid  $H_2$  behaves like a "quantum solid" and still rotates in its adsorbed state. However, the average difference between  $H_2$ 's rotational energy before and after the  $H_2$  is adsorbed is much harder to measure and is generally neglected in comparison to the order of magnitude higher vibrational energy. Our proposed model for MOF-74 and MOF-5 approximates the potential energy by assuming that the translational energy will dominate in a single dimension and uses parameters for  $H_2$  in its rotational  $J = 0$  ground state and  $v = 0$  vibrational state. Therefore, the potential we refer to is the center of mass potential of the hydrogen molecule as a whole. This approximation becomes less accurate as binding energy increases.

### 3.2.3 Cu-MFU-4l Potential Energy Model

CuMFU-4l also contains open metal sites and has an even larger binding energy. Therefore, the potential energy model for MOF-5 and MOF-74 was not applied to CuMFU-4l and instead values were taken from density functional theory calculations done by Reference [1]. Density functional theory involves using functionals to rep-



resent electron density. These calculations take into account more of the changes to vibrational and rotational energy during adsorption, but the potential is still only calculated along a single translational coordinate. The three different models each use a different functional to generate their potential energy values. The three models are labeled as B3LYP-D2/6-31++G\*\* (6-31G\*), B3LYP-D2/def2-TZVPPD (def2-SVP), and  $\omega$ B97M-V. These models use data from neutron diffraction to establish the atom coordinates. Neutron diffraction works under the assumption that the structure under study is consistent and repeating. When CuMFU-4l is formed, some of the  $[Zn - Cl]^+$  clusters are randomly replaced with  $Cu^+$ . This disorder creates noise in the diffraction measurement and adds uncertainty to their potential energy model.

### 3.3 Solving the Schrodinger Equation

The energy states for a quantum mechanical system are found using the Schrodinger Equation. The time independent Schrodinger equation is

$$\frac{\hbar^2}{2m} \frac{\partial^2 \psi}{\partial z^2} + V(z)\psi = E\psi \quad (3.6)$$

where  $\Psi$  is the wave function,  $E$  is the energy,  $\hbar$  is the reduced Planck constant,  $z$  is the distance along the symmetry axis, and  $V$  is the potential energy. We used the potential energy functions described in Section 3.2. If  $z$  is discretized, the second derivative can be approximated using the equation:

$$\frac{\partial^2 \psi}{\partial z^2} \Big|_{z_i} = \frac{\psi_{i+1} + \psi_{i-1} - 2\psi}{(z_i - z_{i-1})^2} \quad (3.7)$$

We used equally spaced points to discretize  $z$ , so we can refer to  $z_i - z_{i-1}$  as  $\Delta z$ . Then, the full time independent Schrodinger equation can be solved numerically using the equation

$$\frac{\psi_{i+1} + \psi_{i-1} - 2\psi}{\Delta z^2} = \frac{2m}{\hbar} (V_i - E)\psi_i \quad (3.8)$$

We used a coordinate range that extended far enough from the potential minimum

so that boundary conditions could be imposed. We set the first two values of  $\Psi$  to be 0 and 0.001, estimated a value for E, and then propagated iteratively through our coordinate range. We ran several control measurements on a range from 0.05 to 0.005 on our second initializing value for  $\Psi$ . We found that changing this value did not alter our wave function and energy state results. We estimated values for E until the wave function approached zero at the end of our coordinate range and followed expected boundary conditions. For each MOF, we calculated the wave function and energy states for both  $H_2$  and  $D_2$ . We determined the energy value for both the first excited state and the ground state.

# Chapter 4

## Results and Analysis

We solved the Schrodinger equation numerically with the potential energy models described in Section 3, calculating the first two energy states and wave functions inside of the three MOFs. This was done for both  $H_2$  and  $D_2$ . From the energy states, we calculated the ZPE spread between  $H_2$  and  $D_2$ . All energies are stated in  $\frac{kJ}{mol}$  in the convention of References [1], [2], and [13]. For conversion purposes,  $1 \frac{kJ}{mol} = 120K = 0.01eV = 84cm^{-1}$ .

### 4.1 MOF-5

The energy state solutions for MOF-5 are listed in Table 4.1. The calculated wave functions were normalized and squared to find the probability densities. The potential energy plot and probability densities are shown in Figure 4-1.

Our potential model yielded a minimum potential energy of  $-9.48 \frac{kJ}{mol}$ . Unlike MOF-74 and CuMFU-4l, MOF-5 does not contain an open metal site. Therefore, the charge interactions between MOF-5 and the adsorbed isotope are less significant than MOF-74 and CuMFU-4l, and our model assumptions described in Section 3 are more reasonable. However, the ground state energy for  $H_2$  was determined to be  $-8.93 \frac{kJ}{mol}$ , and experimental data places the  $H_2$  binding energy at around  $-5 \frac{kJ}{mol}$  [2]. Our value is around 1.8 times larger than the experimental value. Experimental infrared spectroscopy data indicates a difference between ground state energy and

first excited state energy or  $H_2$  in MOF-5 of  $0.67 \frac{kJ}{mol}$  [6]. Our theoretical value for this was  $1.02 \frac{kJ}{mol}$ , which is around 1.5 times larger than experiment. The ZPE for each isotope is calculated by finding the difference between the potential energy minimum and the isotope ground state as described in Section 3.1. The value for  $H_2$  ZPE was  $0.55 \frac{kJ}{mol}$ , and the value for  $D_2$  ZPE was  $0.38 \frac{kJ}{mol}$ , giving a difference of  $0.17 \frac{kJ}{mol}$ . As a first approximation we can assume a linear relationship between potential well depth and ZPE. We can also make this approximation for the difference between first excited and ground state energy and ZPE. That is, if our model overestimates the well depth and difference between first excited and ground state energy by a factor of around 1.5-1.8 then it should also overestimate the ZPE by a factor of 1.5-1.8. This assumption yields a ZPE difference of around  $0.1 \frac{kJ}{mol}$ .

Table 4.1: Energy state solutions for MOF-5. All values are in kJ/mol.

All energies in kJ/mol	MOF-5 Potential Model
Potential Minimum	-9.48
$H_2$ Ground State Energy	-8.93
$H_2$ First Excited State Energy	-7.90
$D_2$ Ground State Energy	-9.09
$D_2$ First Excited State Energy	-8.39
Difference between $H_2$ and $D_2$ GS	0.17
ZPE $H_2$	0.55
(Excited-Ground) $H_2$	1.02
ZPE $D_2$	0.38
(Excited-Ground) $D_2$	0.70

