COMPUTATIONAL STUDIES OF GASES ADSORBED ON GRAPHENE-LIKE MATERIALS

A Dissertation

Presented to

the Faculty of the Graduate School

at the University of Missouri-Columbia

In Partial Fulfillment

Of the Requirements for the Degree

Doctor of Philosophy

By

Todd Lombardi

Dr. Carlos Wexler, Dissertation Supervisor

May 2022

The undersigned appointed by the Dean of the Graduate School, have examined the dissertation entitled

COMPUTATIONAL STUDIES OF GASES ADSORBED ON GRAPHENE-LIKE MATERIALS

Presented by Todd Lombardi,

a candidate for the degree of doctor of philosophy,

and hereby certify that, in their opinion, it is worthy of acceptance.

Professor Gary Baker

Professor Guang Bian

Professor Carsten Ullrich

Professor Wouter Montfrooij

Professor Carlos Wexler

DEDICATION

To my love, Bo, who was present to support me at every low and celebrate every high during my graduate studies at Mizzou.

To my parents who gave me enough persistence to finish graduate school.

To the St. Francis Catholic Worker Community, who embraced me and made my time in Columbia colorful and memorable.

ACKNOWLEDGEMENTS

I would first like to acknowledge my committee for their time and efforts: Dr. Gary Baker, Dr. Guang Bian, Dr. Wouter Montfrooij, Dr. Carsten Ullrich, and Dr. Carlos Wexler.

Part of the research presented in this dissertation is funded by NSF Award IIP-2044726.

I would like to acknowledge the following visiting scholars: Dr. Alberto Albesa for sharing his ideas to help complete and publish part of this research, Dr. Gonzalo dos Santos for his contributions that led to publishing, and Dr. Michael Roth for his valuable input and encouragement.

Additionally, I would like to thank the undergraduate researchers who have joined the lab over the years for their assistance: Joshua Miles, Corey Valleroy, and Zachary Valleroy.

And thank you to visiting graduate students Mahmoud Attia and Kenedy Tabah for their contributions in collecting data.

Obtaining and maintaining the NSF grant would not have been possible without Dr. Fateme Rezaei or Brett Maland. Their guidance and aid in completing the Innovation Corps NSF Workshop was critical to the grant's success which allowed us to continue research.

I am happy to give a special thank you to my advisor Dr. Carlos Wexler for introducing me to the world of computational physics and providing me the guidance and tools to complete my research as well as introducing me to almost all the previously acknowledged people.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Abbreviation	Meaning	Page
2D	Two-dimensional	67
AC	Activated Carbon	1
ANG	Adsorbed Natural Gas	2
BHM	Bose Hubbard Model	69
CA	Graphene carbon atoms	49
CA-GOF	Covalent Angled Graphene Oxide Framework	15
CC	Coupled Cluster	71
CH_4	Methane	1
CNG	Compressed Natural Gas	3
CO_2	Carbon-dioxide	1
COF	Carbon-organic Framework	1
CP-GOF	Covalent Pillared Graphene-oxide Framework	12
DBA	Benzene-1,4-Diboronic Acid	10
DFT	Density Functional Theory	20
GCMC	Grand Canonical Monte Carlo	41
GO	Graphene-oxide	10
GOF	Graphene-oxide Framework	1
H_2S	Hydrogen Sulfide	4
^4He	Helium-4	67
LCFS	Low Carbon Fuel Standard	6
MD	Molecular Dynamics	15
MOF	Metal-organic Framework	1
MP2	2nd order Møller–Plesset perturbation theory	70

MU	University of Missouri	3
N_2	Nitrogen	4
NG	Natural Gas	41
NOAA	National Oceanic and Atmospheric Administration	6
NSF	National Science Foundation	7
OH	Hydroxyl	42
PBC	Periodic Boundary Conditions	42
PME	Particle mesh Ewald	19
QMC	Quantum Monte Carlo	69
SSA	Specific Surface Area	1
TCL	Tool Command Language	15
vdW	van der Waals	67
vdW-GOF	van der Waals Graphene-oxide Framework	12
VMD	Visual Molecular Dyanmics	15

ABSTRACT

Nanoporous activated carbons generate interest for their gas storage and separation potential. Generally, adsorbents are assumed rigid, even though they are formed by feeble quasi-2D flakes of graphene. In 2019, Schaeperkoetter *et al.*[1] observed swelling of graphene oxide frameworks (GOFs) upon supercritical adsorption of various gases. We performed molecular dynamics (MD) simulations of methane and xenon in various models of GOF's with interaction parameters derived from *ab initio* Density Functional Theory. We observe a monotonic increase of the interlayer spacing consistent with experiments only for a model of benzene-1,4-diboronic acid (DBA) molecules bonded covalently to graphene on both sides of the pore at random orientations, establishing the structure of the DBA-GOFs.

Adsorbents are also useful for the separation of gases, e.g., methane and carbon dioxide from organic waste biogas. We performed MD and grand canonical Monte Carlo simulations of the coadsorption of CH_4 and CO_2 in pores of different sizes and surface functionalization. We observe significant selectivity for the adsorption of CO_2 - potentiated by the presence of polar surface groups and determined optimal conditions for gas separation in this system.

Finally, atomically flat graphene allows the emergence of two-dimensional films of weakly adsorbed helium with interesting quantum properties. We performed *ab initio* 2^{nd} order Møller–Plesset calculations with large basis sets of the interaction of 1, 2, and 3 He atoms on graphene-like systems. The interaction parameters are then used in the Bose-Hubbard model, and under certain conditions it is predicted that superfluid or Mott insulating phases can occur.

Chapter 1: Introduction

1.1 Motivation

Carbon-based porous adsorbent materials have a significant potential to store and/or separate the mixture of methane (CH_4) and carbon dioxide (CO_2) that often comes from the decomposition of organic matter in landfills, manure lagoons, and other waste processing facilities. However, the development of these materials and technologies is not yet of sufficient maturity, resulting in less than optimal harnessing of the energy that is released during the decomposition of such organic waste. Why are adsorbents potentially useful? By sequestering these gases into small pores (i.e., acting as *nano sponges*) they can store equivalent quantities of gas at pressures that are substantially lower than an empty tank! Furthermore, because different gases interact with the substrate differently, these adsorbents also offer the chance to selectively adsorb one gas out of a mixture, thus permitting on-system purification of these biogases to suit some need, e.g., sequestration of CO_2. Carbon nanostructures vastly vary in their manifestations which include amorphous activated carbons (AC)[2–4], metal organic frameworks (MOFs)[5,6], covalent organic frameworks (COFs)[7,8] and graphene oxide frameworks (GOFs)[9,10]. Carbon nanostructures typically have sizeable specific surface areas (SSA) that correlate to high storage capacities due to the attraction of various gas molecules to those surfaces; this being the reason why these gases are packed much more tightly than they would be in empty space (in fact, the adsorbed film has gas densities comparable to the same substance in the liquid state[11,12] even though the gas is at temperatures much higher than its critical temperature!). Carbon materials are

particularly useful because these nanostructures' SSAs are often close to the maximum limit, i.e., to that of graphene (2,600 m^2/g), and for some materials the pore sizes are close to 1 nm, where there is some overlap in the attraction of the gas molecules to *both* surfaces. One important factor often ignored, however, is that the adsorption potential has to be "just right": too weak and there is insufficient gas storage, too strong and it is difficult to get the gas out afterwards. Bhatia and Myers[13] determined that at room temperature an ~ 18kJ/mol adsorption enthalpy is optimal, and Kutcha *et al.*[14] found homogeneous adsorbents are optimal. For our purposes, CH_4 and CO_2 are the main molecules of interest, and their enthalpies of adsorption are close to (or a bit higher than) the optimal value, as will be shown later.

Specifically, we are interested in using ACs to capture CH_4 and CO_2 in the form of adsorbed natural gas (ANG). AC is not too different from charcoal, created in a similar process but with a few differences. Charcoal is made at temperatures of ~300 °C with little or no pressure control, whereas ACs are made in environments with more controlled pressures and temperatures (as high as 1200 °C), and sometimes in inert gases for further control. However, even with this control, typical ACs have broad pore size distributions, which inhibits to some degree the understanding of the physics involved at the atomic level[1]. As we will see later in the dissertation, some materials have allowed finer control over the pore structure, which facilitates comparison of experimental observations with theoretical and computational results.

Models and simulations of ANG are thus important for understanding what is happening at an atomic level inside of the pores. The challenges of accurately modeling the adsorbate-adsorbent interaction and the fine tuning of simulation parameters make

this a continuous process as new types of pores are synthesized by laboratories all over the world. For example, groups at the University of Missouri (MU) have made promising starts creating ANG prototypes[11,15–20]. **Table 1.1** and **Figure 1.1** directly compare the performance of ANG and CNG tanks. The ANG tank can be seen to follow a typical adsorption isotherm, filling up faster and with more natural gas than a classic Compressed Natural Gas (CNG) tank even at room temperature[11,15–20]. For a given storage capacity, due to the lower operational pressure and/or total required volume, ANG tanks are considerably lighter making them even more attractive in terms of fuel efficiency[11,15–20]. Using ACs for ANG means one could create gas tanks with smaller external volumes, lighter tank masses, and lower pressures[21]. ANG can achieve comparable energy densities to mainstream CNG but at lower pressures and at room temperature[22] meaning that thinner walls can be used safely for the tank. Compared to the bulky cylinders used in vehicles today, an ANG tank could fit more discretely and increase cargo capacity in gas-powered vehicles. As of now, the most prominent ANG tank producer in the USA is *Ingevity*[23] which has also made news recently through expansion of its ANG refueling network[24].

Table 1.1 – The ANG tank's specifications are from MU's ANG tanks as described in [11,15–20].

	Adsorbed Natural Gas	**Compressed Natural Gas**
Capacity (kg)	20	20
Storage Pressure (bar)	35	250
Material	Al 6063-T52	Al 6063-T52
Mass of Tank + Fuel (kg)	172	340
External Volume (L)	190	230

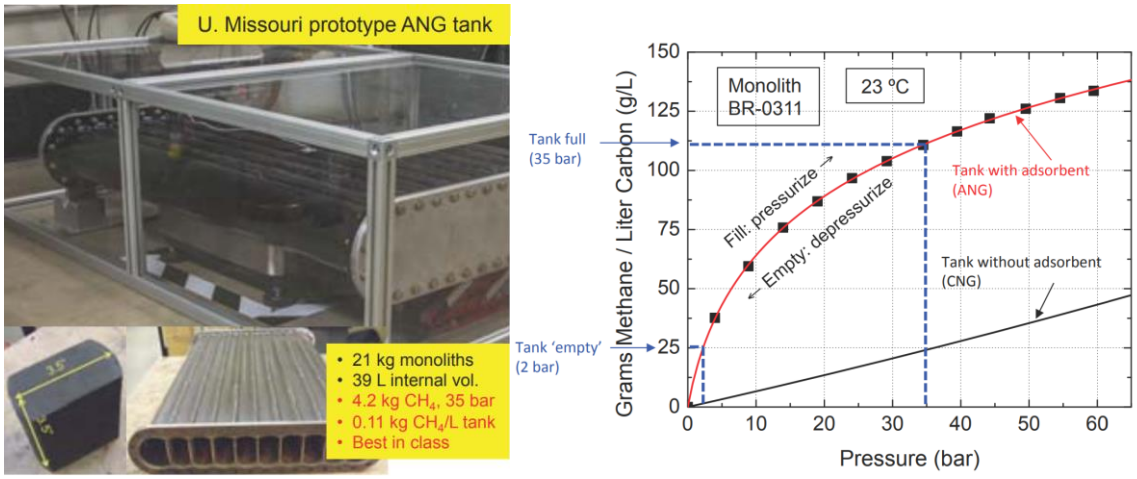

Figure 1.1 – University of Missouri's protype and view of its nanoporous material (left). A comparison of ANG and CNG tanks (right) shows ANG contain significantly more gas at lower pressures[16]. Figures courtesy of Rash *et al.*[16]

It is fair to question why we would study materials that appear to encourage the consumption of CH_4, a hydrocarbon that generates CO_2 when combusted; the answer lies in *renewable* natural gas (but also see below). The U.S. dumps over 80 million metric tons of organic waste into landfills every year[25] and produces approximately 300 million dry metric tons of animal manure[26], all of which decompose emitting biogas - a mixture of 50-75% CH_4, 25-50% CO_2 and traces of hydrogen sulfide (H_2S), nitrogen (N_2), water and siloxanes. Capturing biogas, sequestering CO_2, isolating the CH_4, and using the CH_4 to displace other carbon emitting fuel sources is a vital component to fighting the climate crisis which is projected to increase Earth's sea level and average temperature to unsustainable and costly levels in much of the world[27]. CH_4 harvested from biogas provides carbon neutral energy - in some cases it may even be a carbon sink! In fact, presently for heavy framed vehicles and vessels, electrification (batteries) is still far from competitive, with substantial sacrifices in range and carrying capacity[28]. The capture of

biogas and understanding the technology that does so is now more important than ever before.

Presently, the main sources of biogas are landfills, wastewater treatment plants, and livestock operations, which are integral to our society's functioning and are not going away any time soon. Even a carbon neutral world has capacity for some combustion engines. And in some cases, these engines have no viable substitute. Therefore, wherever it is that combustion engines are found to be necessary for, it would be best if the hydrocarbons being burned were to come from socially vital, renewable sources.

Also note that the primary component of natural gas, CH_4[29], when burned is overall less harmful to humans and the environment compared to other hydrocarbons in use today. First, CH_4 burning does not release any sulfur or nitrogen oxide[30–32] which are primary causes of acid rain[33] and pollutant induced respiratory illnesses in humans[34], respectively. Additionally, CH_4 yields more energy per molecule of CO_2 than coal or petroleum[35,36] making it a less harmful option when considering the climate crisis for which radiative forcing of CO_2 is the smoking gun[37]. **Figure 1.2** shows different examples of where CH_4 uses less energy than its diesel counterparts.

As it seems quite unlikely that a transition away from hydrocarbons will be imminent and complete, they will continue to be burned as a large portion of the US energy consumption based on current trends[38,39]. So, if hydrocarbons with lower CO_2/energy ratios, or in the case of biogas a negative CO_2/energy ratio, are burned, we could buy precious time to delay the worst effects of climate change. Secondly, given the urgency of the climate crisis, burning CH_4 from a renewable source and thereby releasing CO_2 into the atmosphere is less harmful than allowing CH_4 to escape into the atmosphere

(every landfill in the U.S.A. is already required to capture and flare their biogas, but leaks and failures exist). The primary reason is that CH_4's *radiative forcing*, a gas's total energy absorbed per second per square meter of surface area, compared to CO_2 is at least 20 times greater per molecule[40,41]. With recent atmospheric readings, CH_4's radiative forcing is approximately 0.279 W/m^2 per ppm while CO_2 is 0.00506 W/m^2 per ppm with CO_2 having around 200 times more ppm than CH_4 using the National Oceanic and Atmospheric Administration (NOAA) greenhouse gas index[42]. (The low concentration of CH_4 is also due to its natural decomposition in our oxygen-rich atmosphere.) This disparity is not because CH_4 is better at absorbing electromagnetic waves than CO_2; it is because CO_2 has saturated the atmosphere so much that it is absorbing much of its most favored frequencies to the point that excess CO_2 molecules are left with only unfavorable frequencies–known as absorption band saturation[40,43]. CH_4's absorption band on the other hand is empty in comparison, so any CH_4 added to the atmosphere will easily absorb the more abundant, favored frequencies.

And it is during crises that profitable opportunities like the Low Carbon Fuel Standard (LCFS) arise for those with initiative to reap its benefits. LCFS refers to a series of legislation passed in California[44], Oregon[45], and Washington[46] that forces companies of over certain sizes to reduce carbon emissions, pay fines or purchase carbon credits from a regulated marketplace unique to each state. The credits are created by offsets of carbon emissions, most notably the capture and purification of biogases emitted from dairies and landfills. The largest carbon credit market of California is not stable yet, fluctuating between $120-$200 per ton of CO2 equivalent displaced[47]. However, these

programs have motivated a spur in the construction of facilities that capture and purify biogas all across the country, nearly quadrupling in four years[48].

As part of the requirements of our National Science Foundation (NSF) grant, we have intimately studied how our research and that of our partners can be profitable using the LCFS. To accomplish this, we completed the NSF's Innovation Corps (I-Corps), a 7-week workshop where a team of no more than 4 people build a business model around their technology. In our case, we built a business model around the CO_2-CH_4 on-board separating ANG tanks of our partners at the University of Missouri Science & Technology.

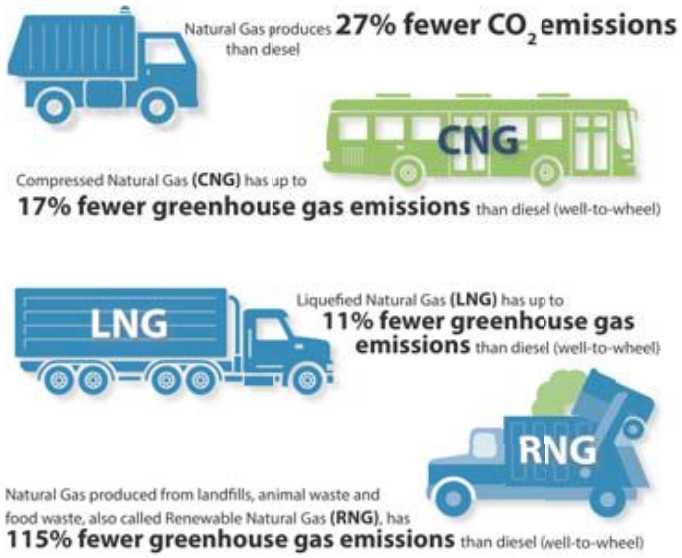

Figure 1.2 - CO2 and greenhouse gas emissions in heavy-duty NG vehicles relative to diesel, with storage as com- pressed natural gas (CNG), liquefied natural gas (LNG), and renewable compressed natural gas (RNG)[49]. Reproduced from Natural Gas Vehicles for America[49].

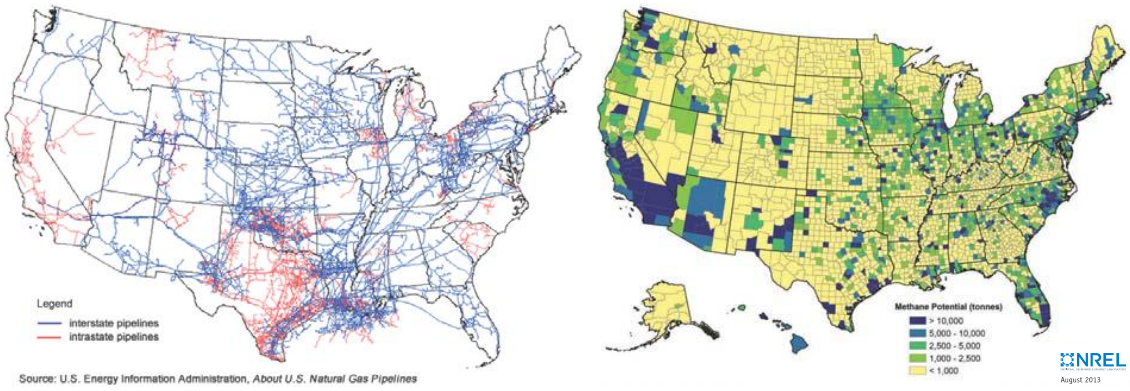

Figure 1.3 - *Left*: Map of the U.S. NG grid[50]. The grid is concentrated where fossil NG is produced, and mostly non- existent where RNG is produced. *Right*: Estimated biomethane generation potential[51].

We initially proposed using the biogas in combustion engine vehicles used at livestock operations far away from pipeline interconnects, with the main selling point being that separating the CH_4 and CO_2 inside the tank would save on costs for upgrading the gas and provide a free substitute for diesel fuel, all this while capturing the CO_2 for later sale. We then tested the business model hypotheses by interviewing 100 people involved in every part of the theoretical business model and used their responses to adjust and develop new business model hypotheses. Ultimately, we found no need for use on-site, but rather a strong desire to inject upgraded (cleaned) biomethane into the pipeline in order to receive LCFS credits. Smaller dairies and hog farms far away from pipelines have a significant problem in that they desire LCFS credits but cannot quite reach the pipelines which cost an estimated $1 million/mile. From making some general observations of **Figure 1.3** which shows the United States' national natural gas pipeline grid and counties by methane yield, we can build a better picture of how some high methane producing counties are far away from natural gas pipelines. So, if our technology can be repurposed for the creation of "virtual pipelines" via low-pressure

trailers that can be brought to injection sites, we believe that there is economic sense in biogas storage technology too.

Biogas's abundance, NG's low CO_2/energy ratio, the critical need to keep CH_4 out the atmosphere, and the possibility for smaller farms to profit add importance to the study of materials and technology that can efficiently store biomethane for purification and consumption.

1.2 Overview

The focus of this dissertation is: **Chapter 2**: the study of a family of monodisperse nanoporous carbons—graphene-oxide framework (GOFs)—and how their structure changes during adsorption of CH_4 and Xe under supercritical conditions, helping determine the details of the structure of these promising materials. **Chapter 3**: extensive simulations of the co-adsorption of CO_2 and CH_4 in activated carbons of varying properties (pore size, surface functionalization with polar groups) to determine optimal conditions for the refinement of biogas. **Chapter 4**: the determination of the He-graphene, and He-He interactions using *ab initio* 2nd Møller–Plesset simulations, which are used in the Bose-Hubbard model to determine whether the adsorbed film is a superfluid, or a Mott insulator.

Chapter 2: Investigation of Structural Changes of Graphene Oxide Frameworks

2.1 Introduction

There are several ideas for how Graphene Oxide Framework (GOFs) pores are supported and kept open by their reactants. In our study we are interested in GOFs formed by benzyl groups. While there are many GOFs pillared with phenyl and benzyl groups[1,9,52–56], there is no agreement on how these acids settle inside the pores and form the resulting structures. The most common model proposed is that pillars of benzyl or phenyl groups bond to the top and bottom surfaces of the pores during synthesis[53–56], see **Figure 2.1-2.2**. Kumar *et al.*[56] and Srinivas *et al.*[55] (**Figure 2.1-2.2**) both experimented on and theorized phenyl and benzyl groups bonded to the top and bottom layers of the pores, while Srinivas also suggested that it is possible only one side of the pore is bonded. Hung *et al.* synthesized and observed pore expansion in amine groups bonding inside of Graphene-oxide (GO) pores, suggesting that these amine groups also can form pillars[53]. Sun *et al.* also experimented creating GOFs with phenyl groups[54], suggesting that the pore stability and spacing comes from the phenyl groups interlocking.

In our study, we focus on the structural properties of a type of GOF formed by using an intercalation procedure originally proposed by Burress *et al.*[9] and reproduced by Mercier *et al.*[52] and Schaeperkoetter *et al.*[1]: the dissolution of benzene-1,4-diboronic acid (DBA) in methanol and then intercalating the DBA into GO which lead to GOFs with specific surface areas (SSA) ≈ 400-1,000 m^2/g, and interlayer distances of ~9 Å (**Figure 2.3**).

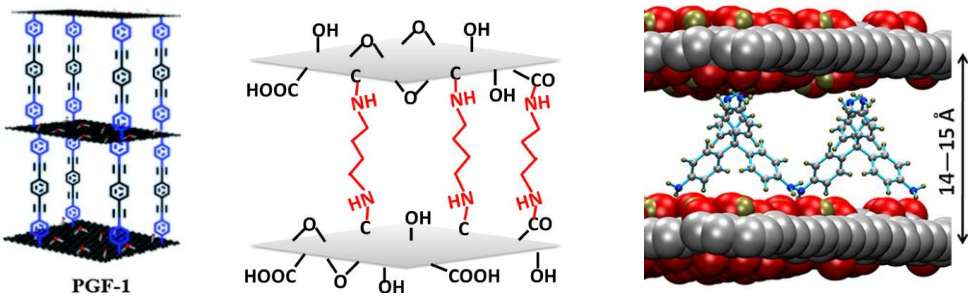

Figure 2.1 – GOF models proposed by Kumar (left)[41], Hung (center)[53], and Sun(right)[54].

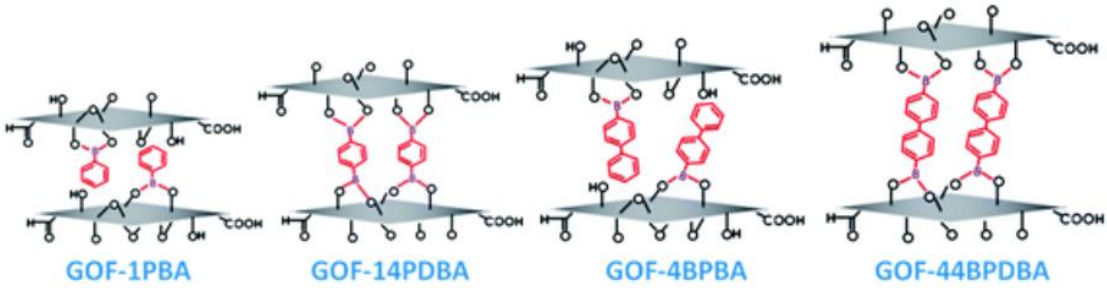

Figure 2.2 - Srinivas[55] proposes that both phenyl and benzyl groups form pillars by bonding to either one or both sides of the pore.

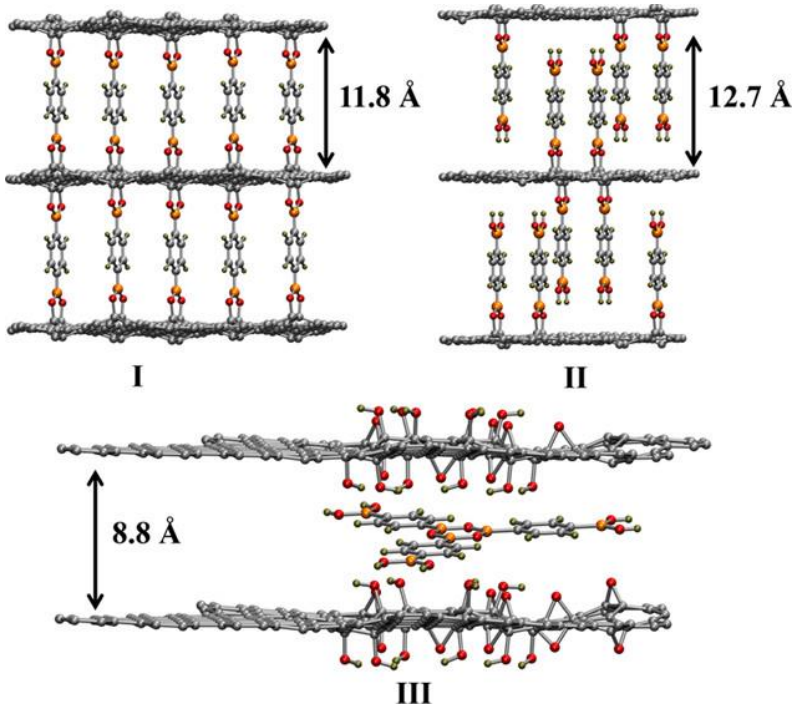

Figure 2.3 – GOF models proposed by Burress *et al*.[9] depicted by Mercier *et al.*[52] including models with DBA covalently bonded to the top and bottom (top left), bonded to one side (top right) and unbonded (bottom).

2.2 Overview of GOFs with Covalently Bonded Benzene-1,4-diboronic Acid: Competing Models

Burress *et al.*[9] assumed that GOFs were comprised of GO layers separated by rigid DBA linkers: oxygen atoms were covalently bonded to both graphene planes and benzene diboronic linkers, and that all other functional groups such as epoxy-oxygen, etc. were absent (**Figure 2.3**). In this work, when the linkers are at 90^{o}, the structures are called **Covalent Pillared** (CP-GOF) (**Figure 2.4**). Mercier *et al.* posited, however, that the proposed structures would be unlikely given the size of the DBA linker and that since they observed the pores to swell by more than 50% when flooded with water, i.e., that the DBA is unlikely to be covalently bonded for this to happen[52] (our detailed results also partially confirm this picture, as we will show below). Moreover, the cross-linking of graphene planes according to GOF structure requires an unlikely combination of four hydroxyl groups located exactly over each other on the neighboring GO sheets. Thus Mercier[52] proposed two potential candidates. In one model, each DBA molecule is attached only to one GO sheet and not attached to the neighboring sheet, structures which in this work we call **Van der Waals GOF** (vdW-GOF) (**Figure 2.5**).

An alternative model was also proposed where the DBA molecules are not covalently bonded at all, remaining in a **fluid-DBA** structure (**Figure 2.6**) while helping separate the GO layers, probably stabilized by electrostatic interactions between the DBA end groups and polar groups on the GO. As we will see below, this model fails to reproduce the layer spacing for GOFs.

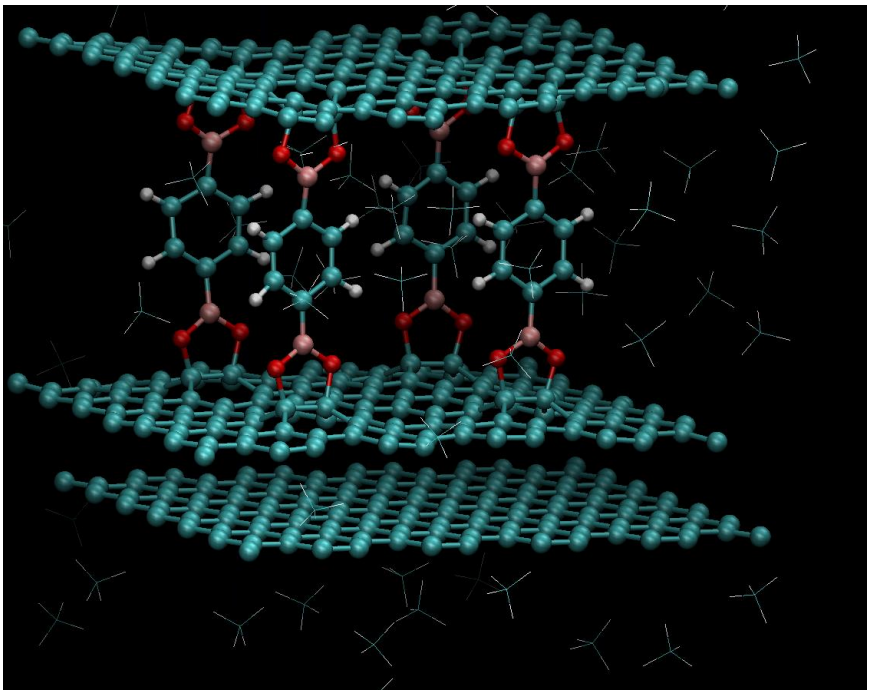

Figure 2.4 - The covalent pillared GOF model. A stabilizing layer is included at the bottom to simulate an underlying solid graphite-like structure. See also **Figs. 2.5-2.7** for additional models.

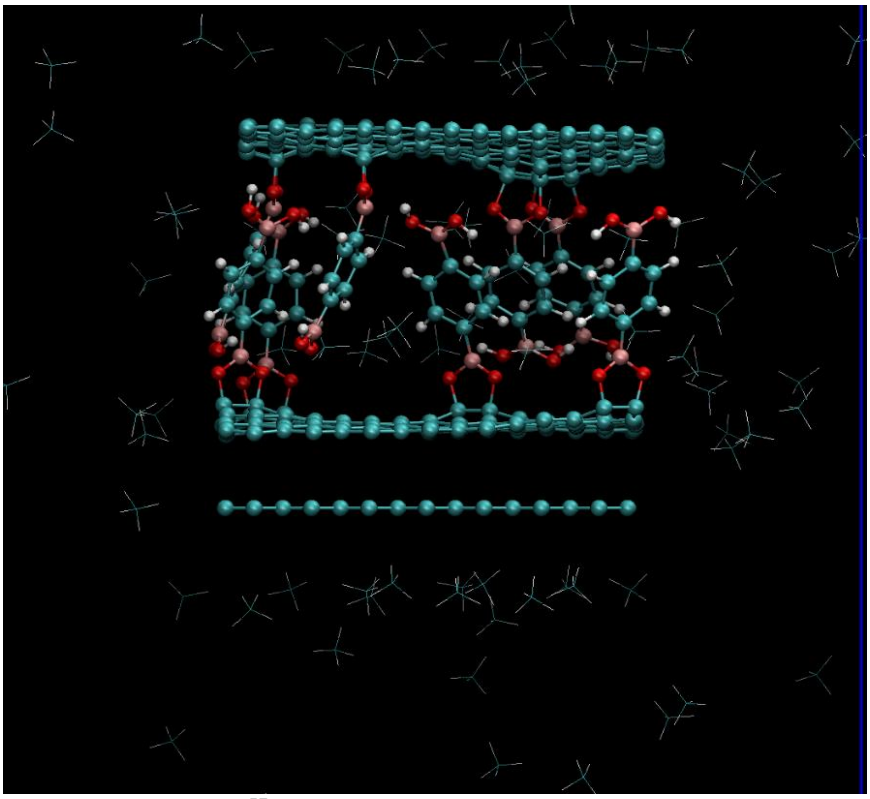

Figure 2.5 - The vdw DBA GOF[57] model**.**

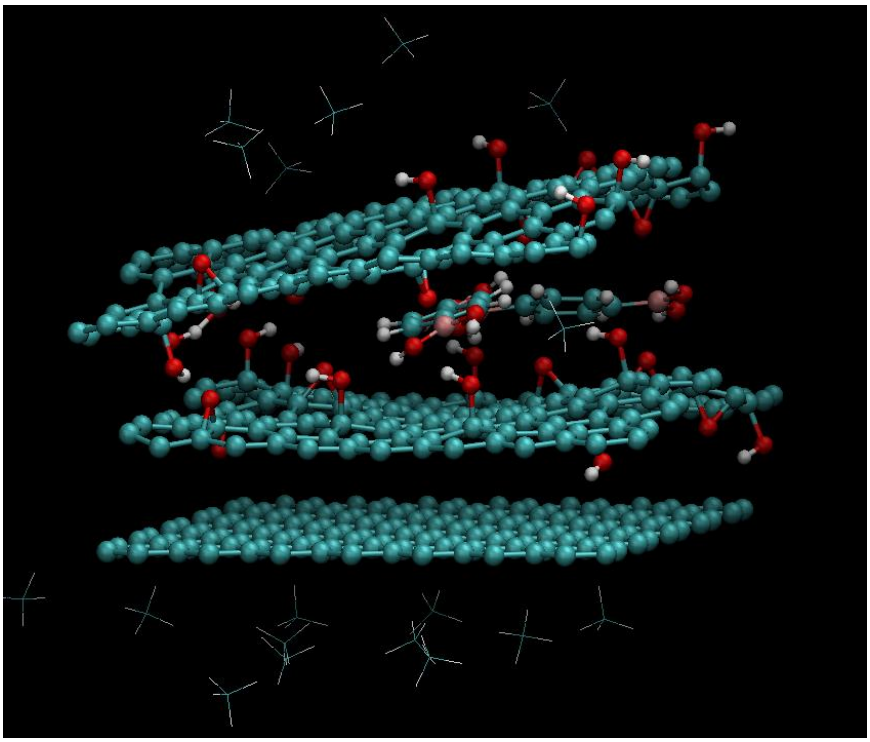

Figure 2.6 - The fluid DBA GOF[57] model.