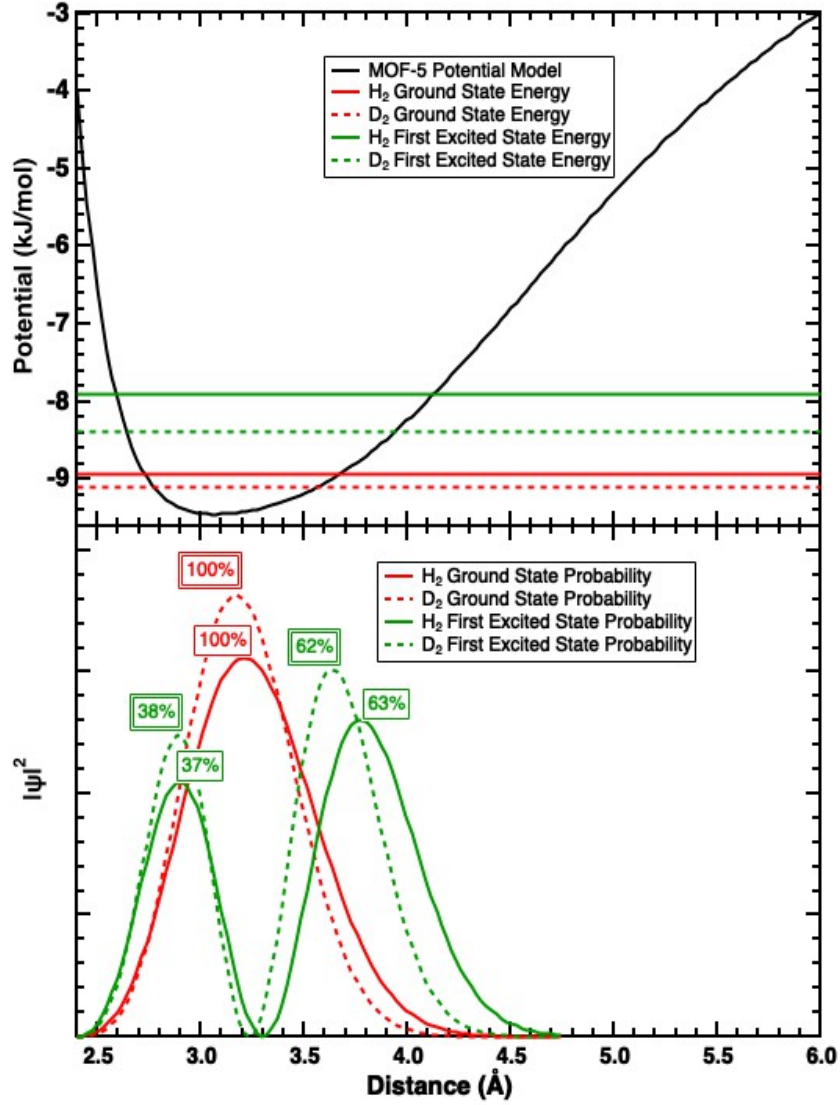


Figure 4-1: Top graph shows the potential energy model for MOF-5 with ground state and first excited state energy for  $H_2$  and  $D_2$ . Bottom graph shows the probability densities.

The wave functions for both  $H_2$  and  $D_2$  are close to that of a simple harmonic oscillator. In a one-dimensional harmonic oscillator, the energy eigenvalues ( $E_n$ ) are

$$E_n = (n + 1)\hbar\sqrt{\frac{k}{m}} \quad (4.1)$$

where  $\hbar$  is the reduced Planck constant and  $k$  is the spring constant and  $m$  is the mass.  $D_2$  has twice the mass of  $H_2$  and therefore the ratio of energy states between

$H_2$  and  $D_2$  in a simple harmonic oscillator will be equal to  $\sqrt{2} = 1.414$ . The ratio  $D_2$  ZPE and  $H_2$  ZPE from our model for MOF-5 is 1.44.

## 4.2 MOF-74

The potential energy model proposed in Section 3 uses parameters from a different paper than the MOF-5 paper, but they are produced by the same group. It appears this potential energy model was less reliable and was not able to appropriately model the potential energy of Mg-MOF-74. The potential produced was overly attractive, calculating potential minimum values 3 times the size of experimental binding energy results. Our model assumes the potential energy is dominated by van der Waals and electrostatic interactions. However, the presence of an open metal site in MOF-74 makes interactions such as charge transfer and the forming of a chemical bond are more significant. The authors of Reference [13] also noted the challenges of incorporating the open metal site and restricted their work to predicting an induced dipole moment. Additionally, our model only included atoms in the MOF that were fewer than 6 Å away from the adsorbed isotope. In future work, our model could be extended to include more long range forces.

## 4.3 CuMFU-4l

The three potential energy models from [1] were used to solve the discretized Schrodinger equation. We calculated the ground state energy and first excited state energy for  $H_2$  and  $D_2$  for each potential model. The results from this are listed in Table 4.2. The calculated wave functions were normalized and squared to find the probability densities. The potential energy plot and probability densities for the B3LYP-D2/6-31++G\*\* (6-31G\*) model and the B3LYP-D2/def2-TZVPPD (def2-SVP) model are shown in Figure 4-3 and Figure 4-3 respectively.

Table 4.2: Energy state solutions for Cu-MFU-4l. All values are in kJ/mol. B3LYP-D2/6-31++G\*\* (6-31G\*), B3LYP-D2/def2-TZVPPD (def2-SVP), and  $\omega$ B97M-V refer to the three functionals used to create the potential energy models from [1].

All Energies in kJ/mol	B3LYP-D2/6-31++G** (6-31G*)	B3LYP-D2/def2-TZVPPD (def2-SVP)	$\omega$ B97M-V
Potential Minimum	-23.9	-43.4	-50.3
$H_2$ Ground State Energy	-17.5	-36.3	-42.8
$H_2$ First Excited State Energy	-7.6	-24.2	-29.7
$D_2$ Ground State Energy	-19.3	-38.3	-44.8
$D_2$ First Excited State Energy	-9.2	-29.1	-35.1
Difference between $H_2$ and $D_2$ GS	1.8	2.0	2.1
ZPE $H_2$	6.4	7.2	7.5
(Excited-Ground) $H_2$	9.9	12.1	13.1
ZPE $D_2$	4.6	5.1	5.4
(Excited-Ground) $D_2$	10.1	9.1	9.8

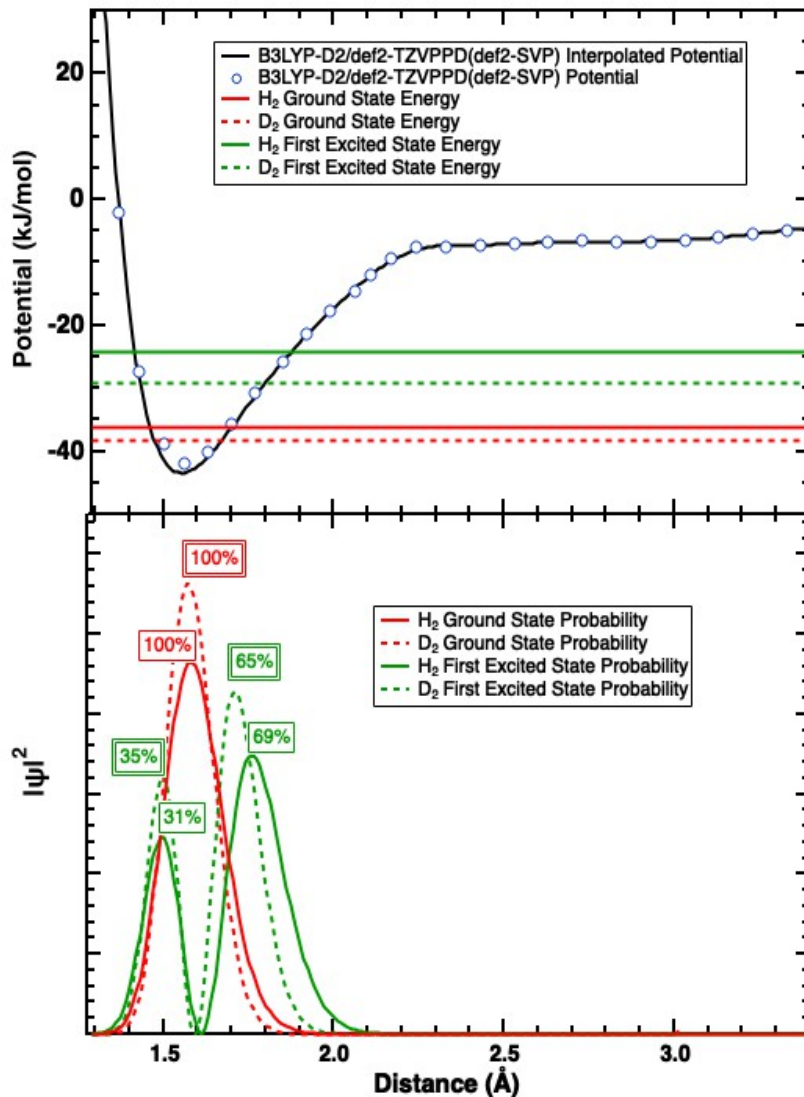


Figure 4-2: Top graph shows the potential energy from the B3LYP-D2/def2-TZVPPD (def2-SVP) model from [1] with blue markers. Interpolated potential is shown by the black line. Bottom graph shows the probability densities for the ground state and first excited state for  $H_2$  and  $D_2$ .

The spread in potential minimum between the three different models highlights the challenges of modeling  $H_2$  interactions within a complex strongly adsorbing system. However, the extracted ZPE difference is relatively consistent ( $1.8 \frac{kJ}{mol}$  to  $2.0 \frac{kJ}{mol}$ ). This may be fortuitous, but in this case we can compare the predicted number to an experimental value of  $2.3 \frac{kJ}{mol}$  determined by isotope adsorption measurements at different temperatures [5]. A significant limitation in our approach is that it only



uses the potential in a single dimension. As of yet, potential energy calculations have not been done for more than one direction. While we expect the difference in ZPE to be dominated motion along this axis, a more complete view of the energy states would include data for a full 3D surface. We are currently in communication with the authors of Reference [1] in the hope that they might be able to perform the additional numerical calculations.

### 4.3.1 B3LYP-D2/6-31++G\*\*(6-31G\*) Model

The potential energy given by the B3LYP-D2/6-31++G\*\*(6-31G\*) model from [1] includes a second potential well. This well refers to the binding site labeled Site I\* in Section 2.3. It occurs slightly after the primary binding site labeled Site I on the symmetry axis. The plot of the potential energy given by the B3LYP-D2/6-31++G\*\*(6-31G\*) model is shown in Figure 4-2 by the blue, circular markers. These data points were then interpolated to have even step sizes along the coordinate axis so that the Schrodinger Equation could be solved. The interpolated potential is shown in Figure 4-2 by the black line.

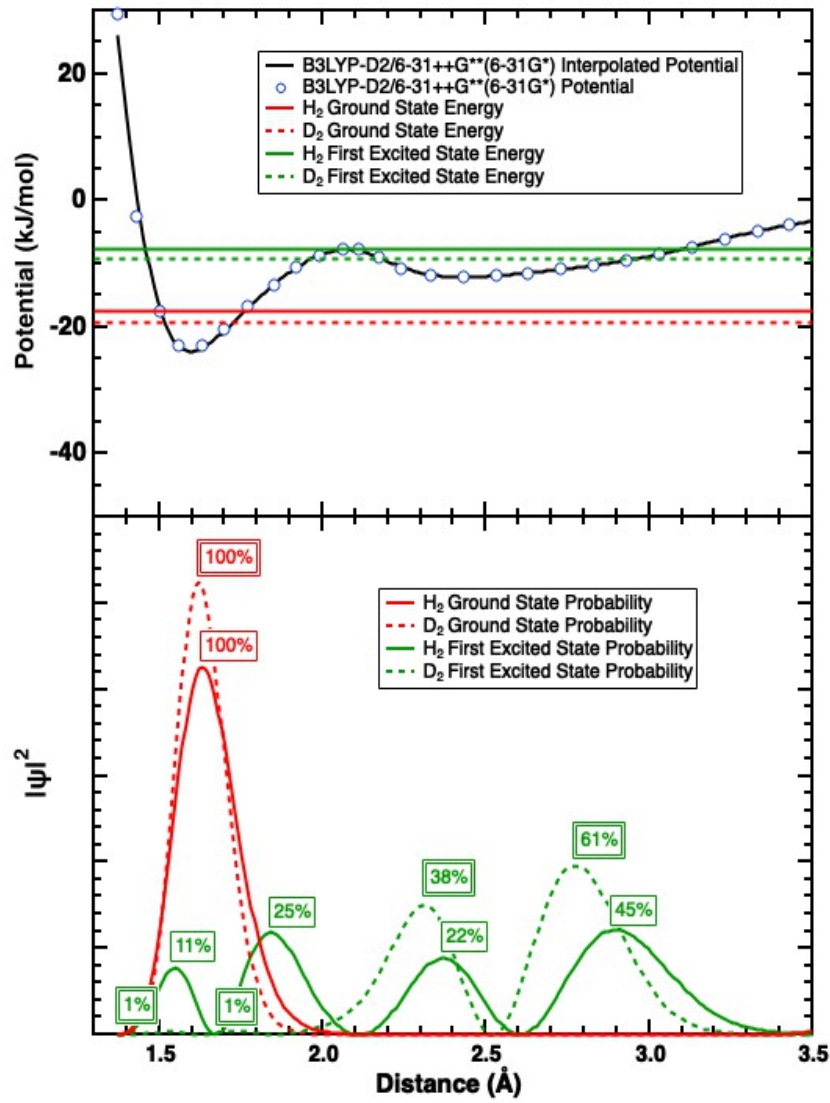


Figure 4-3: Top graph shows the potential energy from the B3LYP-D2/6-31++G\*\*(6-31G\*) model from [1] with blue markers. Interpolated potential is shown by the black line. Bottom graph shows the probability densities for the ground state and first excited state for  $H_2$  and  $D_2$ .

Although not directly related to hydrogen isotope separation, our results from the B3LYP-D2/6-31++G\*\*(6-31G\*) model yielded an interesting outcome consistent with experimental data found in [1]. When  $D_2$  was dosed into a sample of CuMFU-4l at 300K, Site I was occupied with around 50% of the total  $D_2$  dosed. However, when  $D_2$  was dosed at 40K, Site I was occupied by only 9% of the total  $D_2$  dosed. These occupancy values for Site I were taken from diffraction data collected at 7K.

Reference [1] suggests that the lack of presence in Site I indicates an increased presence in Site I\*. Site I\* is described to be a precursor adsorption state that is separated from Site I by a thermal barrier. Thus, when the  $D_2$  is dosed into the system at a colder temperature, it does not have enough energy to overcome this thermal barrier. Therefore the dosed  $D_2$  stays in Site I\* and is unable to reach its ground state. Our calculated probability densities show that in the ground state, both  $H_2$  and  $D_2$  will be completely in the first potential well, or Site I. However, in their first excited states, the probability densities of  $H_2$  and  $D_2$  exist in the second potential well (Site I\*) too. This aligns with the suggestion that Site I\* is a precursor state that exists when the isotope is not able to reach its ground state.

The inclusion of the second potential well, or second binding site, also shows a noticeable difference between  $H_2$  and  $D_2$  behavior within the MOF. In its first excited state,  $H_2$  is relatively likely to occupy either binding site, but  $D_2$  is much less likely to occupy Site I. Objects dominated by quantum mechanics experience a phenomenon called quantum tunneling where their wave function can propagate through potential barriers. Because  $H_2$  is less massive than  $D_2$ , it is more subject to quantum mechanical effects. Therefore, we speculate that  $H_2$  may be more likely to propagate through the potential barrier between Sites I\* and Site I.

## 4.4 Difference in ZPE Comparison

The ZPE difference between  $H_2$  and  $D_2$  in MOF-5 and the B3LYP-D2/6-31++G\*\*(6-31G\*) model of CuMFU-4l are 0.17 kJ/mol and 1.8 kJ/mol respectively. As expected, the ZPE difference between  $H_2$  and  $D_2$  increases as binding energy increases. The ZPE difference also increases with the curvature of the potential well. MOF-5's potential well drops about 5 kJ/mol in around 0.5Å where as CuMFU-4l's potential well at Site I from the B3LYP-D2/6-31++G\*\*(6-31G\*) model drops about 20 kJ/mol in around 0.3 Å. According to our model, CuMFU-4l has a ZPE spread between  $H_2$  and  $D_2$  that is around 10 times larger than MOF-5. A better analysis of the relationship between ZPE spread, binding energy, and potential energy curvature could be

done by comparing the ZPE spreads in MOF-5 and CuMFU-4l to that of Mg-MOF-74. However, under the assumptions that our potential energy model made, we were unable to appropriately model Mg-MOF-74.

# Chapter 5

## Conclusion and Future Work

### 5.1 Feynmann-Hibbs Equation

Our potential energy model works under the assumption that the potential in our defined  $z$  direction will dominate. Even if the Schrodinger equation was solved using our potential model along the other two orthogonal axes, the complete solution for energy state assumes the wave functions are separable. The Feynmann Hibbs equation is an alternative method to solving the Schrodinger equation in three dimensions. The Feynmann Hibbs equation operates under the assumption that the potential energy at the binding site can be approximated as a three dimensional simple harmonic oscillator. It averages over a weighted Gaussian wave function to get the energy states [11]. It follows the form:

$$U_{FH}(x_0, y_0, z_0) = \frac{\sum_{x,y,z} U(x, y, z) e^{-\sigma((x-x_0)^2+(y-y_0)^2+(z-z_0)^2)}}{\sum_{x,y,z} e^{-\sigma((x-x_0)^2+(y-y_0)^2+(z-z_0)^2)}} \quad (5.1)$$

where  $U(x, y, z)$  is our potential model and  $\sigma$  is the weighting factor. The weighting factor is based on the de Broglie wavelength of the isotope adsorbed, and it equals  $\frac{6m}{\beta h^2}$ , where  $m$  is the mass of the isotope and  $\beta$  is  $\frac{1}{k_B T}$  [11]. This model still is based off of the parameters from Reference [2]. The effective potential that is created by the Feynmann Hibbs equation can be used to model diffusion of the isotope within the MOF. We used the Feynmann Hibbs equation to calculate the binding energy

of  $H_2$  in MOF-5 at 77K. We placed the origin at a corner oxygen (atom labeled 2 in Figure 2-2 in Section 2.1) with the z-axis pointing along the body diagonal that contains the primary  $H_2$  site. The potential energy equations in Section 3.2 were used for  $U(x, y, z)$ .