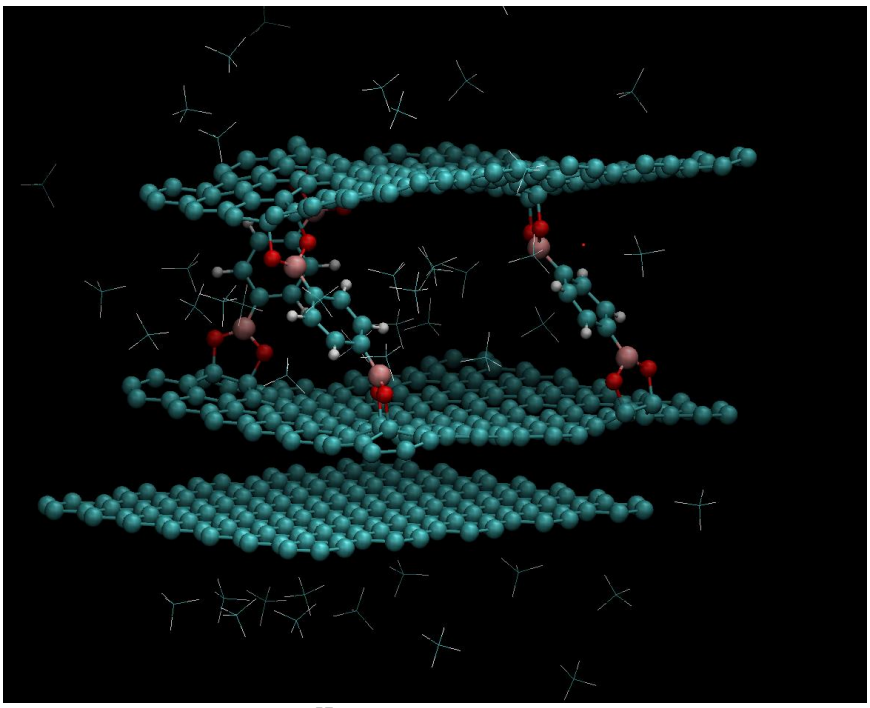

Figure 2.7 - The covalent angled GOF[57] model.

In this work we propose a new GOF model that retains the main characteristics of Burress *et al.*'s[9] but that is compatible with experimental observations of base layer

spacing and swelling with adsorption of both CH_4 and Xe[1]. The model also reproduces reasonably well the adsorption isotherms determined experimentally[1]. Instead of considering that the linkers are perpendicular to the graphene plane, which would imply that the carbons to which they are linked are exactly one above the other and the distance between the different graphene planes would be 10.6 Å, we allow the DBAs to bind to one GO at some random pair of carbon atoms, and then bind to the other GO layer in a random place within geometrical constraints to produce a reasonable configuration i.e. at displacements between 4.5 and 7.5 Å in the plane of the GOs. An initial energy minimization using the CHARMM22 force field[58] in NAMD2[59] is then performed. This model is called **Covalent Angled** (CA-GOF) (**Figure 2.7**). We placed enough DBA molecules so that the boron content is comparable to that observed[1,9,52].

2.3 Molecular Dynamics Simulations

In this work we use fully atomistic molecular dynamics (MD) simulations to observe the behavior of the previously stated models (**Figure 2.4**-**Figure 2.7**) of GOF-DBA under conditions similar to those of the experiments of Schaeperkoetter *et al.*[1]: CH_4 and xenon pressures in the 0-120 bar range at room temperature (T = 300 K). All simulations were completed using the NAMD2[59] MD code and analyzed using Visual Molecular Dynamics (VMD) and various Tool Command Language (TCL) scripts[60]. Interactions were cut off for distances larger than 12 Å and a list of neighbors with a radius of 14 Å was maintained for accelerated calculations of forces. For each run, 5,000 steps of energy minimization were performed, followed by 3,000,000 × 1 fs steps (i.e., 3 ns) of MD simulations in the canonical (N, V, T) ensemble using a velocity rescaling

thermostat every 20 fs with configurations saved for analysis every 1,000 time steps (i.e., every 1 ps). The last 2 ns of each simulation was used for calculations of time averages (all simulations equilibrated in less than 0.5 ns). All simulations were performed inside a 21.3 Å × 37.2 Å × 50 Å parallelepiped box with periodic boundary conditions (PBC) in all directions, see **Figure 2.8**.

In each simulation the GOF structure (CP-GOF, vdW-GOF, CA-GOF), GO and DBA (fluid-DBA) were simulated fully-atomistically and the atoms were allowed to move with the exception of a fixed graphene bottom layer that simulates a solid graphite substrate, increases the stability of the simulation, and facilitates the analysis by keeping the structures roughly aligned parallel to the *xy* plane. This is a reasonable model since GOF samples have SSAs ≈ 400-500 m^2/g[1] compared to graphene's theoretical surface area of 2600 m^2/g.

The gases (CH_4, xenon) were also simulated fully-atomistically. The number of gas molecules $N_{met} \in [0, 400]$, $N_{Xe} \in [0, 400]$ was varied (placed randomly using **Program C8** in **Appendix C**), and the pressure was calculated using an interpolating function (**Appendix C**, **Eqs. C1-2**) based on NIST's Thermophysical Properties of Fluid Systems database[61] from gas densities averaged far from the GOF (and DBA), i.e., in the bottom 10 Å of the simulation box, see **Figure 2.8**[62].

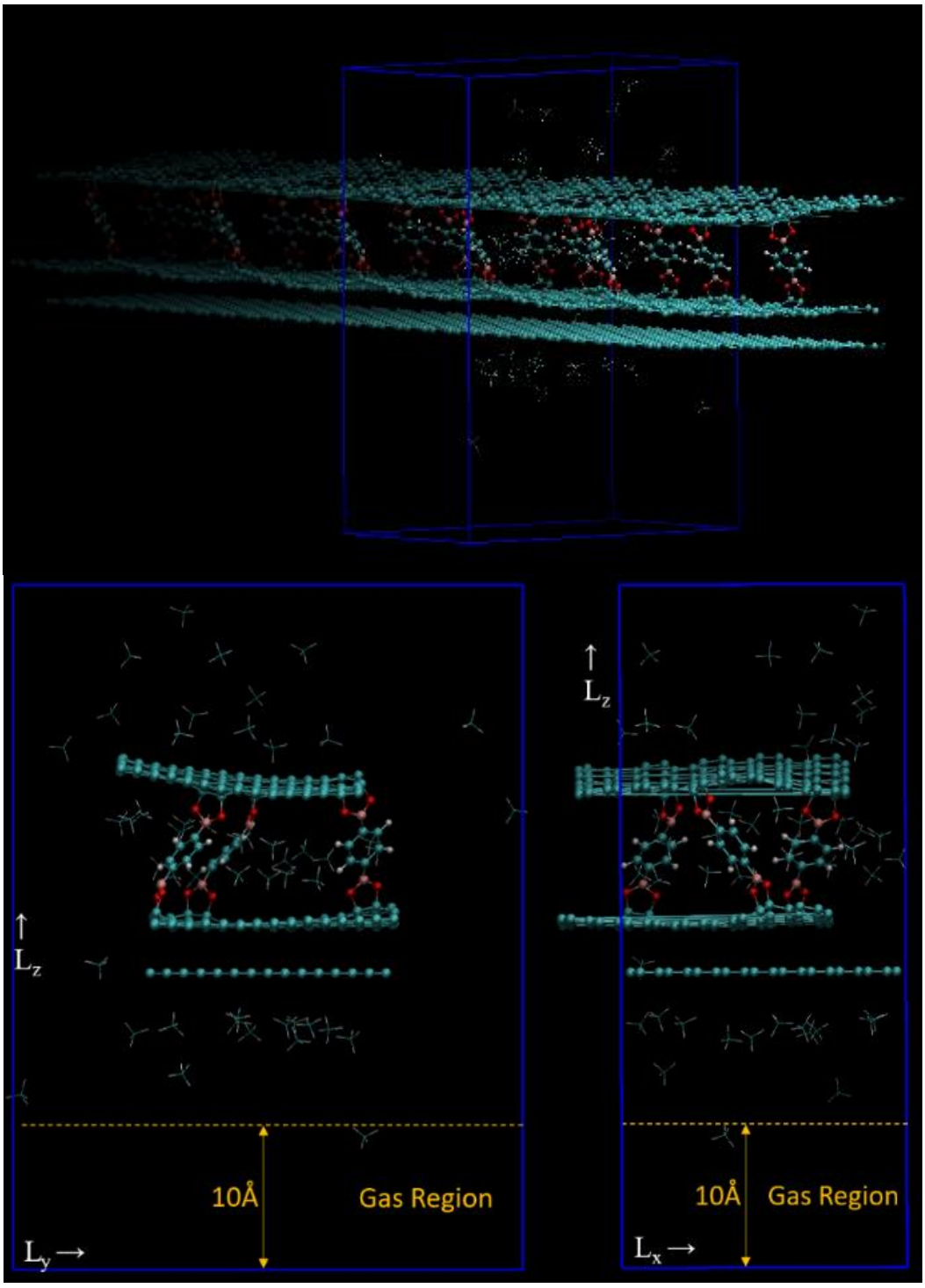

Figure 2.8 - Head on view of a typical system along *x* axis (bottom left) showing regions (small regions (small and large *y*) where gas molecules can transfer between the pore and gas phases. The system viewed along the *y* axis (bottom right) shows carbon atoms in GOF bonded across the PBC. The "Gas Regions" in yellow, situated farther than the potential cut-off distance (12 Å) from the GOF depict where the gas density is measured[62] and the pressure calculated using NIST's Thermophysical Properties of Fluid Systems database[61]. The top panel illustrates the "strip" shape of the simulated GOF, i.e., infinite in the *x* but finite in the *y* direction[62].

For the interactions we used the CHARMM22 force field[58]:

$$V = V_{\text{bond}} + V_{\text{angle}} + V_{\text{dihed}} + V_{\text{nb}} . \quad (2.1)$$

The bond term is modeled in the harmonic form:

$$V_{\text{bond}} = \sum_{\text{bonds}} k_b(b - b_0)^2 , \quad (2.2)$$

where b is the bond length between a pair (1-2) of atoms, b_0 is the equilibrium bond length and k_b is the bond force constant. To avoid unnecessary short time scales the C−H bonds were made rigid ($b = b_0$) using the RATTLE algorithm[63], which allowed a time step of 1 fs. The second term accounts for the angle deformation:

$$V_{\text{angle}} = \sum_{\text{angles}} (k_\theta(\theta - \theta_0)^2 + k_{\text{UB}}(S - S_0)^2) , \quad (2.3)$$

where θ is the angle between a sequence of 3 bonded atoms, θ_0 the bond-angle equilibrium value, k_θ the angular harmonic stiffness constant, S_0 is the equilibrium distance between the 1−3 pairs, and k_{UB} the Urey−Bradley constant. The third term accounts for four-body dihedral torsion contributions:

$$V_{\text{dihed}} = \sum_{\text{dihedrals}} k_\chi(1 + \cos(n\chi - \delta)) , \quad (2.4)$$

where χ is the dihedral angle formed by a sequence of 4 bonded atoms, k_χ is the torsional stiffness, n is the multiplicity factor, and δ is the phase shift. The final term makes up the nonbonded interactions derived from two-body interactions for atom pairs either in the same or different molecule and adatom-substrate pair interactions:

$$V_{\text{nb}} = \sum_{i,j'} \frac{q_i q_j}{4\pi\varepsilon_0 r_{ij}} + \sum_{i,j'} \varepsilon_{ij} \left[\left(\frac{r_{\text{min},ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{r_{\text{min},ij}}{r_{ij}} \right)^{6} \right] , \quad (2.5)$$

which are applied only to atom pairs separated by at least three bonds[64], with 1−4 interactions modified by a scaling factor of 0.4[64,65]. Here q_i are the Mulliken partial atomic charges, r_{ij} are the interatom distances, ε_0 is the electric constant, ε_{ij} is the Lennard-Jones potential depth and $r_{\min,ij}$ is the distance of the Lennard-Jones minimum. For heterogeneous atom pairs we use the Lorentz-Berthelot combination rules[66]:

$$\varepsilon_{ij} = \sqrt{\varepsilon_i \varepsilon_j}\,, \qquad r_{\min,ij} = \frac{r_{\min,i}}{2} + \frac{r_{\min,j}}{2}. \tag{2.6}$$

All LJ potentials were taken out to a pair separation of r_l = 10 Å and then smoothly diminished to zero at a cut-off distance r_c = 12 Å. Coulomb interactions were calculated using the particle mesh Ewald (PME) summation method[67]. The values of potential parameters are from a combination of *ab initio* density functional theory (DFT/B3LYP)[68–71] calculations using the Gaussian09 code[72] and the CGenFF database[73,74]. The values of the interaction parameters are listed in the **Appendix A**, **Tables A1-A5**, while the methodology for obtaining some of said parameters is in **Section 2.4**.

2.4 *Ab Initio* Calculations

Ab initio calculations, meaning calculations "from the beginning", are used to solve for a molecule's electronic wavefunctions by only providing physical information about the system and solving for the rest. A typical end goal is to find molecular configurations that are energetically favorable (energy minima). Additionally, we often are also interested in the dependence of the molecule's energy on its configuration, and interactions between different molecules, i.e., the potential energy surface, or force field.

Given that DBA molecules in general and DBA bonded to graphene/GO/GOF are not well characterized in the literature, we performed extensive *ab initio* calculations in Gaussian09[72] using density functional theory (DFT) with the B3LYP functional[68–70,75] and Pople-type split valence 6-31G(d) basis set[68,75,76] to determine the optimized structure, and interaction parameters that we used for our molecular dynamics simulations. The calculations described below resulted in the determination of all bond lengths and potentials, all bond angles and potentials, and some dihedrals.

Figure 2.9 and **Figure 2.10** illustrate the atom types used in the interaction parameters, which are listed in **Appendix A**, **Tables A1-5**.