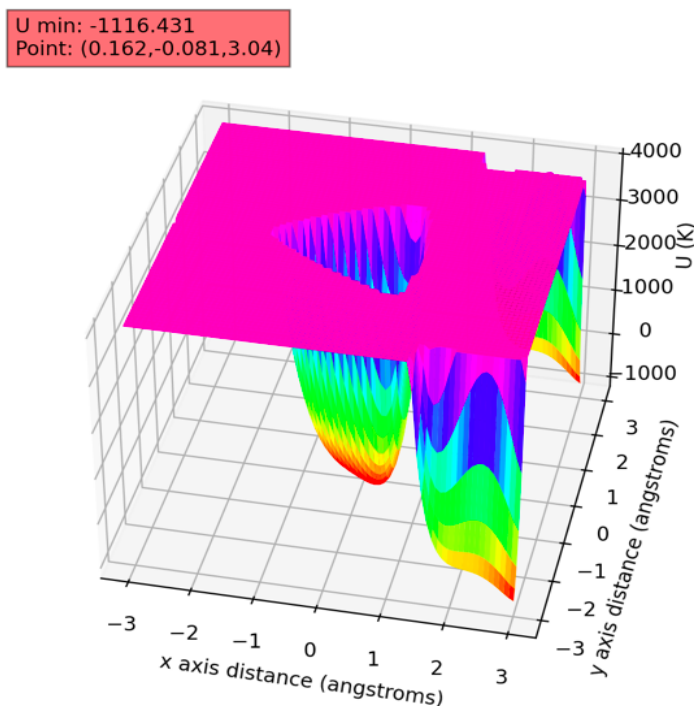


Figure 5-1: Slice of the effective potential energy for  $H_2$  using Feynmann Hibbs equation at  $z = 3.04 \text{ \AA}$ .  $z = 3.04 \text{ \AA}$  was chosen by finding distance along the z axis where our effective potential energy was at a minimum. The red box shows the effective potential energy minimum in K and the coordinate values for that minimum in  $\text{\AA}$ .

An unfinished version of this work is shown in Figure 5-1. The plot shown is a slice of the effective potential energy at  $z = 3.04 \text{ \AA}$ . We chose  $z = 3.04 \text{ \AA}$  by finding distance along the z axis where our effective potential energy was at a minimum. The red box in Figure 5-1 shows the effective potential energy minimum in K and the coordinate values for that minimum in  $\text{\AA}$ . The effective potential energy minimum is around -1120 K, or  $-9.33 \frac{\text{kJ}}{\text{mol}}$ . Compared to our  $H_2$  potential energy minimum from iteratively solving the Schrodinger equation of  $-8.93 \frac{\text{kJ}}{\text{mol}}$ , the Feynmann-Hibbs model is farther from the experimentally measured  $-5 \frac{\text{kJ}}{\text{mol}}$ . However, the Feynmann

Hibbs equation is helpful for modelling diffusion and visualizing the three dimensional potential energy.

The Feynmann Hibbs approach is computationally intensive as each point is found by doing a weighted average over every other point in our three dimensional grid space. That is, in an  $n \times n \times n$  grid space, each point has a complexity of  $O(n^3)$  and there are  $n^2$  points calculated to make an effective potential energy slice for a single atom. Therefore, the entire process has a complexity of  $O(n^5)$ . Figure 5-1 includes 13 atoms from MOF-5 using dimensions of  $75 \times 75 \times 75$ , resulting in approximately 30 billion iterations. Future work to possibly cut computational run time could involve using existing Python libraries that optimize vector operations.

## 5.2 Concluding Statements

Our theoretical approach to exploring the ZPE spread between  $H_2$  and  $D_2$  was comparable to experimental values and suggests that theoretical models might be useful in helping determine optimal MOFs for hydrogen isotope separation. Our results suggest that MOFs with higher binding energies are more efficient for hydrogen isotope separation, but would need to be confirmed by future, more elaborate modeling. The most significant limitation to theoretical models is the complexity of the system. As with most many-body problems, it is difficult to include every interaction of every particle during the hydrogen adsorption process. Thus, theoretical models are limited by assumptions and approximations to capture the interactions inside a MOF. Specifically, this work is limited by the assumption that A) the potential energy is dominated by translational energy in a single dimension, and B) the potential for MOF-5 and MOF-74 is dominated by electrostatic and van der Waals interactions. Possible future work to expand our theoretical model would be to include translational energy in three dimensions and expand the potential energy model for MOF-74. Despite these limitations, theoretical models offer insight into cost effective hydrogen isotope separation methods inside of MOFs that can lend themselves to sustainable energy sources.





# Chapter 6

## Appendix

### 6.1 Code Used to Calculate Energy States

#### 6.1.1 point.py

```
1 import numpy as np
2 #point class
3 #the atom class inherits from the point class (each atom has a
   coordinate)
4 #these classes are just an efficient way to store and access the
   atoms in a primary MOF binding site
5
6 class point:
7     def __init__(self, x, y, z):
8         self.x = x
9         self.y = y
10        self.z = z
11
12    def setPoint(self, x, y, z):
13        self.x = x
14        self.y = y
15        self.z = z
16
17    #function to give distance between two points
```

```

18     def distance(self, pt):
19         return np.sqrt((self.x-pt.x)**2+(self.y-pt.y)**2+(self.z-pt.
20         z)**2)
21
22     #function to rotate the coordinates (coordinate transformations)
23     #give this method the axis you want to rotate on ('x', 'y', or '
24     z') and the angle you want to rotate in degrees
25
26     def rotate(self, rotAxis, angle):
27         angle *= np.pi/180
28         if rotAxis == 'x':
29             originaly = self.y
30             self.y = self.y*np.cos(angle)+self.z*np.sin(angle)
31             self.z = -originaly*np.sin(angle)+self.z*np.cos(angle)
32         if rotAxis == 'y':
33             originalx = self.x
34             self.x = self.x*np.cos(angle)-self.z*np.sin(angle)
35             self.z = originalx*np.sin(angle)+self.z*np.cos(angle)
36
37     #function to do coordinate translation
38     #give this method the axis you want to rotate on ('x', 'y', or '
39     z') and the amount you want to shift
40
41     def translation(self, axis, amount):
42         if axis == 'x':
43             self.x += amount
44         if axis == 'y':
45             self.y += amount
46         if axis == 'z':
47             self.z += amount
48
49     #function to flip direction of all coordinates
50
51     def flipDirection(self):
52         self.x = -self.x
53         self.y = -self.y
54         self.z = -self.z