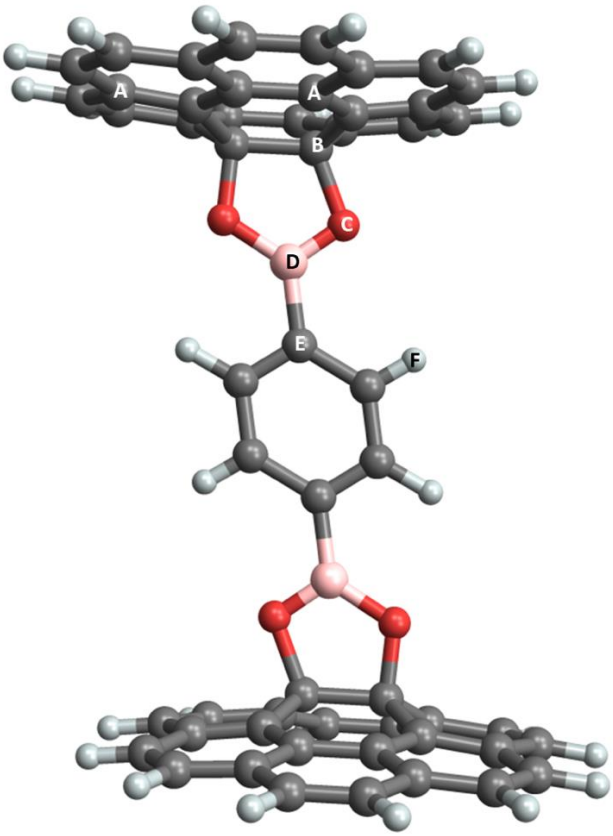

Figure 2.9 - Description of atom types for GOF. **A**: sp2 graphene carbons CA, **B**: partially hybridized sp3 graphene carbons CSP23, **C**: oxygen linker between B and C OXGN, **D**: DBA boron BCOO, **E**: aromatic carbons CG2R61, **F**: aromatic hydrogens HGR61[57].

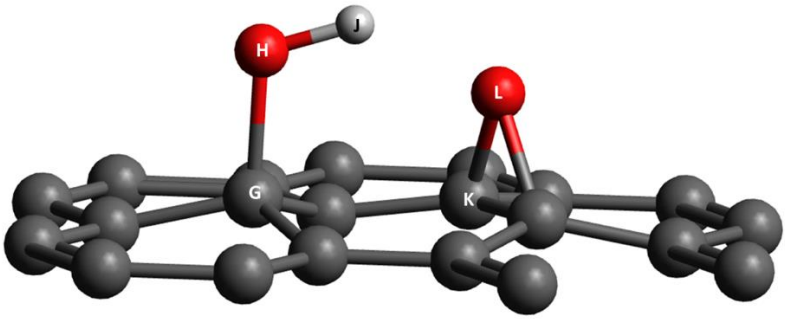

Figure 2.10 - Description of atom types for GO. **G:** partially hybridized sp3 graphene carbons type CSP23, **H:** hydroxyl oxygen OXGN, **J:** hydroxyl hydrogen HCP1, **K:** epoxy carbon CG3C31, **L:** epoxy oxygen OG3C31[57]. Coronene-Hydroxyl: qO = -0.60, qH = +0.39, qCSP23 (bonded to O) = +0.21. bCCSP23 = 1.51 Å, bCSP23CSP23 = 1.59 Å, bOC = 1.46 Å, bOH = 0.97 Å, aCOH = 105.5°, dCCOH = 180°, 60.6°, -60.6° (i.e., n = 3), kbOH = 550 kcal/mol Å^2, kbCO = 254-400 kcal/mol Å^2, kaCOH = 67 kcal/mol rad^2, kaCCO = 95 kcal/mol rad^2, $k_{\chi 3}$CCOH = 0.41 kcal/mol.

The determination of the bonded elastic constants (**Eqs. 2.2-2.4** and **Tables A1-4**) was performed by first optimizing the structure(s) and then varying one or two parameters of interest (see below) to determine the curvature of the potential. We proceeded from stronger (bonds), to weaker (angles), to weakest (dihedrals) performing constrained optimizations in each case. It should be noted that because of geometric constraints it is not always possible to vary some structural parameters independently of each other's, so some of the elastic constants are the result of reasonable partitions of the interaction energies, aided by consulting parameters published in the CGenFF database[73,74]. To minimize the number of independent atom types we also varied angles in the aromatic ring and around the B atom by ~ ± 2° so that all angles were 120°. All Mulliken partial charges (**Table A5**) are the result of our DFT calculations. However van der Waals ε and r_{min} for each atom type (**Table A5**) are from the CGenFF database[73,74].

Overall, our parameters for the GOF-DBA, GO, and DBA, are comparable to the set calculated by Nicolai *et al*[77] for a qualitatively similar, but simpler, system. It should

be noted that, particularly for the non-bonded terms, both our parameters and those of Nicolaï *et al.*[77] do not take into account modifications by the chemical environment (e.g., presence of water or other polar solvents). Thus, they should be considered with caution when used outside the scope of the scenarios considered in this work.

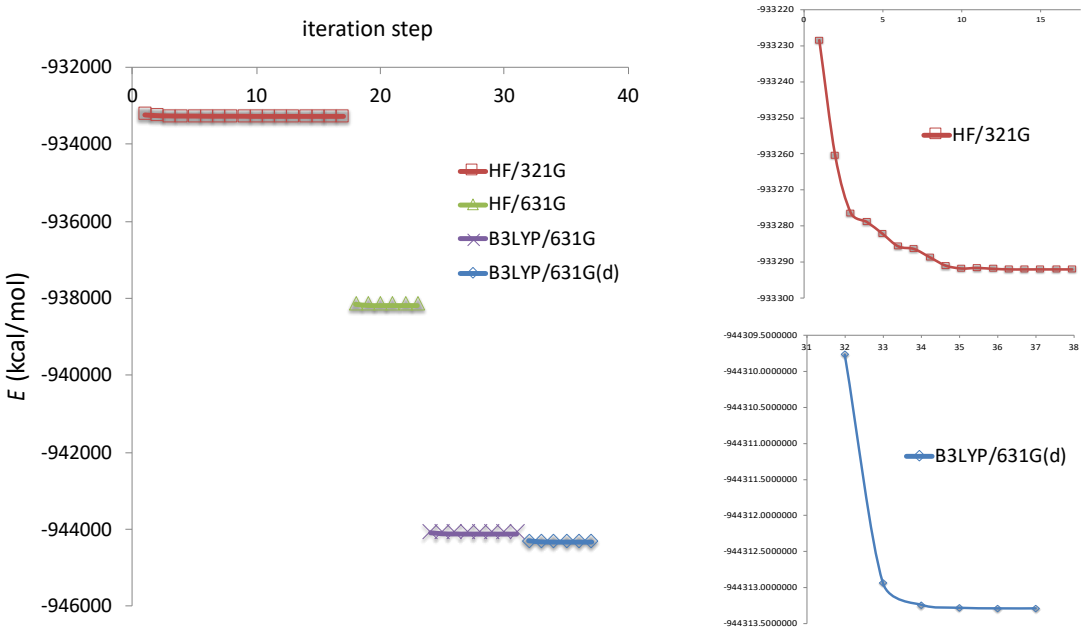

Figure 2.11 - Structural optimization for DBA-Coronene (left) (see **Figure 2.13**). Insets (right) show initial HF/3-21(G) and final B3LYP/6-31G(d) steps[57].

For all systems below, optimization of the molecule was performed in a multi-step process starting with a Hartree-Fock approximation with a simple 3-21G basis set and culminating with a B3LYP/6-31G(d) DFT calculation. **Figure 2.11** shows the minimization procedure for the system DBA-Coronene described below. Convergence within each theory is reasonably fast, with the steps between theories resulting from the different theory or basis sets but with very small structural changes.

DBA Molecule: We first optimized a single DBA molecule *in vacuo*. At the B3LYP/6-31G(d) level of DFT simulation, the structures shown in **Figure 2.12** are local

minima with energies within a few k_BT of one another and thus are accessible. Partial charges shown in the figure are comparable, and of similar values (for atoms removed from the endpoints) to DBAs linked singly, or doubly to graphene.

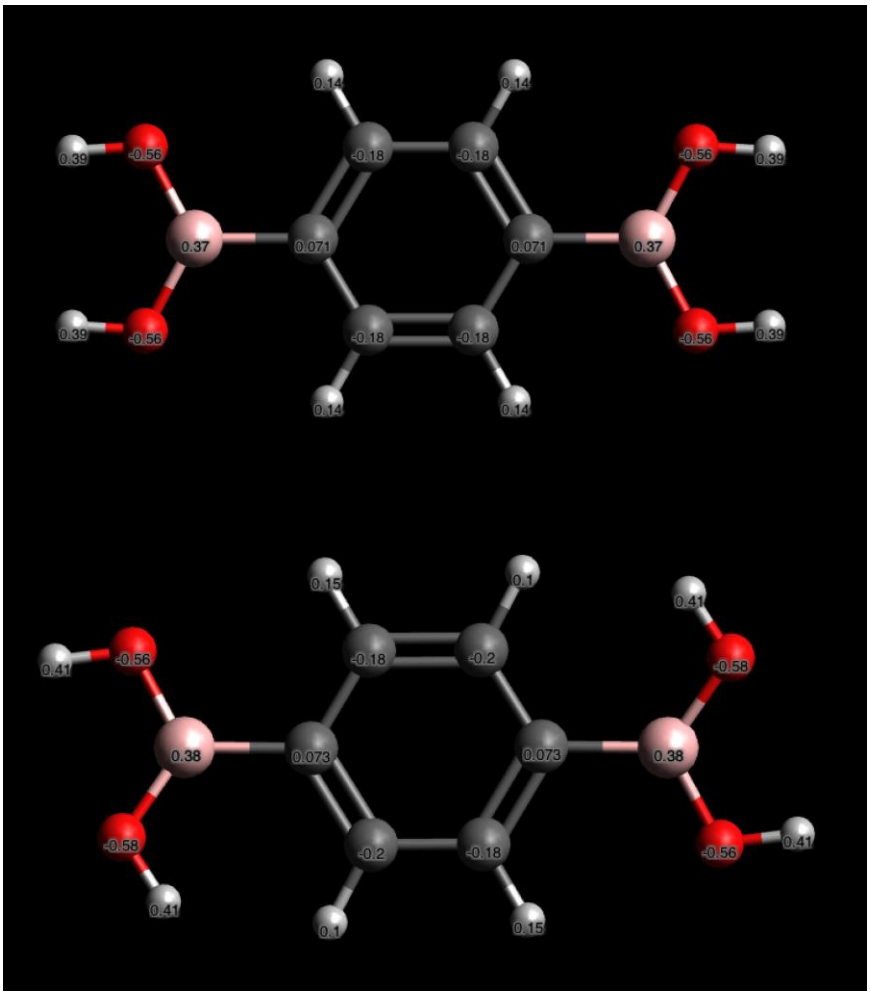

Figure 2.12 - B3LYP/6-31G(d) local minima for DBA molecules. Partial charges are shown and quite similar to one another. The top structure is planar, the bottom has a twist of ~ 5°. Bond lengths (in Å) are *b*CH = 1.081, *b*CC = 1.397, *b*CB = 1.554, *b*BO = 1.397, *b*OH = 0.967. Angles are *a*OBO = 114°, *a*OBC = 123°, *a*BCC = 120°, *a*BOH = 117° and were rounded to 120° in the simulations to minimize the number of independent parameters. Dihedrals HOBO, HOBC, OBCC had $n = 2$ and $\delta = 180°$ [57].

DBA-Coronene: To model the interaction of DBA with graphene, we first considered a single DBA molecule covalently bonded to coronene, as shown in **Figure 2.13**. For the purposes of simulation of DBA-graphene, partial charges for C atoms beyond the ones linked to O were taken to be 0, and overall charge neutrality of the

system was ensured by making small adjustments of ± 0.01esu to other atomic charges. Structural parameters are shown below. **Figure 2.14** and **Figure 2.15** show the procedures to determine some of the elastic constants for this system.

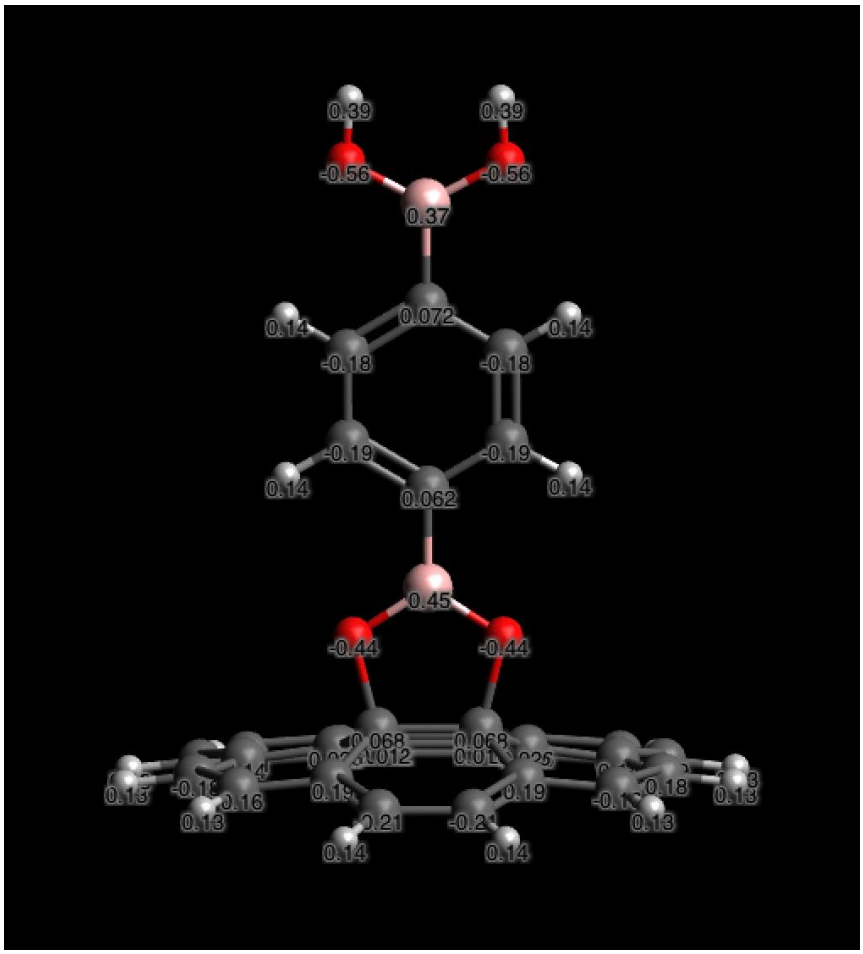

Figure 2.13 - B3LYP/6-31G(d) local minima for DBA-Coronene. Partial charges are shown. Charges (esu): qO1 = -0.44, qCSP23 = +0.07 (later revised for local consistency), qB1 = +0.45, qCB1 = +0.06, qCH = -0.18, qH = +0.14, qCB2 = +0.07, qB2 = +0.37, qO2 = -0.56, qHO = +0.39. Bonds (Å): *b*CCSP = 1.52 (all), *b*CSPCSP = 1.59, *b*CSPO = 1.47, *b*O1B = 1.37, *b*B1C = 1.56, *b*CB1C = 1.41, *b*CC = 1.39, *b*CH = 1.09, *b*CB2C = 1.41, *b*B2O2 = 1.56, *b*O2H = 0.97. Angles (°): aCCSPCSP = 116, aCCSPO = 104, aCSPCSPO = 104, aCSPOB = 110, aO1BO1 = 113, aO1BC = 123, aB1CC = 121 (→ 120), all angles in benzene 120 ± 1 (→ 120), aCCB2 = 121 (→ 120), aCB2O = 118 (→ 120), aO2BO2 = 124 (→ 120), aB2OH = 115. Dihedrals δ (°) and *n*: dCCSPOB = 122 or -122, dCSPCSPOB = 0, nCCOB = 3 for, dCCCSPO = 74, -75, -103, 103, -113, 113, *n* = 1. All dihedrals inside DBA made 0 or 180 for consistency, nOBCC = 2, nBCCH = 1, nBCCC = 1, nCCCC = 1, nHCCH = 1, nHCCC = 1, nCCBH = 2, nCBOH = 1[57].

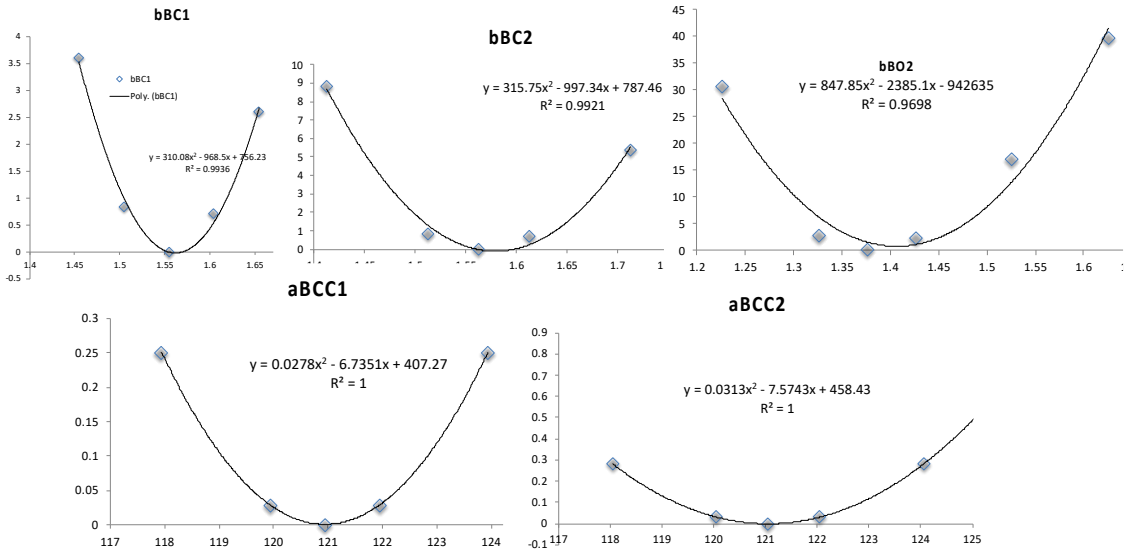

Figure 2.14 - B3LYP/6-31G(d) example bond and angle elastic constants (**Eqs. 2**, **3**) determined by constrained optimizations of DBA-Coronene with varying bonds and angles. The procedures above yield *k*BC1 = 310 kcal/mol Å^2, *k*BC2 = 316 kcal/mol Å^2, *k*BO2 = 424 kcal/mol Å^2, *k*BCC1 = 46 kcal/mol rad^2, *k*BCC2 = 51 kcal/mol rad^2 [57]**.**

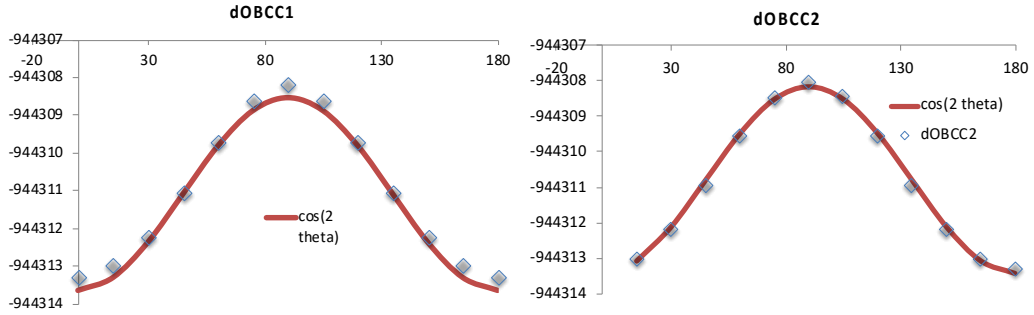

Figure 2.15 - B3LYP/6-31G(d) dihedral rotation parameters [**Eq. 4**: $V(\chi) = V_0 + k_\chi \cos(n\chi + \delta)$] by constrained optimizations of DBA-Coronene with varying dihedrals. In both cases $n = 2$, $\delta = 180°$. These result in $k_\chi = 2.59 \pm 0.05$ kcal/mol[57].

Coronene-DBA-Coronene: To model the interaction of DBA covalently bonded with graphenes on both sides, we considered a single DBA molecule covalently bonded to two coronenes, as shown in **Figure 2.16**. For the purposes of simulation of DBA-graphene, partial charges for C atoms beyond the ones linked to O was taken to be zero, and overall charge neutrality of the system was ensured by making small adjustments of ± 0.01 to other atomic charges. Structural parameters are shown below. Elastic constants

for bonds, angles, and dihedrals (**Eqs. 2.2-2.4**) were determined similarly to the case DBA-Coronene (**Figure 2.14** and **Figure 2.15**).

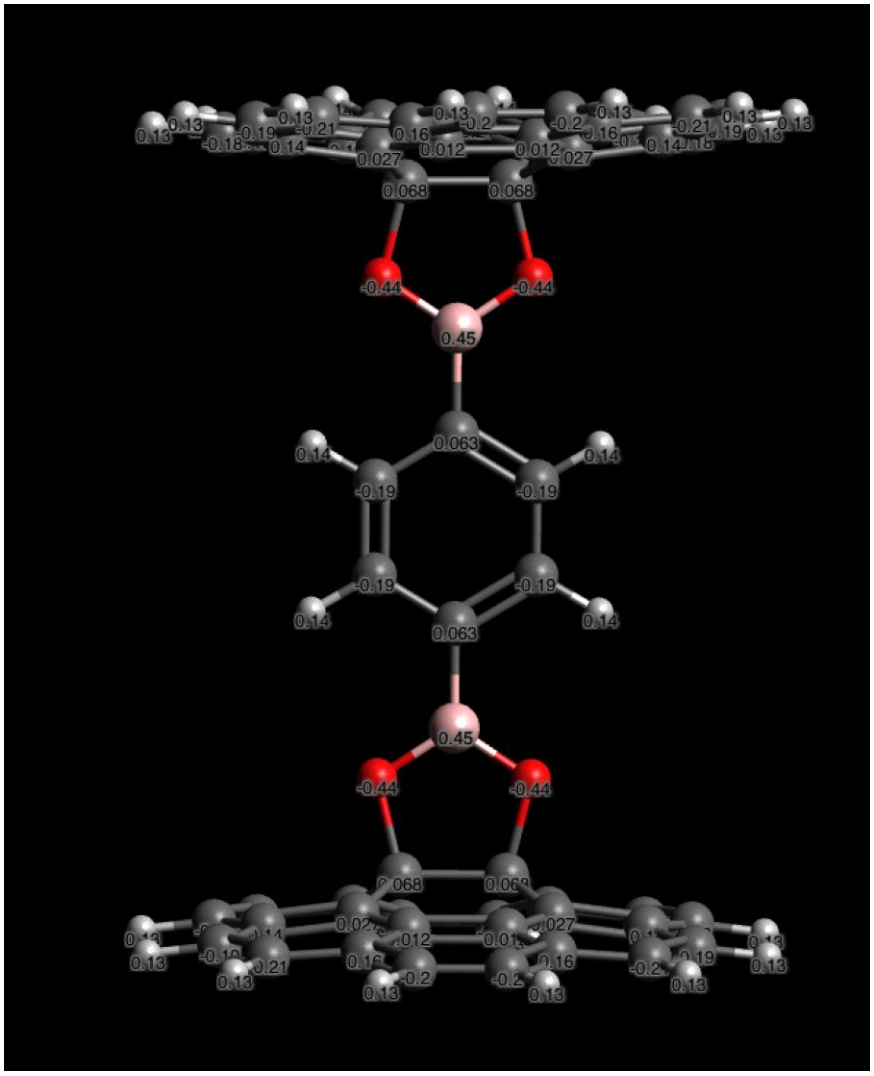

Figure 2.16 - B3LYP/6-31G(d) local minima for DBA-Coronene. Partial charges are shown. Charges (esu): qB = +0.45, qO = -0.44, qCB = +0.06, qCH = -0.19, qH = +0.14, qCSP = +0.68. Bonds (Å): bOB = 1.37, bBC = 1.56, bOC = 1.47, bCSPCSP = 1.59, bCCSP = 1.52, bCBC = 1.41, bCC = 1.39, bCH = 1.09. Angles (°): CSPOB = 110, aCCSPO = 104, aCSPCSPO = 104, aCCSPC = 112, aCCSPCSP = 116. aOBO = 113, aOBC = 123 (→120), aBCC = 121 (→ 120), all angles in benzene 120 ± 1 (→ 120). Dihedrals δ (°) and n: see analysis for DBA-Coronene[57].

Other systems: For GO, the charges, structure and elastic parameters were computed by analyzing one and two hydroxyl and one epoxy oxygen attached to

coronene in a way similar to the discussion above for DBA, DBA-Coronene, and Coronene-DBA-Coronene as can be seen in **Figure 2.10.**

2.5 Running and Selecting Configurations

In every simulation type, DBA molecules were randomly placed across multiple trials. For the CP-GOF, a random pair of bonds was made on the top and the bottom. For Fluid-DBA, the molecules were placed randomly in the pore perpendicular to the graphene layers. For the vdW-GOFs, an even number of single random bonds were selected on the top layer or bottom layer. The CA-GOF was placed randomly using ***Cov-DBA-Placer6.py*** listed in **Appendix C** (where all programs/scripts listed below reside).

Starting out, we ran MD simulations for one CP-GOF, twenty-six vdW-GOF, twelve fluid-DBA, and ten CA-GOF's with $N_{\text{met}} \in [0, 400]$ before deciding to test the CA-GOF further. To reach this decision, we analyzed the final 2 ns of every MD simulation after 1 ns equilibration using Visual Molecular Dynamics (VMD) and TCL scripts[60] to calculate adsorption isotherms and obtain other structural configuration information about the system. We used the **script *com_pz_GO-201806.tcl*** (**Appendix C, Program C2**) to record the z-positions of every carbon atom in the graphene layers for the final 2,000 frames (2,000,000 MD steps) which were then used to calculate the average pore spacing for that configuration. Since stable pores have spaces between them without another layer's atoms coming near the other, we are able to take the positions of each atom over many frames which form distinct bumps in a histogram like in **Figure 2.17**. A bump is then averaged to calculate each layer's average position and thus the difference in average spacing between the two bumps yields the pore size for that

configuration; we used **Program C5** for this process. Configurations that collapsed look like ones below in **Figure 2.18**; in both cases the average pore spacing is either too low or the histograms very "disperse" compared to the experiments[1] we were trying to match.

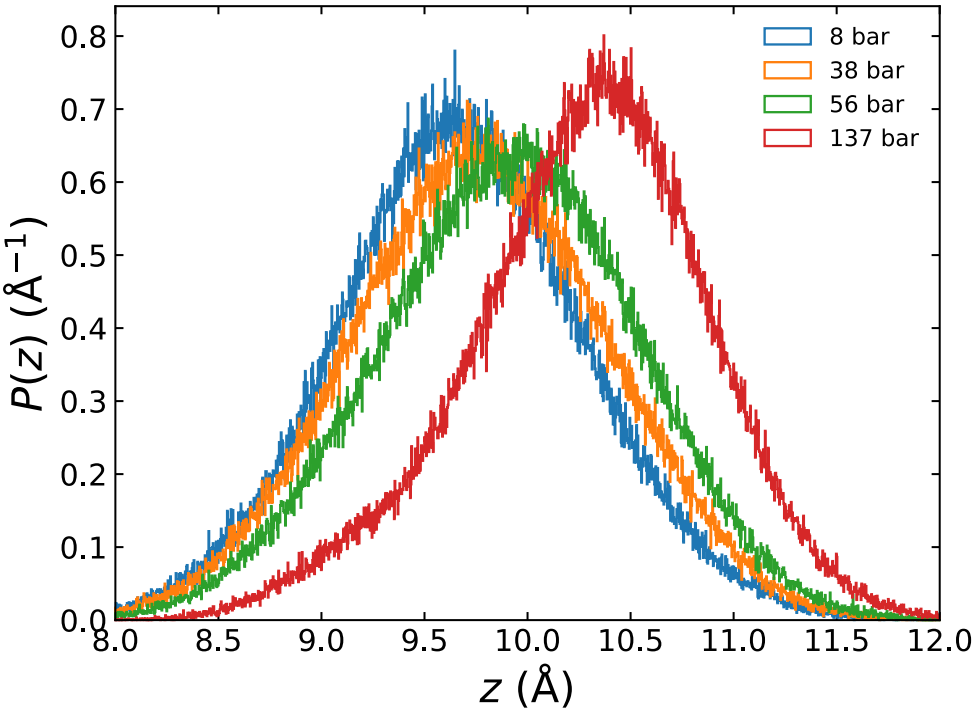

Figure 2.17 - Height density plot of carbon atoms in the top layer of graphene in the simulated GOF relative to the bottom layer at various CH_4 pressures for one typical "**covalent angled GOF**" configuration[57]. We observe a gradual increase of the average pore spacing d_{001} = <z> with pressure similar to what is observed experimentally[1].

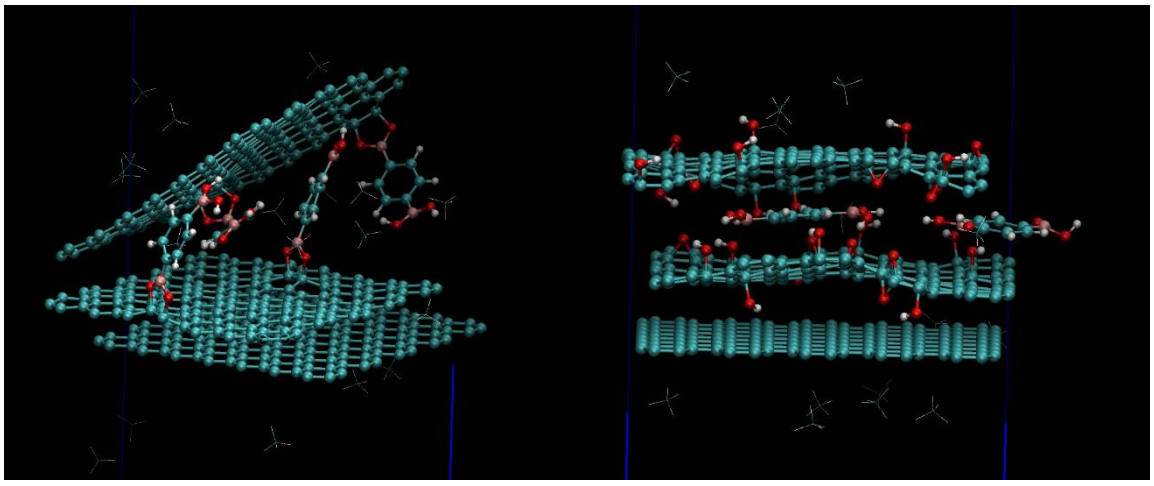

Figure 2.18 - Typical collapsed van der Waals GOF (left) and fluid-DBA (right) models. These configurations almost completely excluded any gas molecules from adsorbing inside the pores[57].

After dozens of test-runs for each configuration, our preliminary results began to suggest that the **CA-GOF** model is the only one close to matching the experiments. We then randomly generated hundreds of test configurations of **CA-GOF** models which included 3 DBA each (i.e., yielding B:C composition ratios compatible with the experimental samples). We then ran the hundreds of test configuration at zero gas pressure using the script **Program C6**, followed by calculating the average pore spacing using **Program C4**, finally running **Program C5** to get images of the pores. From the list, we would choose pores within ±0.5Å of the initial pore spacing of the experiments[1] and check their images to make sure the top layer was reasonably flat. After identifying stable pores with the desired spacing, we would run them with $N_{\text{met}} \in [0, 400]$ and $N_{\text{Xe}} \in [0, 400]$ using **Program C7**.

2.6 Results

After 1 ns equilibration, we analyzed the final 2 ns of every MD simulation using Visual Molecular Dynamics (VMD) and the TCL scripts[60] mentioned above (**Program C2** and **C3**) to calculate adsorption isotherms and obtain other structural configuration information about the system.

Since our structures were created at random, some resulted in pores that partially collapsed (e.g., because all DBA linkers were clustered in one small area)—these runs were discarded. We observe that the density distribution $P(z)$ varies with the number of gas molecules (pressure) and, in general, we observe that the average pore spacing $d_{001} = \langle z \rangle$ increases with pressure as observed via neutron scattering by Schaeperkoetter *et al.* [1]. Schaeperkoetter observed that the interlayer spacing d_{001} increases gradually and monotonically as a function of pressure for both CH_4 and xenon[1]. This slow, gradual

change does not fit the typical "gate-opening" transitions that had been previously observed in other porous materials under subcritical adsorption[78–80]. Furthermore, in these other systems a bulk phase change usually precedes the gate-opening such as capillary condensation, but our gas's isotherms were measured at supercritical conditions making a bulk phase change impossible. Schaeperkoetter also observed that the increases in d_{001} appeared to cluster into a single curve under scaling laws that could be derived from the van der Waals parameters of each gas[1].

For the **CP-GOF**[9] a number of DBA molecules (3 or 4 in our simulation box, as per the experimental B:C ratios of GOF samples[1,9,52,55]) are covalently linked to both sides of the pore through C-O bonds (**Eq. 2.2**) to graphene at sites that are almost exactly on top of each other (**Figure 2.4**, top left). Depicted in **Figure 2.20,** this results in a layer separation d_{001} = 10.6 Å in disagreement with the experimental neutron diffraction results for d_{001}: between 9.30 Å (*in vacuo*) and 9.75 Å (P_{Xe} = 40 bar)[1] (note that in Ref.[1] d_{001} = 10.3 Å using X-ray diffraction, but this result is for samples in "wet" air, consistent with observations in Ref.[52]). The covalent pillared GOF structures were unsurprisingly very stable, rigid, and pressure independent (**Figure 2.19**) and as we shall see showed no expansion during gas adsorption due to the substantial rigidity of the C-O covalent bonds and within the DBA itself against stretching.