```

## 6.1.2 atom.py

```
1 import point as pt
2 import numpy as np
3
4 #atom class - defines a single atom in a primary MOF binding site,
   inherits from point class in point.py
5 class atom(pt.point):
6     def __init__(self, x, y, z, charge=0, sigma=0, epsilon=0, mass
   =0):
7         super().__init__(x,y,z)
8         self.charge = charge
9         self.sigma = sigma
10        self.epsilon = epsilon
11        self.mass = mass
12
13        #references point class to get the distance of one atom to
   another
14        def magnitude(self,atom):
15            return self.distance(atom)
16
17        #gives combined sigma value for hydrogen and specific atom (used
   in lennard-jones equation)
18        def jointSigma(self,atom):
19            return (self.sigma+atom.sigma)/2
20
21        #gives combined sigma value for hydrogen and specific atom (used
   in lennard-jones equation)
22        def jointEpsilon(self,atom):
23            return np.sqrt(self.epsilon * atom.epsilon)
24
25        #just displays the coordinate of the atom
26        def display(self):
27            print(self.x, self.y, self.z)
28
29
```

```

30 #MOF-5 atom definitions taken from Mercury and Rosnow Thesis. Sigma
    and epsilon values taken from space paper
31 #Zn:0-2, O1:3, O2:4-9, C1:10-12
32 #listing goes x, y, z, charge, sigma, epsilon, mass
33 #origin is center oxygen
34 atoms = [atom(-1.855325180842072, 0.0, 0.656043110880173, 1.8529,
    2.4616, 62.3993, 0),
35         atom(0.9276625904210358, -1.606758738890191,
    0.656043110880173, 1.8529, 2.4616, 62.399, 0),
36         atom(0.9276625904210358, 1.6067587388901914,
    0.656043110880173, 1.8529, 2.4616, 62.399, 0),
37         atom(0.0, 0.0, 0.0, -2.2568, 3.118, 30.19, 0),
38         atom(-2.2071535575638297, -1.575575329839865,
    1.745098923652563, -1.0069, 3.118, 30.19, 0),
39         atom(2.468065039999284, -1.1236633859835425,
    1.745098923652563, -1.0069, 3.118, 30.19, 0),
40         atom(-0.26091148243545437, 2.6992387158234075,
    1.745098923652563, -1.0069, 3.118, 30.19, 0),
41         atom(2.468065039999284, 1.1236633859835425,
    1.745098923652563, -1.0069, 3.118, 30.19, 0),
42         atom(-0.2609114824354546, -2.6992387158234075,
    1.745098923652563, -1.0069, 3.118, 30.19, 0),
43         atom(-2.2071535575638292, 1.575575329839865,
    1.745098923652563, -1.0069, 3.118, 30.19, 0),
44         atom(-1.46152887986063, -2.53144227664784,
    2.0669139636988607, 1.0982, 3.431, 52.84, 0),
45         atom(2.92305775972126, 0.0, 2.0669139636988607, -0.1378,
    3.431, 52.84, 0),
46         atom(-1.4615288798606296, 2.53144227664784,
    2.0669139636988607, -0.0518, 3.431, 52.84, 0)]
47
48
49 #atom definitions for protium and deuterium (set coordinates to zero
    to start but changes later in the code)
50 muH = 2*(1.7*10**-27)
51 hydrogen = atom(0, 0, 0, 0, 2.571, 22.14, muH)

```

```
52 deuterium = atom(0, 0, 0, 0, 2.571, 22.14, 2*muH)
```

### 6.1.3 utilities.py

```
1 # Common parameters
2 import numpy as np
3 from scipy.misc import derivative
4
5
6 # parameters in the electrostatic potential for H2
7 # units of angstroms squared times q
8 Q = 0.12
9 # k is used when using mks units - is in units of 1/(F/m)
10 k = 1/(4*np.pi*8.8*(10**-12))
11 # print(k)
12 # charge of electron in coulombs
13 qe = 1.6*10**-19
14 # alpha is in unites of Angstroms cubed
15 alpha = 0.675
16
17 qe = (1.6*10**-19)
18
19 # coordinates - Zn:0-2, O1:3, O2:4-9, C1:10-12
20 # gives distance - taken from rosnowCode
21 def magnitude(r, rspace):
22     # 3D distance formula
23     return np.sqrt((r[0]-rspace[0])**2+(r[1]-rspace[1])**2+(r[2]-
24         rspace[2])**2)
25
26 def beta(C,T):
27     return 1/(C*T)
28
29 # kB units in J*K^-1
30 kB = (1.38*10**-23)
31 # in units of J*s
32 hbar = (1.054*10**-34)
33
```

```

32 def deBroglieCoeff(isotope,T):
33     return ((6*isotope.mass)/(beta(kB,T)*hbar**2))/(10**20)
34
35 def secondDeriv(f,x,dx):
36     def firstDeriv(xx):
37         return derivative(f,xx,dx)
38     return derivative(firstDeriv, x, dx)
39
40 #print(deBroglieCoeff(at.hydrogen,300))
41
42 def f(x):
43     return x**2
44
45 #print(secondDeriv(f,2,0.1))

```

## 6.1.4 realisticPotentialModel.py

```

1 import copy as cp
2 import numpy as np
3 import utilities as uT
4 import time
5 import atom as at
6
7 # the electrostatic potential is dependent Electric field but
   electric field is a vector so we use this equation to get the
   vector components
8 def correctionFactors(HAtom, AAtom, R):
9     X = (HAtom.x-AAtom.x)
10    #print("x", X)
11    #print("R", R)
12    Y = (HAtom.y-AAtom.y)
13    Z = (HAtom.z-AAtom.z)
14
15    xCorrection = X/R
16    yCorrection = Y/R
17    zCorrection = Z/R

```

```

18
19     return(xCorrection, yCorrection, zCorrection)
20
21 #calculate the electric field (used in electrostatic potential
    equation). Note that the electric field must be calculated over
    all atoms so that certain parts cancel out (dealing with vectors
    here)
22 def Efield(isotope, atoms):
23     Ex = 0
24     Ey = 0
25     Ez = 0
26     for atom in atoms:
27         R = atom.magnitude(isotope)
28         correctionFactor = correctionFactors(isotope, atom, R)
29         chargeFactor = atom.charge/(R**2)
30         Ex += chargeFactor*correctionFactor[0]
31         Ey += chargeFactor*correctionFactor[1]
32         Ez += chargeFactor*correctionFactor[2]
33     print(Ex, Ey, Ez)
34     # print("efield",Ex**2+Ey**2+Ez**2)
35     # print("R", R)
36     return (Ex**2+Ey**2+Ez**2)
37
38
39 # Gives electrostatic part of potential (Upol)
40 def Upol(isotope, atoms):
41     return (-((1/uT.kB)*(10**10)*uT.k*((uT.qe)**2)*(0.675/2)*Efield(
    isotope, atoms)))
42
43 # Lennard Jones component of potential
44 def Ulj(isotope, atoms):
45     Ulj = 0
46     for atom in atoms:
47         R = atom.magnitude(isotope)
48         epsilon = atom.jointEpsilon(isotope)
49         sigma = atom.jointSigma(isotope)

```

```

50         Ulj += 4*epsilon*((sigma/R)**12-(sigma/R)**6)
51     # print("ulj", Ulj)
52     return Ulj
53
54 # full potential (sum of Upol and Ulj) - depending on how you call
    this function you can include only certain parts of the potential
    (use "Upol" for just Upol, "Ulj" for just Ulj, and "U" for sum
    of both)
55 def U(potentialType, isotope, atoms):
56     if potentialType == "Upol":
57         return Upol(isotope, atoms)
58     elif potentialType == "Ulj":
59         return Ulj(isotope, atoms)
60     elif potentialType == "U":
61         #print("fullU", Ulj(isotope, atoms) + Upol(isotope, atoms))
62         return Ulj(isotope, atoms) + Upol(isotope, atoms)
63
64
65 # feynmann hibbs potential - computes the average of the classical
    potential weighted by a gaussian, sums over triple loop takes a
    very long time to compute for larger grids
66 def UFH(potentialType, xArray, yArray, zArray, atoms, source, T):
67     u_sum = 0
68     normalizationFactor = 0
69     isotope = cp.copy(source)
70     for gridZ in zArray:
71         for gridY in yArray:
72             for gridX in xArray:
73                 isotope.setPoint(gridX, gridY, gridZ)
74                 u_classical = U(potentialType, isotope, atoms)
75                 decayFactor = np.exp(-uT.deBroglieCoeff(isotope,T)*
isotope.magnitude(source))
76                 u_sum += u_classical*decayFactor
77                 normalizationFactor += decayFactor
78     return (u_sum/normalizationFactor)
79

```



```

80 # an attempt at vectorizing the UFH function, doesn't really make
    the timing any less
81 # def UFH(potentialType, xArray, yArray, zArray, atoms, source, T):
82 #     u_sum = 0
83 #     normalizationFactor = 0
84 #     isotope = cp.copy(source)
85 #     for x, y, z in it.product(xArray, yArray, zArray):
86 #         isotope.setPoint(x, y, z)
87 #         u_classical = U(potentialType, isotope, atoms)
88 #         decayFactor = np.exp(-uT.deBroglieCoeff(isotope,T)*isotope
    .magnitude(source))
89 #         u_sum += u_classical*decayFactor
90 #         normalizationFactor += decayFactor
91 #     return u_sum/normalizationFactor
92
93 #Uses potential energy equations to compute potential energy along
    the Z axis, source is the isotope (d2 or h2), potentialType is "U
    ", "Ulj", or "Upol", atoms is a list of atom class definitions (
    see atom.py), and quantump is if we are using the feynmann hibbs
    or not (ie: do we just want classical potential or do we want to
    try to approximate quantum effects with FH)
94 def generate2DPotentialData(potentialType, xArray, yArray, zArray,
    atoms, source, quantump, T):
95     #time is just included to see if different alterations shortened
        F-H time at all
96     start = time.perf_counter()
97     minPotential = float("inf")
98     zval_at_minPotential = 0
99     potentials = []
100    # loads in isotope object
101    isotope = cp.copy(source)
102    for val in zArray:
103        print(val)
104        # sets point of isotope to look at all points along z axis
105        isotope.setPoint(0, 0, val)
106        if quantump:

```

```

107         u_sum =UFH(potentialType, xArray, yArray, zArray, atoms,
isotope, T)
108     else:
109         # classssical version
110         u_sum = U(potentialType, isotope, atoms)
111         if u_sum < minPotential:
112             minPotential = u_sum
113             zval_at_minPotential = val
114         potentials.append(u_sum)
115     end = time.perf_counter()
116     elapsedTime = (end-start)*(1/(10**3))
117     return (potentials, zval_at_minPotential, minPotential,
elapsedTime)
118
119 #generates potential energy along X axis along a slice at the z
value where potential is minimum
120 def generate2DXPotentialData(potentialType, xArray, yArray, zArray,
atoms, source, quantump, T,zval_at_min):
121     start = time.perf_counter()
122     minPotential = float("inf")
123     xval_at_minPotential = 0
124     potentials = []
125     # loads in isotope object
126     isotope = cp.copy(source)
127     for val in xArray:
128         # sets point of isotope to look at all points along z axis
129         isotope.setPoint(val, 0, zval_at_min)
130         if quantump:
131             u_sum =UFH(potentialType, xArray, yArray, zArray, atoms,
isotope, T)
132         else:
133             # classssical version
134             u_sum = U(potentialType, isotope, atoms)
135         if u_sum < minPotential:
136             minPotential = u_sum
137             xval_at_minPotential = val

```

```

138     potentials.append(u_sum)
139 end = time.perf_counter()
140 elapsedTime = (end-start)*(1/(10**3))
141 return (potentials, xval_at_minPotential, minPotential,
142        elapsedTime)
142
143 #generates potential energy along Y axis along a slice at the z
144   value where potential is minimum
144 def generate2DYPotentialData(potentialType, xArray, yArray, zArray,
145   atoms, source, quantump, T,zval_at_min):
145     start = time.perf_counter()
146     minPotential = float("inf")
147     yval_at_minPotential = 0
148     potentials = []
149     # loads in isotope object
150     isotope = cp.copy(source)
151     for val in yArray:
152         # sets point of isotope to look at all points along z axis
153         isotope.setPoint(0, val, zval_at_min)
154         if quantump:
155             u_sum = UFH(potentialType, xArray, yArray, zArray, atoms
156   , isotope, T)
157         else:
158             # classsical version
159             u_sum = U(potentialType, isotope, atoms)
160             if u_sum < minPotential:
161                 minPotential = u_sum
162                 yval_at_minPotential = val
163                 potentials.append(u_sum)
164     end = time.perf_counter()
165     elapsedTime = (end-start)*(1/(10**3))
166     return (potentials, yval_at_minPotential, minPotential,
167           elapsedTime)
166
167 #Gives 3D potential at Z minimum slice.  takes a very long time to
168   run

```

```

168 def generate3DPotentialData(potentialType, xArray, yArray, zArray,
    zval, atoms, source, quantump, T):
169     start = time.perf_counter()
170     potentials = []
171     minPotential = float("inf")
172     yval_at_minPotential = 0
173     xval_at_minPotential = 0
174     isotope = cp.copy(source)
175     for yval in yArray:
176         row_potentials = []
177         for xval in xArray:
178             isotope.setPoint(xval, yval, zval)
179             if quantump:
180                 u_sum = UFH(potentialType, xArray, yArray, zArray,
    atoms, isotope, T)
181             else:
182                 # classical version
183                 u_sum = U(potentialType, isotope, atoms)
184                 u_sum = min(u_sum, 4000)
185                 if u_sum < minPotential:
186                     minPotential = u_sum
187                     yval_at_minPotential = yval
188                     xval_at_minPotential = xval
189                     row_potentials.append(u_sum)
190                 potentials.append(row_potentials)
191     end = time.perf_counter()
192     elapsedTime = (end-start)*(1/(10**3))
193     return (potentials, xval_at_minPotential, yval_at_minPotential,
    minPotential, elapsedTime)

```

### 6.1.5 realisticPotentialGraph.py

```

1 import atom as at
2 import realisticPotentialModel as pm
3 import numpy as np
4 import matplotlib.pyplot as plt

```

```

5 from matplotlib import cm
6
7 def UvZPlot(potentialTypes, atoms, T):
8     z = np.linspace(1.8, 5)
9     x = np.linspace(-3, 3)
10    y = np.linspace(-3, 3)
11    minimums = []
12
13    # Default size is about 6.5 x 4.75 (w x h)
14    plt.figure(figsize=(8.125, 5.937))
15    for pType in potentialTypes:
16
17        def UzClassical(z):
18            pots, zval_at_min, minPotential, elapsedTime = pm.
generate2DPotentialData(pType, x, y, z, atoms, at.hydrogen, False
, T)
19            minimums.append((zval_at_min, minPotential, "Classical "
+ pType))
20            return pots
21
22        def UzHFH(z):
23            pots, zval_at_min, minPotential, elapsedTime = pm.
generate2DPotentialData(pType, x, y, z, atoms, at.hydrogen, True,
T)
24            minimums.append((zval_at_min, minPotential, "H2 " +
pType))
25            return pots
26
27        def UzDFH(z):
28            pots, zval_at_min, minPotential, elapsedTime = pm.
generate2DPotentialData(pType, x, y, z, atoms, at.deuterium, True
, T)
29            minimums.append((zval_at_min, minPotential, "D2 " +
pType))
30            return pots
31

```

```

32     plt.plot(z, UzClassical(z), color="black", label="Classical
" + pType)
33     #plt.plot(z, UzHFH(z), color="red", label="H2 " + pType)
34     #plt.plot(z, UzDFH(z), color="green", label="D2 " + pType)
35     plt.xlabel("z axis distance (angstroms)")
36     plt.ylabel("Potential (K)")
37     lastMinIndex = len(minimums) - 1
38     minText = ""
39     for index in range(lastMinIndex + 1):
40         minval = minimums[index]
41         minText += minval[2] + " Minimum: " + str(round(minval[0],3)
) + ", " + str(round(minval[1]))
42         if index != lastMinIndex:
43             minText += "\n"
44     plt.text(0.02, .985, minText, transform=plt.gcf().transFigure,
va='top', bbox=dict(facecolor='red', alpha=0.5))
45     plt.legend()
46     plt.show()
47
48
49 def UvXPlot(potentialTypes, atoms, T, zmin):
50     z = np.linspace(3, 5)
51     x = np.linspace(-1.5, 1)
52     y = np.linspace(-3, 3)
53     minimums = []
54
55     # Default size is about 6.5 x 4.75 (w x h)
56     plt.figure(figsize=(8.125, 5.937))
57     for pType in potentialTypes:
58
59         def UxClassical(x):
60             pots, xval_at_min, minPotential, elapsedTime = pm.
generate2DXPotentialData(pType, x, y, z, atoms, at.hydrogen,
False, T, zmin)
61             minimums.append((xval_at_min, minPotential, "Classical "
+ pType))