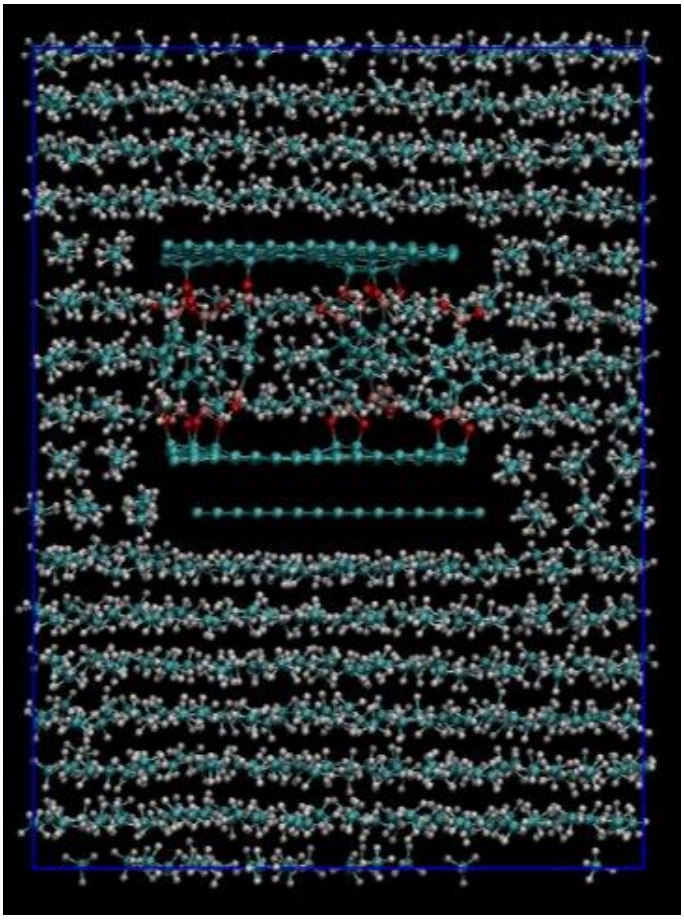

Figure 2.19 - A depiction of the covalent-pillared model with CH_4 at extreme (and unrealistic) pressure. Despite CH_4 "crystallizing", this particular GOF model did not expand.

The second model we ruled out was the **vdW-covalent model** (**Figure 2.5**). We again placed 2-4 DBA molecules (again consistent with experimental B:C ratios[1,9,52,55]) but bonded them through C-O bonds to alternate sides, allowing non-bonded (**Eq. 2.5**) interactions between the DBA molecules to stabilize the structure. For 2-3 DBA molecules the structures were unstable (i.e., the pores collapsed to d_{001} < 6 Å, thus making them inaccessible to gas adsorption, see **Figure 2.18**, left panel), requiring at least 4 DBA molecules to remain open, but in that case the d_{001} ≈ 11-12.5 Å (see **Figure 2.20)** was also too large (and, as we shall see later, remarkably pressure independent, as expected for larger pores[81–83]). To test whether the problem was a finite-size effect we

doubled the *xy* dimensions of the simulation cell (quadrupling the GOF surface area), but the results were unchanged: the van der Waals GOFs resulted in either a collapsed pore or a very large and stable pore.

The third model we dismissed was the **fluid-DBA model** (**Figure 2.6**). We created GO by adding epoxy and hydroxyl groups randomly to graphene based on the Lerf-Klinowski model[84–86] and randomly placed 2-6 DBA molecules between the GO layers. We tested numerous randomly produced GO/fluid-DBA configurations, but the results were always the same: a complete collapse of the pores to interlayer distances d_{001} < 7.8 Å (see **Figure 2.20)** that almost completely excluded any gas molecules from adsorbing inside the pore, see **Figure 2.18**, right panel.

The only model consistent with the experimental evidence was the **covalent angled model** (**Figure 2.7**). We made sure to retain the main characteristics of the models by Burress *et al*[9]., and managed to observe that it is compatible with experimental observations of base layer spacing $d_{001} \approx 9.3$ Å (see **Figure 2.20)** and, as we shall see below, shows adsorption isotherms and swelling of the interlayer spacing consistent with the experimental results of Schaeperkoetter *et al.*[1] for both CH_4 and xenon. The generalization is intuitive; instead of linking the DBAs perpendicularly to the graphene planes, we firstly covalently bond them (**Eq. 2.2**) to a random pair of C atoms in one layer of graphene and bond them to the other layer at another random location displaced between 4.5 and 7.5 Å in the graphene plane to produce the desired layer spacings. In a typical simulation we place 3 DBA molecules per simulation cell so that the boron content is comparable to that observed[1,9,52]. An initial energy minimization in NAMD2 was performed to eliminate geometrically unstable configurations, in addition to

eliminating those for which the starting pore spacing d_{001} (N=0) differed by more than ±0.5 Å from the experimental results (9.3 Å). The idea is that the configurations that match the pore spacing of Schaeperkoetter *et al.*'s experiments[1] are representative of the GOFs from said experiments.

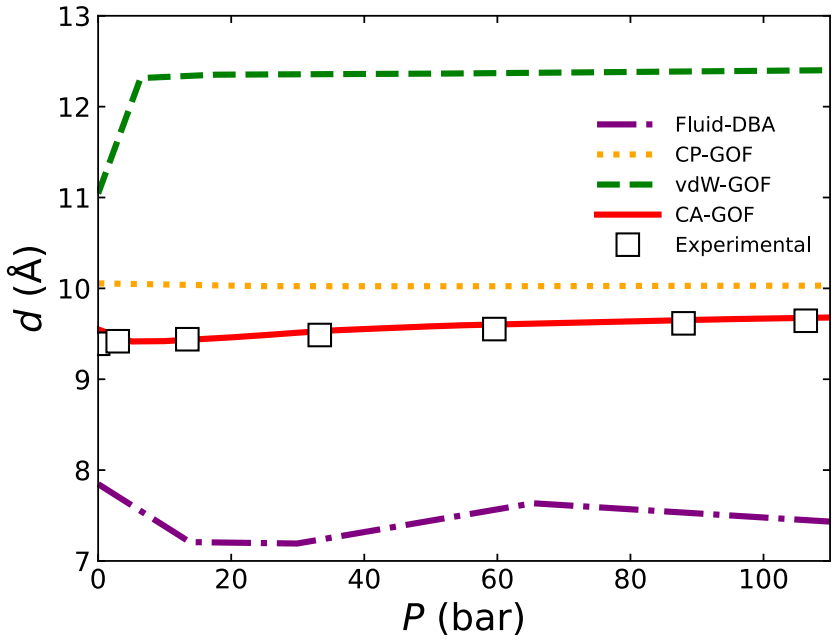

Figure 2.20 - Pore spacing d_{001} calculated by averaging over the all configurations for each of the discussed GOF-DBA models (**Figure 2.4-Figure 2.7**), and comparison with experimental results of Schaeperkoetter *et al.*[1] Error bars of the experimental data are omitted because they are too small at this scale. The dispersion of the computational data is shown in **Figure 2.21**[57].

2.6.1 Interlayer Spacing

In the case of the **CA-GOF** model proposed, one must be careful. The random arrangement of DBA molecules causes a range of possible distances between graphene sheets. For the distance between graphene sheets to be equal to the experimental distance, we found that the diborane bonds with the two graphene layers must be shifted in the *xy* direction by ~ 6 Å after testing random shifts between 4.5 and 7.5 Å. As mentioned above, we discarded results in which geometries resulted in pores that

collapsed or had spacings too large. **Figure 2.21** shows how the layer spacing percent changes $d_{001}/d_0 = \langle z/z_0 \rangle$ varies with CH_4 pressure for 32 random configurations color coded by how close they are to the low pressure experimental spacing[1]. It is evident that the results are quite disperse. However, averaging over these configurations (thick lines in **Figure 2.21**) results in curves that are in reasonable agreement with neutron diffraction experiments[1], and unsurprisingly the average calculated for the 7 configurations closest to the experimental low-pressure pore spacing have better agreement than the more disperse data sets. Regarding absolute pore spacing, results can be seen for $d_{001} = \langle z \rangle$ in **Figure 2.22**. A similar analysis was performed for the swelling of the interlayer spacing upon adsorption of Xe. The data set is more limited but shares CH_4's characteristics.

Figure 2.23 shows the variation of the layer spacing d_{001} as function of pressure during adsorption of CH_4 and xenon. The lines represent the *ensemble averages* of many GOF configurations as described above, symbols are from Schaeperkoetter *et al.*'s *in situ* neutron diffraction experiments[1]. Overall, the *ensemble average* variation of layer spacing with gas pressure for this model is in very reasonable agreement with experiments.

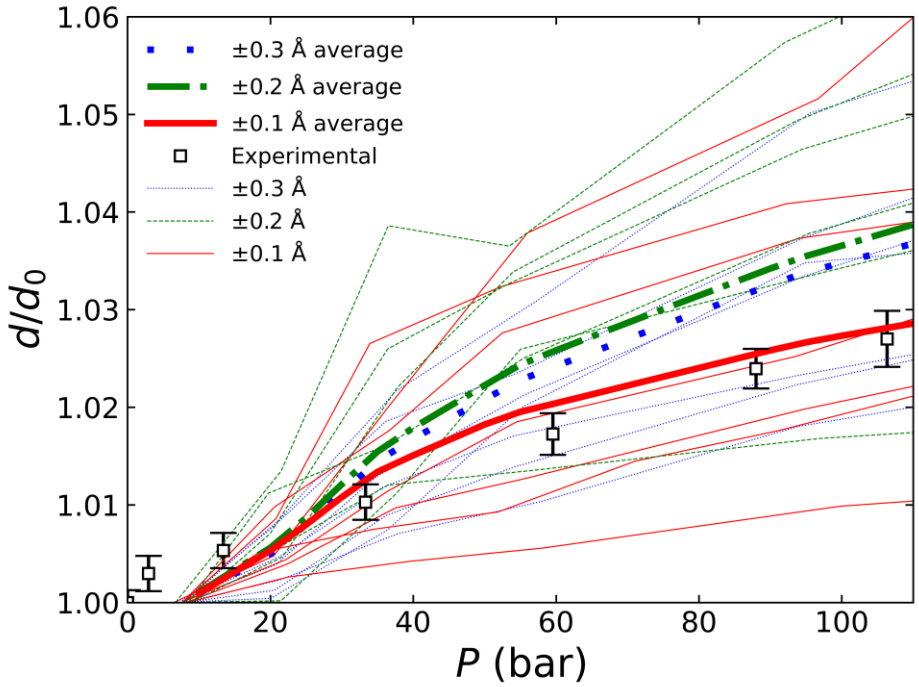

Figure 2.21 - Pore spacing relative increase d/d_0 as a function of CH_4 pressure P_{met}. Thin lines are individual configurations color coded by their low-pressure variation in d_{001} vs. the experiment[1] (note: d_0 = 9.35 Å). Thick lines represent averages within each class[57]. ±0.1 Å average means that only configurations where the low pressure pore spacing matched the experiments[1] were included in the average.

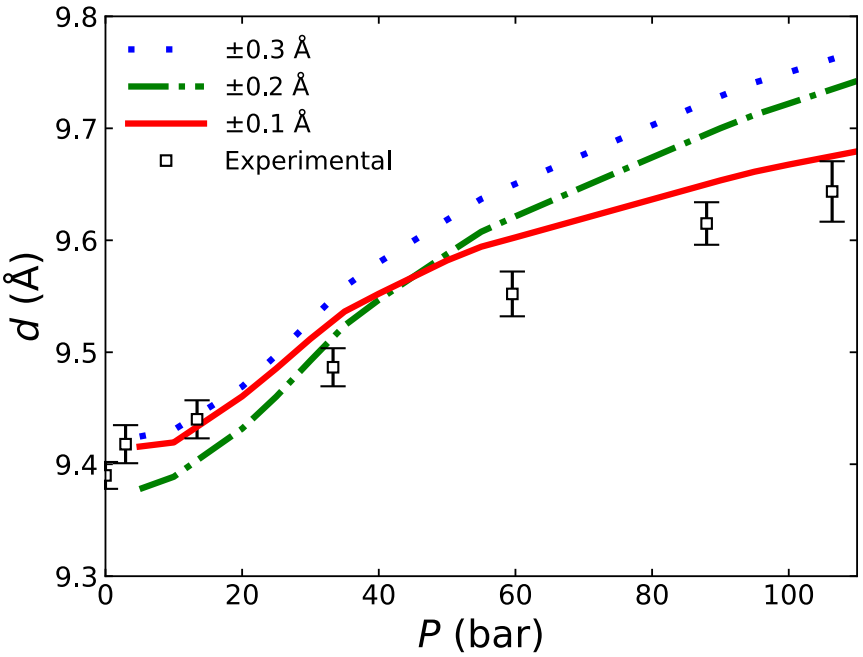

Figure 2.22 – Averaged pore spacing d as a function of CH_4 pressure.

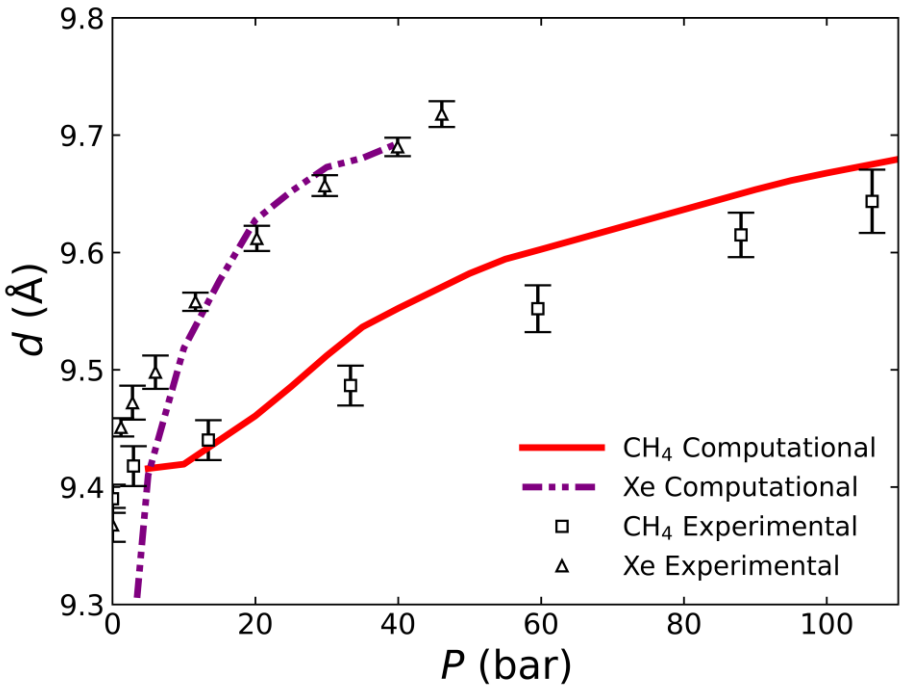

Figure 2.23 – Absolute pore spacing d_{001} as a function of CH_4 and xenon pressure. Experimental data from Schaeperkoetter *et al.*[1] The larger variation of d_{001} with pressure for Xe was explained by Schaeperkoetter *et al.*[1] as being a result of the different gas interaction parameters[57].

When looking at **Figure 2.21** for the CH_4 pore expansion isotherms, one can see the expansion follows two paths: either $> 4\%$ or $< 3\%$ increases in the pressure range considered. We find that the group with the higher percent increase in pore spacing correlates positively with the number of DBAs bonded to atoms on the y-edge or one atom from the y-edge (non-bonded edge) of the graphene sheet. Each configuration we studied had a DBA bonded to at least one carbon atom near the edge, indicating that at least one DBA must bond on the y-edge of the system if it is to be both stable and flexible.

2.6.2 Adsorption Isotherms

We created VMD/TCL scripts (**Program C3**) to compute the average number of gas molecules residing inside the pores to calculate the absolute adsorption isotherms for **CA-GOF** model structures. **Figure 2.24** shows the individual configurations' adsorption isotherms and their *ensemble averages* determined from **Figure 2.21**. There is much less sensitivity to the details of the configuration in this case and all "d_{001} tolerance levels" yield similar averages.

Figure 2.25 shows a comparison between the experimental and average simulated adsorption isotherms for CH_4 and xenon. Xenon adsorbed noticeably more rapidly in the simulations than in Schaeperkoetter *et al.*'s[1] experiments, but overall there is a reasonable agreement for both gases considering the large degree of variability in the simulated GOF structures.

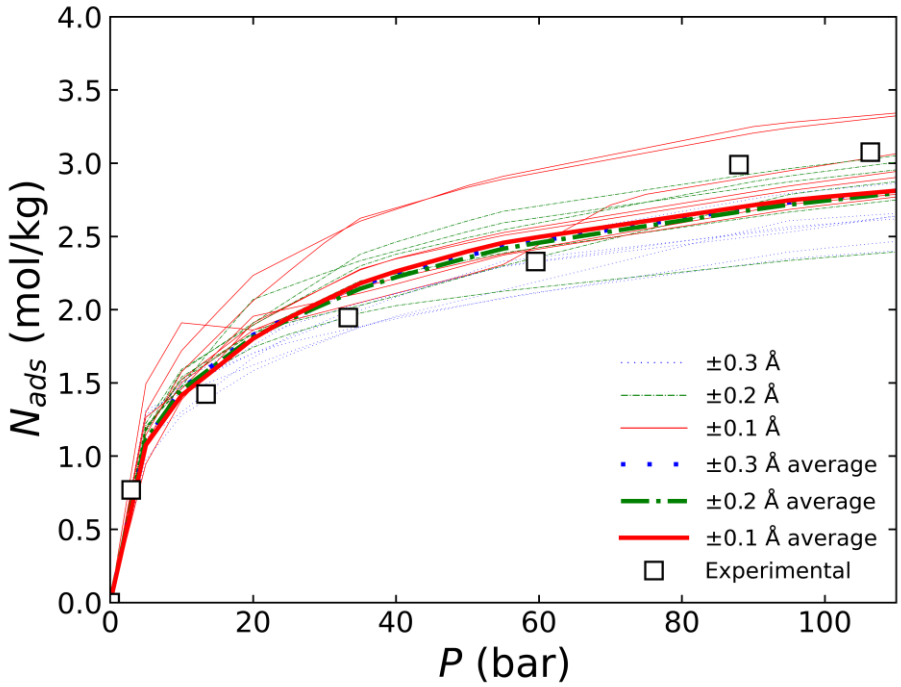

Figure 2.24 - Absolute adsorption of CH_4 in "**covalent angled GOF**" systems and their *ensemble averages* and experimental results[1]. See **Fig. 2.21** for color scheme[57].

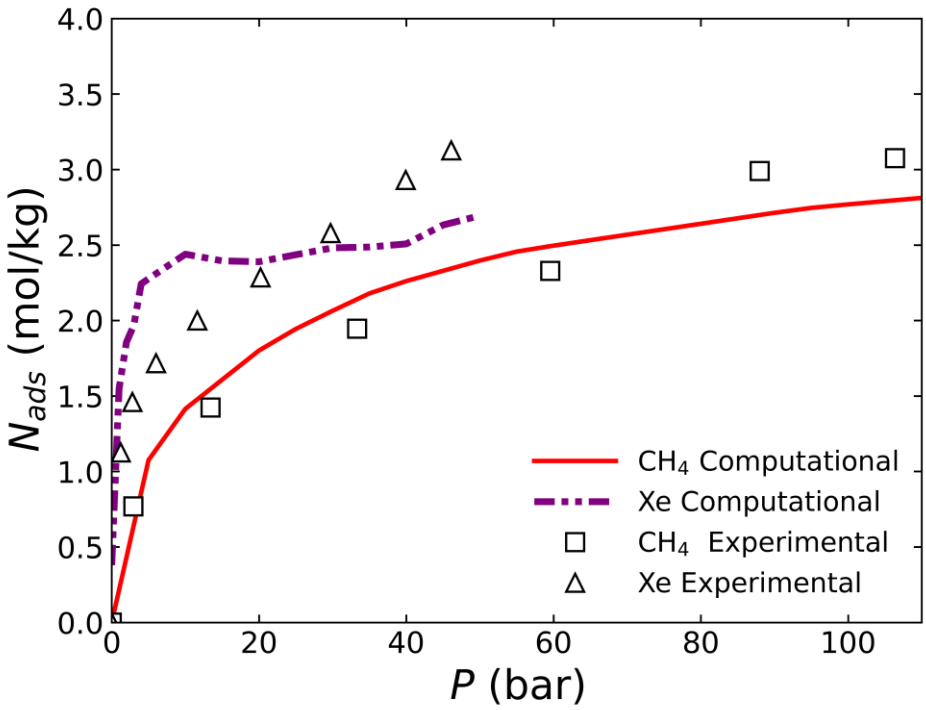

Figure 2.25 - Absolute adsorption of CH_4 and xenon in "**covalent angled GOF**" *ensemble averages* for the ± 0.1 Å tolerance level compared to experimental results[1,57].

The CH_4 adsorption isotherms follow the typical pattern of a strong initial uptake at low pressure followed by the gradual increase at higher pressure from pore saturation. In the case of the xenon adsorption isotherms, the uptake is even stronger, as the pore appears to saturate at even lower pressures than CH_4, which is reasonable given the stronger Xe-adsorbate interaction potential[87].

2.7 Discussion

Schaeperkoetter's results show that in supercritical conditions, the interlayer spacing d_{001} increases slowly and monotonically as a function of pressure for both CH_4 and xenon[1]. On the other hand, under subcritical conditions the structural changes fit with the sudden "gate-opening" transitions seen in other porous materials[78–80]. And given that a bulk phase change typically comes before a gate-opening transition for other

materials, the fact that Schaeperkoetter's experiments were completed at supercritical conditions means that there was almost certainly no bulk phase change, and we do not need to worry about whether we are looking for a gate-opening transition or not. Schaeperkoetter's pore expansion results for d_{001} indicate that van der Waals forces are the driving mechanism for the expansion since when the expansion and the calculated van der Waals forces are scaled together (plotted as spacing versus vdW force), they form a single curve[1].

Perhaps the main criticism of any of the covalently bonded models (CP-GOF and CA-GOF) is by Mercier *et al.*[52]: they posit that because the pore size swells up to 15.4 Å when in water, the DBA cannot be covalently linked as 12 Å would be the limit of the height[52]. However, an alternative explanation for the large swelling when flooded with a polar solvent may be boronate esterification, a well-studied reversible reaction[88,89] where a boron bonded to two oxygen and a diol, in this case carbon, will have its oxygens detach from a surface such as graphene in the presence of water, leaving behind OH on the surface as the boron's two newly detached oxygens each gain a hydrogen, all within a short time[89]. When the boronate's oxygens bond back with the surface, the reaction produces water. If no water is present, the boronate remains bonded to the surface. Therefore, flooding the pores with water could in theory detach all DBA molecules and then swell the pores to distances too great for the DBAs to bond to both sides. Conversely, continuous heat induced evaporation of water in a GOF will result in DBAs being only covalently bonded to the surface. In conclusion, it is possible for the covalently bonded models to be consistent with pore sizes swelling up to 15.4 Å.

The results of our covalent angled model support non-perpendicular covalent bonding sites for the DBA molecules. Because the *angle* and *dihedral* molecular degrees of freedom (**Eqs. 2.3, 2.4**) are involved in the expansion—much "softer" than the *bond lengths* (**Eq. 2.2**)—this enhances the flexibility of the framework and allows the expansion of the d_{001} spacing of the GOF. Although current diffraction techniques cannot verify the orientation of the DBA molecules, we suggest that the small changes to DBA-graphene bond site positions proposed in our covalent angled model are viable. The results suggest that the structure of the GOF is made up of DBA pillars randomly linked to the carbons of the graphene sheets. This implies that the GOF may not form a perfect three-dimensional network but may have some dispersion. Schaeperkoetter *et al*.[1] observed GOF crystallite domains sizes of ~13.7 nm, a decrease of ~20% from GOs, which may support this picture. A more detailed analysis of how the crystallite domain size varies with the incorporation of DBA pillars and during adsorption may help clarify this point.

In conclusion, we have simulated the adsorption of three existing and one new models of GOF-DBA and find that only the model "**covalent angled GOF**" is compatible with the observed swelling of the d_{001} spacing during supercritical adsorption of CH_4 and xenon. This should help the development of new structured porous materials for high performance adsorption of gases and catalytic reactions.

This is research is published in *ACS Omega* titled "*Adsorption-Induced Expansion of Graphene Oxide Frameworks with Covalently Bonded Benzene-1,4-diboronic Acid: Numerical Studies*"[57].

Chapter 3: Investigation of Adsorption Selectivity and Coadsorption of Methane and Carbon Dioxide in Graphene-Oxide Pores

3.1 Introduction

Hydrocarbon adsorption in activated carbon (AC) has captured the imagination of scientists for decades now. An experimental cornerstone from 1980 by Reich *et al.*[90] investigated the adsorption of CH_4, ethane and propane on activated carbons using various mixtures of the three, and established adsorption isotherms at low pressures[91]. Walters *et al.* [92] later studied the adsorption of heavier hydrocarbons which helped establish models such as Langmuir's and Toth's[87,93,94].

When heterogeneous natural gas (NG) is physisorbed onto AC's, the heavier hydrocarbons will naturally have stronger van der Waals interactions and push out the smaller gas molecules[87]. Over time, if the AC is filled and refilled, a residue of heavy gases may remain but their adsorption ultimately reversible[62].

Presently, our experimental partners at Missouri S&T are studying co-adsorption isotherms of CH_4 and CO_2 in AC, with promising prior research on 3D-printed AC monoliths[95]. They are currently studying adsorption and desorption of CH_4-CO_2 mixtures in fuel tank protypes with CH_4-CO_2 separating membranes. Our goal at the University of Missouri was to provide detailed insight at the atomic level, which is not available from experiments alone. We achieved this by performing extensive molecular dynamics (MD) and grand canonical Monte Carlo (GCMC) simulations of the co-adsorption of CH_4 and CO_2 in AC's.

As part of the process of modelling our partner's isotherms, we have investigated how different functionalizations of an AC can lead to improvements in the selectivity of the adsorption of CH_4 or CO_2. Given the relatively polar nature of CO_2 (which has a substantial electric quadrupole) vs. the completely non-polar character of CH_4 (or even propane, which we also studied), we expected that there is potential to optimize AC's selectivity by changing the adsorbent's surface with the addition of polar groups such as epoxy oxygens and/or hydroxyl (OH). We, thus, performed simulations (MD, Grand Canonical Monte Carlo) of the adsorption of said gases for slit-shaped pores of various sizes and different oxidation levels. Our results can be used to determine what type of pore is the best at adsorbing CO_2 or CH_4, and how to best produce cycles of adsorption/desorption that enable purification of gases from mixtures that are common in renewable biogas produced from organic waste (see **Section 1.1** Motivation

3.2 Molecular Dynamics Simulations

In our simulations we utilized the slit-shaped pore model of activated carbons[96] which places graphene sheets in parallel separated by distance *H* hereby dubbed "pore-size". Note that for us *H* is the distance between the planes corresponding to carbon atom-centers; this is different from typical experimental results which refer to pore-size as *available* space (i.e., they differ by the van der Waals diameter of a carbon atom ~ 3 Å). For our setup, three layers of graphene comprised of 946 carbon atoms/layer, interatomic distance a = 1.42 Å, and of dimensions $(33\,a, 21\sqrt{3}\,a) \approx$ (46.86 Å, 51.65 Å) are stacked in the *z* direction in separate runs where pore separation H = 8, 10, 15, and 20 Å. We used periodic boundary conditions (PBCs) in all three dimensions using a

simulation box size of L_x = 33 a, L_y = 120 Å, and L_z = 3 H. We tested both pristine graphenes, and graphene oxides (GO) comprised of 30 epoxies and 30 hydrogen-monoxide (1,036 total atoms/layer), and 100 epoxies and 100 hydrogen monoxides (total 1,246 atoms/layer). In the MD simulations the simulation box extends beyond the size of the graphene in the *y* direction to allow interpore diffusion in that direction and to have a substantial gas reservoir far from the substrate to determine the gas pressure from an undisturbed gas density[31,36,97]. Molecules whose center of mass lie at least 12 Å outside the graphene sheets are labeled in the gas phase while centers of mass inside the graphene are labeled adsorbed. Molecules within 12 Å of the pore are ignored to ensure the molecules in the adsorbed and gas phase are not interacting with each other since 12 Å is the coded interaction cutoff distance in our simulations.

With N_{gas} ranging from 50 to 1,000 molecules and separate simulations for each gas type, CH_4, propane and CO_2 were all placed at random positions and orientations at least 2 Å away from the structure and from any previously placed molecules. Given that the adsorbent is far more rigid than the adsorbate's motion, the position of the unoxidized carbons were fixed to improve computational efficiency (oxidized atoms are allowed to move since they typically have a hybridization state different than sp^2 and move out of the graphene plane). Epoxy O and hydroxyl OH groups are also mobile, and molecules are modeled fully atomistically. The only constraint used was that C-H and O-H bond lengths were kept fixed. Simulations are performed in the canonical ensemble (*N*, *V*, *T*) using the NAMD2 MD code[59]. The runs used a 1fs time step at a constant temperature with a velocity rescaling thermostat applied every 20 fs. Every simulation allotted time for 10^5 energy minimization steps to remove bad contacts, followed by 5-20 × 10^6 steps

of MD. Analysis of the different components of the energy (see **Eq. 2.1**) indicate that after ~ 0.5 ns the system appears fully equilibrated. However, based on our previous studies of CH_4, ethane, and propane in carbon nanopores[62] we noticed that a truly diffusive regime takes about 1 ns to be established, thus we skipped the first 2 ns of the simulations to have a large margin of safety with respect to equilibration.

We used the CHARMM22 force field and standard parameters[58], except for the epoxy and OH groups whose characteristics (bond lengths, angles, dihedrals and their respective constants, and partial charges—see **Eqs. 2.2-2.5**) for which we performed *ab initio* density functional theory calculations using Gaussian 09[72]. See **Figure 2.10** in **Section 2.2** and **Tables A1-5** in **Appendix A** for the interaction parameters used in our simulations[66].

3.3 Molecular Dynamics Results and Discussion

We calculated adsorption isotherms using the last 3 ns of each simulation for H = 8, 10, 15, and 20 Å pore sizes of graphene and the two GO variations. For all gases, a smaller pore size results in a strong initial uptake of molecules at low pressures due to the significant overlap between the adsorption potential of both faces of the pore[62], but also results in a fast saturation (see **Figure 3.1**). As the pore expands, this initial uptake weakens, and total capacity increases. In general, we observe a substantially larger uptake of CO_2 than of CH_4 in the activated carbons. This is due mostly to the larger van der Waals interaction between CO_2 vs CH_4 and the graphene layers.

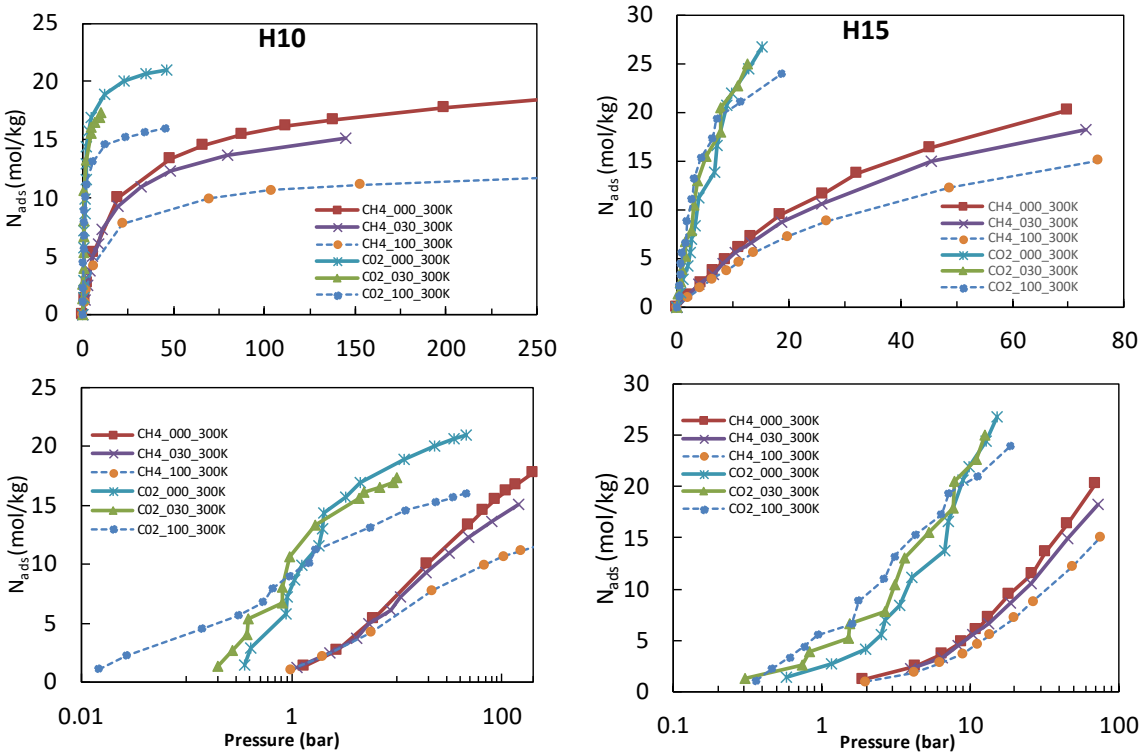

Figure 3.1 – Adsorption isotherms for homogeneous CH_4 and CO_2 inside of the 10 Å and 15 Å sized pores. The bottom semi-log scales are included to help visualize the low-pressure regimes. Results for H = 8, 20 Å can be found in **Appendix B**, **Figures B16-B17**.