```

```

62         return pots
63
64     def UxHFH(x):
65         pots, xval_at_min, minPotential, elapsedTime = pm.
generate2DXPotentialData(pType, x, y, z, atoms, at.hydrogen, True
, T, zmin)
66         minimums.append((xval_at_min, minPotential, "H2 " +
pType))
67         return pots
68
69     def UxDFH(x):
70         pots, xval_at_min, minPotential, elapsedTime = pm.
generate2DXPotentialData(pType, x, y, z, atoms, at.deuterium,
True, T, zmin)
71         minimums.append((xval_at_min, minPotential, "D2 " +
pType))
72         return pots
73
74     plt.plot(x, UxClassical(x), color="black", label="Classical
" + pType)
75     #plt.plot(x, UxHFH(x), color="red", label="H2 " + pType)
76     #plt.plot(x, UxDFH(x), color="green", label="D2 " + pType)
77     plt.xlabel("x axis distance (angstroms)")
78     plt.ylabel("Potential (K)")
79     lastMinIndex = len(minimums) - 1
80     minText = ""
81     for index in range(lastMinIndex + 1):
82         minval = minimums[index]
83         minText += minval[2] + " Minimum: " + str(round(minval[0],3)
) + ", " + str(round(minval[1]))
84         if index != lastMinIndex:
85             minText += "\n"
86     plt.text(0.02, .985, minText, transform=plt.gcf().transFigure,
va='top', bbox=dict(facecolor='red', alpha=0.5))
87     plt.legend()
88     plt.show()

```

```

89
90
91 def UvYPlot(potentialTypes, atoms, T, zmin):
92     z = np.linspace(3, 5)
93     x = np.linspace(-1, 1)
94     y = np.linspace(-1, 1)
95     minimums = []
96
97     # Default size is about 6.5 x 4.75 (w x h)
98     plt.figure(figsize=(8.125, 5.937))
99     for pType in potentialTypes:
100
101         def UyClassical(y):
102             pots, yval_at_min, minPotential, elapsedTime = pm.
generate2DYPotentialData(pType, x, y, z, atoms, at.hydrogen,
False, T, zmin)
103             minimums.append((yval_at_min, minPotential, "Classical "
+ pType))
104             return pots
105
106         def UyHFH(y):
107             pots, yval_at_min, minPotential, elapsedTime = pm.
generate2DYPotentialData(pType, x, y, z, atoms, at.hydrogen, True
, T, zmin)
108             minimums.append((yval_at_min, minPotential, "H2 " +
pType))
109             return pots
110
111         def UyDFH(y):
112             pots, yval_at_min, minPotential, elapsedTime = pm.
generate2DYPotentialData(pType, x, y, z, atoms, at.deuterium,
True, T, zmin)
113             minimums.append((yval_at_min, minPotential, "D2 " +
pType))
114             return pots
115

```



```

116     plt.plot(y, UyClassical(y), color="black", label="Classical
    " + pType)
117     #plt.plot(y, UyHFH(z), color="red", label="H2 " + pType)
118     #plt.plot(y, UyDFH(z), color="green", label="D2 " + pType)
119 plt.xlabel("y axis distance (angstroms)")
120 plt.ylabel("Potential (K)")
121 lastMinIndex = len(minimums) - 1
122 minText = ""
123 for index in range(lastMinIndex + 1):
124     minval = minimums[index]
125     minText += minval[2] + " Minimum: " + str(round(minval[0],3)
    ) + ", " + str(round(minval[1]))
126     if index != lastMinIndex:
127         minText += "\n"
128 plt.text(0.02, .985, minText, transform=plt.gcf().transFigure,
    va='top', bbox=dict(facecolor='red', alpha=0.5))
129 plt.legend()
130 plt.show()
131
132
133 def UvZPlot3D(potentialType, source, atoms, zval, quantump, T):
134     z = np.linspace(2.5, 6, 5)
135     x = np.linspace(-3, 3, 5)
136     y = np.linspace(-3, 3, 5)
137     X, Y = np.meshgrid(x, y)
138     potentials, xval_at_minPotential, yval_at_minPotential,
    minPotential, elapsedTime = pm.generate3DPotentialData(
    potentialType, x, y, z, zval, atoms, source, quantump, T)
139     # Default size is about 6.5 x 4.75 (w x h)
140     fig = plt.figure(figsize=(8.125, 5.937))
141     ax = fig.gca(projection='3d')
142     ax.plot_surface(X, Y, np.array(potentials), rstride=1, cstride
    =1, cmap=cm.gist_rainbow)
143     ax.set_xlabel("x axis distance (angstroms)")
144     ax.set_ylabel("y axis distance (angstroms)")
145     ax.set_zlabel(potentialType + " (K)")

```

```

146     ax.text2D(0, 1, "U min: " + str(round(minPotential,3)) + "\
nPoint: (" + str(round(xval_at_minPotential, 3)) + "," + str(
round(yval_at_minPotential, 3)) + "," + str(round(zval,3)) + ")",
transform=ax.transAxes, bbox=dict(facecolor='red', alpha=0.5))
147     plt.show()
148
149
150 atoms = at.atoms
151 hydrogen = at.hydrogen
152 deuterium = at.deuterium

```

## 6.1.6 CuUtilities.py

```

1 # utility functions for Cu Potential Program
2 import numpy as np
3
4 def readinPotential(file):
5     file1 = open(file, "r")
6     distance = []
7     potential = []
8     numlist = file1.read().split()
9     for i in range(len(numlist)):
10         numlist[i] = float(numlist[i])
11         if i%2 == 0:
12             distance.append(numlist[i])
13         else:
14             numlist[i] = numlist[i]/.008314
15             potential.append(numlist[i])
16     file1.close()
17     return distance,potential
18
19 def readinFile(file, potp):
20     file1 = open(file, "r")
21     newlst = []
22     numList = file1.read().split()
23     for i in range(len(numList)):

```

```

24     numList[i] = float(numList[i])
25     if potp == True:
26         numList[i] = numList[i]/.008314
27     newlst.append(numList[i])
28 file1.close()
29 return newlst
30
31 def findMinimum(dist, pot):
32     minimumPot = 100000
33     minimumDist = 0
34     for i in range(len(pot)):
35         if pot[i] < minimumPot:
36             minimumPot = pot[i]
37             minimumDist = dist[i]
38     return minimumDist, minimumPot
39
40 def sumLstSection(lst, start, end):
41     sum = 0
42     for i in range(start, end+1):
43         sum += lst[i]
44     return sum
45
46 def squaredlist(lst):
47     lstSquared = []
48     for i in range(len(lst)):
49         lstSquared.append(lst[i]**2)
50     return lstSquared
51
52
53 def makeFile(fileName, Lst):
54     np.savetxt(fileName, Lst, delimiter=", ")
55
56
57 def normLst(lst, area):
58     for i in range(len(lst)):
59         lst[i] = lst[i]/area

```

## 6.1.7 SE.py

```
1 import utilities as uT
2 import matplotlib.pyplot as plt
3 import atom as at
4 import CuUtilities as ct
5 import numpy as np
6 import realisticPotentialModel as rm
7
8 UXclassical = rm.generate2DXPotentialData("U", np.linspace(-1.5,1),
      np.linspace(-1,1), np.linspace(2.4,6,150), at.atoms, at.hydrogen,
      False, 77, 3.04)[0]
9 UYclassical = rm.generate2DYPotentialData("U", np.linspace(-1,1), np
      .linspace(-1,1), np.linspace(2.4,6,150), at.atoms, at.hydrogen,
      False, 77, 3.04)[0]
10 xdist = np.linspace(-1.5,1)
11
12 # SE function to find Energy levels and wavefunction
13
14 def SE(potential, psi1, psi2, coordArray, mass, E):
15     psi = []
16     psi.append(psi1)
17     psi.append(psi2)
18     for i in range(1, len(coordArray)-1):
19         deltaCoordSquared = (coordArray[i]-coordArray[i-1])**2
20         convertedCoeff = 2*mass/(uT.hbar**2)*uT.kB*(1/10**20)
21         psiNext = (convertedCoeff*deltaCoordSquared*(potential[i]-E
+2)*psi[i]-psi[i-1]
22         psi.append(psiNext)
23     #print(psi)
24     return psi
25
26 #print(SE(ct.readFile("potential01.txt", False), 0, 0.001, ct.
      readFile("distance01.txt",False),at.hydrogen.mass, -2101.27875)
      )
27 #print(SE(ct.readFile("potential01.txt", False), 0, 0.001, ct.
```

```

    readinFile("distance01.txt",False),at.hydrogen.mass, -907.6))
28
29 def wagDog(potential, psi1, psi2, coordArray, mass, E):
30     def psi(coordArray):
31         return SE(potential, psi1, psi2, coordArray, mass, E)
32     plt.plot(coordArray, psi(coordArray))
33     plt.show()
34
35 def MOF5Main(E):
36     wagDog(UXclassical, 0, 0.001,xdist, at.deuterium.mass, E)
37
38 def CuMain(E):
39     coordArray = ct.readinFile("distance01.txt", False)
40     potArray = ct.readinFile("potential01.txt", False)
41     wagDog(potArray, 0, 0.001, coordArray, at.deuterium.mass, E)
42
43
44 MOF5Main(-917)
45
46 def findFlipPoint(potential, psi1, psi2, coordArray, mass, Estart,
    Eend, step):
47     psi = SE(potential, psi1, psi2, coordArray, mass, Estart)
48     if psi[len(psi)-1] > 0:
49         positive = True
50     else:
51         positive = False
52     Evalue = Estart+step
53     while Evalue <= Eend:
54         psiNew = SE(potential, psi1, psi2, coordArray, mass, Evalue)
55         if positive:
56             if psiNew[len(psiNew)-1] > 0:
57                 Evalue += step
58             else:
59                 break
60         else:
61             if psiNew[len(psiNew)-1] < 0:

```

```

62         Evaluate += step
63     else:
64         break
65     return Evaluate

```

## 6.2 generateWavefunction.py

```

1  import SE as se
2  import CuUtilities as ct
3  import atom as at
4
5  #takes a given potential FILE, distance FILE, isotope (h2 or d2),
6  GSE (ground state energy), and FEE (first excited state energy)
7  and returns lists containing normalized psi and psi^2 for GS and
8  FES
9
10 def generateWavefunction(potentialFile, distanceFile, isotope, GSE,
11 FEE):
12
13     potential = ct.readinFile(potentialFile, False)
14     distance = ct.readinFile(distanceFile, False)
15     psiGS = se.SE(potential, 0, 0.001, distance, isotope.mass, GSE)
16     psiFE = se.SE(potential, 0, 0.001, distance, isotope.mass, FEE)
17     probGS = ct.squaredlist(psiGS)
18     probFE = ct.squaredlist(psiFE)
19     areaGS = sum(probGS[0:98])
20     areaFE = sum(probFE[0:98])
21     #print(areaGS)
22     #print(areaFE)
23     ct.normLst(psiGS, areaGS)
24     ct.normLst(psiFE, areaFE)
25     ct.normLst(probGS, areaGS)
26     ct.normLst(probFE, areaFE)
27     #print(sum(probGS))
28     #print(sum(probFE))
29     return (psiGS, probGS, psiFE, probFE)

```

```

25 #calls previous function and saves lists as files (useful for
    plotting in Igor to have these saved as text file)
26 def main():
27     #psiGSH,probGSH, psiFEH, probFEH = generateWavefunction("
    potential01.txt", "distance01.txt", at.deuterium,
    -2312.186611170497, -1100.063)
28     psiGSH,probGSH, psiFEH, probFEH = generateWavefunction("
    UClassical.txt", "zClassical.txt", at.hydrogen, -1071.29, -948.52)
29     ct.makeFile("M5psiGSH.txt", psiGSH)
30     ct.makeFile("M5probGSH.txt", probGSH)
31     ct.makeFile("M5psiFEH.txt", psiFEH)
32     ct.makeFile("M5probFEH.txt", probFEH)
33
34 main()
35
36 #calculates area under curve to see percentages in each section of
    the probability densities
37 def bumpPercentage(potentialFile, distanceFile, isotope, GSE, FEE,
    startPoint, endPoint):
38     probFE = generateWavefunction(potentialFile, distanceFile,
    isotope, GSE, FEE)[3]
39     bumpPercent = ct.sumLstSection(probFE, startPoint, endPoint)
40     return bumpPercent

```





# Bibliography

- [1] Brandon Barnett, Gregory M Su, Henry ZH Jiang, Romit Chakraborty, Didier Banyeretse, Tyler Hartman, Madison Martinez, Benjamin A Trump, Jacob Tarver, Matthew Dods, et al. Observation of an intermediate to  $h_2$  binding in a metal–organic framework. 2020.
- [2] Jonathan L Belof, Abraham C Stern, and Brian Space. A predictive model of hydrogen sorption for metal-organic materials. *The Journal of Physical Chemistry C*, 113(21):9316–9320, 2009.
- [3] CMB.Contact@noaa.gov. Global climate change indicators.
- [4] Stephen A FitzGerald, Kelty Allen, Patrick Landerman, J Hopkins, J Matters, R Myers, and Jesse LC Rowsell. Quantum dynamics of adsorbed  $h_2$  in the microporous framework mof-5 analyzed using diffuse reflectance infrared spectroscopy. *Physical Review B*, 77(22):224301, 2008.
- [5] Stephen A FitzGerald, Daniel Mukasa, Katharine H Rigdon, Naiyuan Zhang, and Brandon R Barnett. Hydrogen isotope separation within the metal–organic framework cu (i)-mfu-4 l. *The Journal of Physical Chemistry C*, 123(50):30427–30433, 2019.
- [6] Stephen A FitzGerald, Jocienne N Nelson, Elizabeth Gilmour, and Jesse LC Rowsell. Infrared overtone spectroscopy of adsorbed hydrogen in mof-5. *Journal of Molecular Spectroscopy*, 307:20–26, 2015.
- [7] Stephen A FitzGerald, Christopher J Pierce, Jesse LC Rowsell, Eric D Bloch, and Jarad A Mason. Highly selective quantum sieving of  $d_2$  from  $h_2$  by a metal–organic framework as determined by gas manometry and infrared spectroscopy. *Journal of the American Chemical Society*, 135(25):9458–9464, 2013.
- [8] Irving Friedman. Deuterium content of natural waters and other substances. *Geochimica et cosmochimica acta*, 4(1-2):89–103, 1953.
- [9] Steven S Kaye, Anne Dailly, Omar M Yaghi, and Jeffrey R Long. Impact of preparation and handling on the hydrogen storage properties of  $zn_4o$  (1, 4-benzenedicarboxylate) 3 (mof-5). *Journal of the American Chemical Society*, 129(46):14176–14177, 2007.

- [10] Mitsuru Kikuchi. *Frontiers in fusion research: physics and fusion*. Springer Science & Business Media, 2011.
- [11] AV Anil Kumar and Suresh K Bhatia. Quantum effect induced reverse kinetic molecular sieving in microporous materials. *Physical review letters*, 95(24):245901, 2005.
- [12] Yun Liu, Houria Kabbour, Craig M Brown, Dan A Neumann, and Channing C Ahn. Increasing the density of adsorbed hydrogen with coordinatively unsaturated metal centers in metal-organic frameworks. *Langmuir*, 24(9):4772–4777, 2008.
- [13] Tony Pham, Katherine A Forrest, Keith McLaughlin, Juergen Eckert, and Brian Space. Capturing the h<sub>2</sub>-metal interaction in mg-mof-74 using classical polarization. *The Journal of Physical Chemistry C*, 118(39):22683–22690, 2014.
- [14] Anthony K Rappé, Carla J Casewit, KS Colwell, William A Goddard III, and W Mason Skiff. Uff, a full periodic table force field for molecular mechanics and molecular dynamics simulations. *Journal of the American chemical society*, 114(25):10024–10035, 1992.
- [15] Kinori Rosnow. Solving for eigenvalues and eigenvectors of a mof-5 lennard-jones potential. 2017.
- [16] Jesse LC Rowsell, Elinor C Spencer, Juergen Eckert, Judith AK Howard, and Omar M Yaghi. Gas adsorption sites in a large-pore metal-organic framework. *Science*, 309(5739):1350–1354, 2005.
- [17] Jesse LC Rowsell and Omar M Yaghi. Metal-organic frameworks: a new class of porous materials. *Microporous and mesoporous materials*, 73(1-2):3–14, 2004.
- [18] Isaac F Silvera. The solid molecular hydrogens in the condensed phase: Fundamentals and static properties. *Reviews of Modern Physics*, 52(2):393, 1980.
- [19] Elinor C Spencer, Judith AK Howard, Garry J McIntyre, Jesse LC Rowsell, and Omar M Yaghi. Determination of the hydrogen absorption sites in zn<sub>4</sub>o(1,4-benzenedicarboxylate) by single crystal neutron diffraction. *Chemical communications*, (3):278–280, 2006.
- [20] T Yildirim and MR Hartman. Direct observation of hydrogen adsorption sites and nanocage formation in metal-organic frameworks. *Physical review letters*, 95(21):215504, 2005.