Methane: As the graphene is oxidized, we notice that CH_4 does not adsorb at a higher rate. The initial uptake is practically unchanged (**Figure 3.1**); at higher P when the pores become saturated, and particularly at high levels of oxidation (100 OH 100 epoxy per layer), the amount of CH_4 adsorbed is lower at high pressures. These can be easily explained by the fact that CH_4 does not interact particularly strongly with the functional groups, but that these reduce the pore volume.

Carbon Dioxide: For CO_2, the oxidization of the graphene enhances the adsorption, in some cases quite substantially! Different from the CH_4 adsorption at low pressures, the pores adsorb noticeably more CO_2 as more OH and epoxies groups are added. But like CH_4 adsorption, we see that at high pressures, less CO_2 is adsorbed for

the more highly oxidized graphenes. We hypothesize that a strong Coulomb interaction between the polar surface groups and the CO_2 electric quadrupole is responsible for this behavior.

We can calculate the enthalpy of adsorption from adsorption isotherms at two different temperatures (**Figure 3.2**) using the Clausius-Clapeyron equation:

$$\Delta H = \frac{RT_1T_2}{T_1 - T_2}\ln\left(\frac{P_1}{P_2}\right). \qquad (3.1)$$

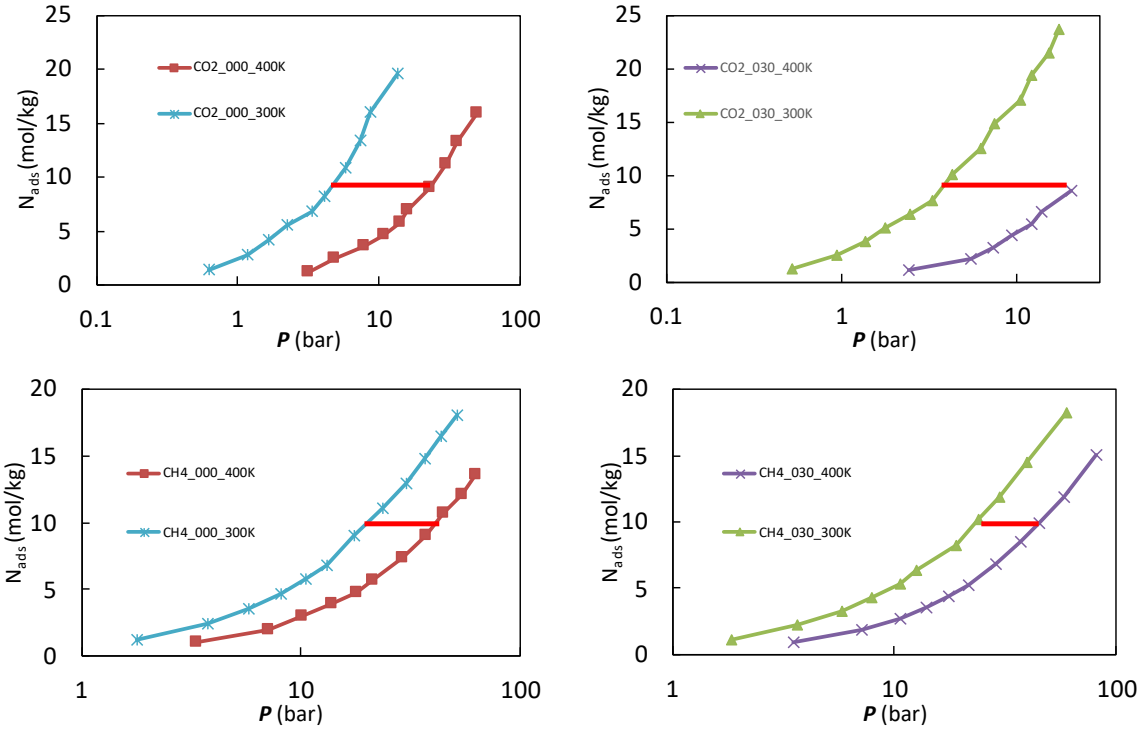

Figure 3.2 – Adsorption isotherms at 300K and 400K for CH_4 and CO_2 in graphene (left) and GO pores (right). The red line depicts of how $\ln(P_1/P_2)$ is measured to calculate the enthalpy of adsorption using the Clausius-Clapeyron equation.

Table 3.1 shows that that the enthalpy of CO_2 is consistently higher than that of CH_4. Significantly, whereas there is essentially no increase in ΔH for CH_4 when the polar groups are added, CO_2's ΔH is substantially enhanced, which is understandable since the

large electric quadrupole of CO_2 is expected to interact strongly with the polar epoxy and OH groups.

Table 3.1 – Average enthalpies of adsorption for CH_4 and CO_2 in various pore sizes with different levels of oxidization. The 30/30 pore type corresponds to pores that had 30 epoxies and 30 OH's attached to each graphene layer (i.e., approximately 1/16 of carbon atoms are bonded to an oxygen). For the 100/100 that amount is approximately 1/5 of carbon atoms.

CH_4	Graphene (kJ/mol)	30/30 (kJ/mol)	100/100 (kJ/mol)	CO_2	Graphene (kJ/mol)	30/30 (kJ/mol)	100/100 (kJ/mol)
H08	21	23	23	**H08**	32	36	39
H10	15	16	17	**H10**	24	26	29
H15	11	12	12	**H15**	18	18	23
H20	7.6	7.4	10.7	**H20**	16	18	21

Mechanism: To better understand where the CO_2 and CH_4 are adsorbed in the pores, we calculated pair correlation functions $g(r)$ between the center of mass of the gas molecules and various pore elements: graphene C, epoxy O, and OH O (see **Figure 3.3**). Pair correlation functions are shown in **Figure 3.4-Figure 3.6.**

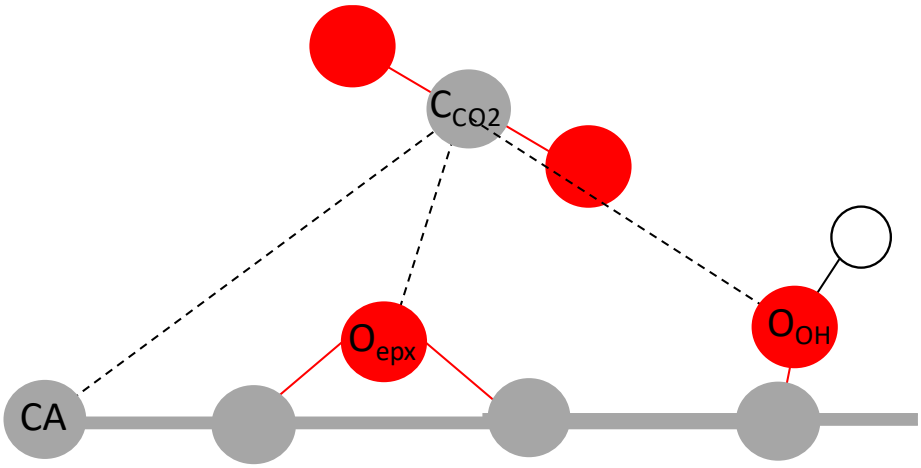

Figure 3.3 – A depiction of the various pairs (dotted lines) used to calculate pair correlation functions, using CO_2 as an example: C_{CO2}-CA (graphene carbons), C_{CO2}-O_{OH} (hydroxyl group), and C_{CO2}-O_{epx} (epoxy groups).

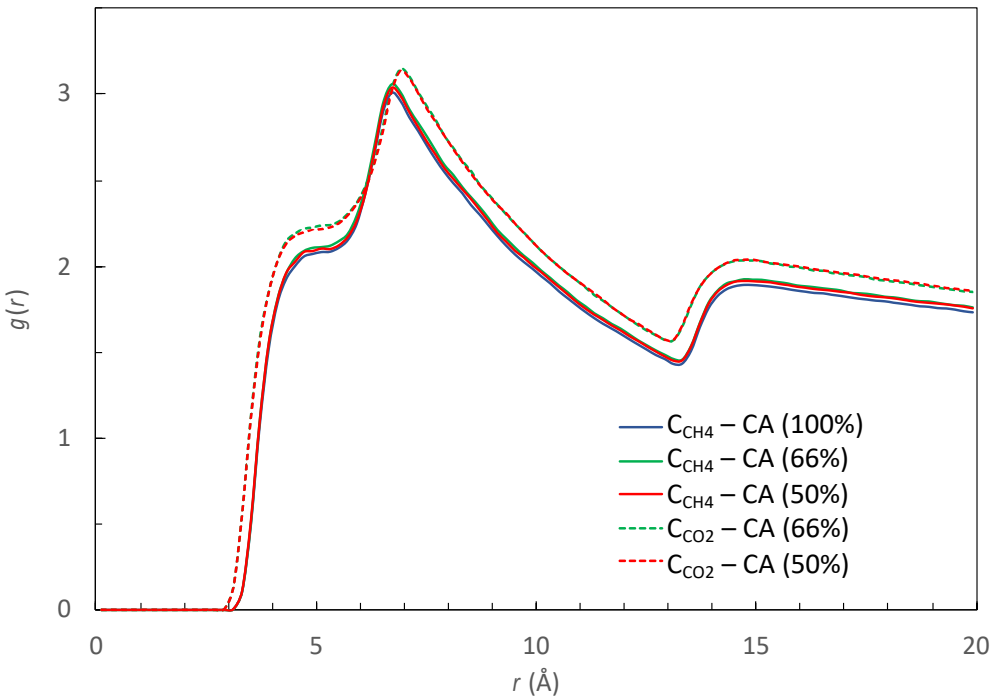

Figure 3.4 – Pair distribution functions between the CH_4 or CO_2 centers of mass (i.e., the C_{CH4}, C_{CO2} atoms, respectively) to "regular" graphene carbons CA. See **Fig. 3.3**.

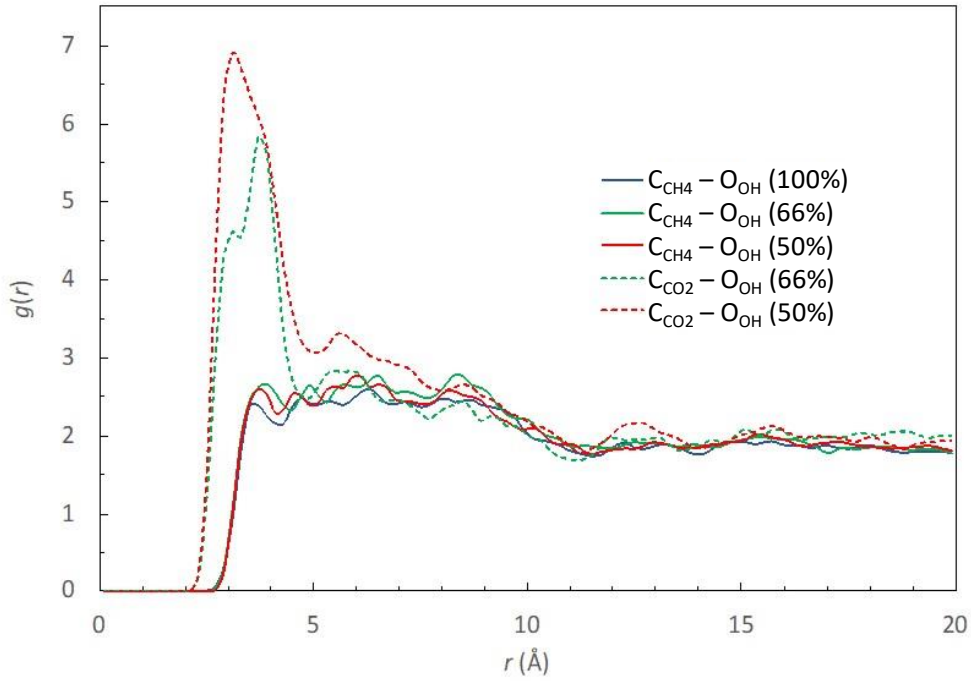

Figure 3.5 - Pair distribution functions between the CH4 or CO2 centers of mass and the oxygen in hydroxyl groups attached to the graphene. See **Fig. 3.3**.

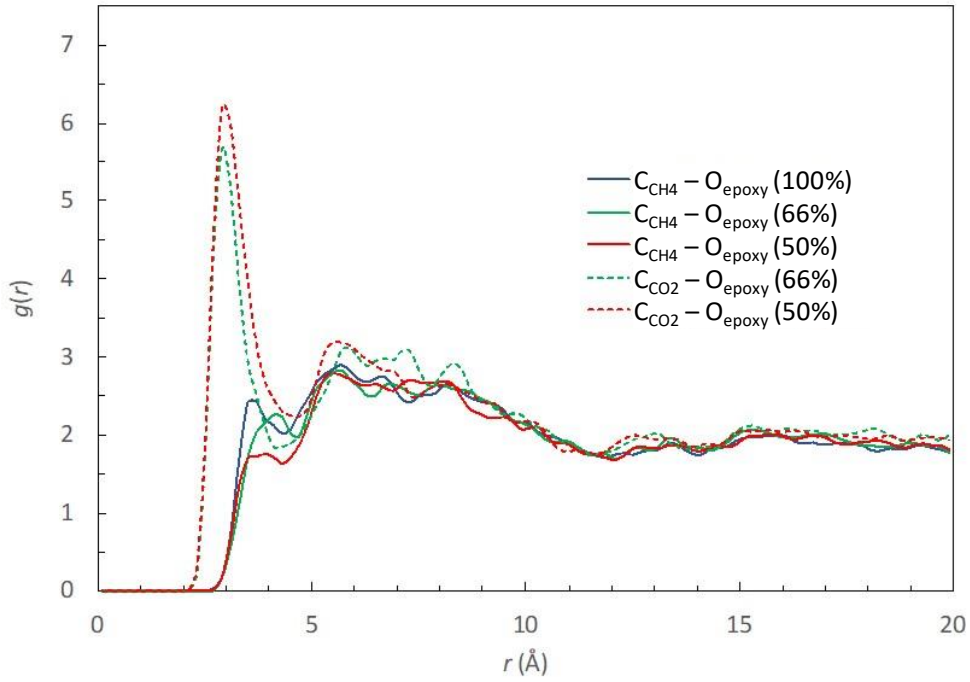

Figure 3.6 - Pair distribution functions between the CH4 or CO2 centers of mass and the oxygen in epoxy groups attached to the graphene. See **Fig. 3.3**.

We can see that the CO_2's center of mass (CO_2 being more linear) tended to adsorb slightly closer to the graphene surface than the CH_4 (**Figure 3.4)**. Because the graphene carbons (CA) are most numerous, adding polar groups did not affect these distributions significantly.

When tracking the interactions with the oxygens of the polar groups, we found that CH_4 interacted very weakly with the oxygens, with no marked correlation between these (**Figure 3.5** and **Figure 3.6**). Conversely, CO_2 clearly interacted quite strongly with all oxygens of the AC, as shown by the strong peaks in $g(r)$ (**Figure 3.5** and **Figure 3.6**). In fact, for the epoxy oxygens the presence of CO_2 even at low concentrations reduced slightly the CH_4's $g(r)$ at small r as the number of CO_2 was increased.

As mentioned above, we calculated estimates of the enthalpy of adsorption (isosteric heat) for both CH_4 and CO_2 by applying the Clausius-Clapeyron equation (**Eq**

3.1). The isosteric heat of CH_4 is practically constant as the graphene is oxidized, and we see that CO_2 isosteric heat increases as the graphene is oxidized. CO_2's strong reaction with oxidized graphene can be seen in the increased uptake of CO_2 at lower pressures in oxidized systems, the high density of CO_2 near oxygens, and the increase in isosteric heat as the graphene is oxidized. The strong attraction can be attributed to CO_2's attraction to the polar OH and epoxy groups, and its stronger vdW interaction attributed to its larger mass. CH_4 weakly interacts with the oxygen of AC as seen through its constant enthalpy between pure graphene and GO, the constant N_{abs} at low pressures, and the flat density plots. CH_4's weaker overall charge attributes to this behavior.

3.4 Grand Canonical Monte Carlo Simulations

In addition to MD, we performed grand canonical Monte Carlo (GCMC) simulations to analyze this system. A GCMC simulation runs in cycles, where it attempts to insert, delete, move, and rotate atoms or molecules with these attempts accepted or rejected with probabilities that lead to the expected equilibrium distribution functions[98]. The main advantage of the grand canonical ensemble is the presence of a "virtual reservoir" from which the molecules are brought into the system (insertion moves) or placed back (deletion moves). This means that there is no waste of computational resources with simulations of a large number of uninteresting molecules in the gas phase (as required in our MD simulations); thus there is no need for the additional space in the *y* direction in the computational cell that was needed previously. This reservoir is characterized by its chemical potential μ and temperature, or equivalently by its pressure *P* and temperature *T* (since experiments usually probe *P* and *T*, those are our control

variables). An added advantage is that we can now control the composition of the "feed gas" to our system, i.e, the total pressure and molar fractions for each component.

In our simulations we ran GCMC for 1,000 cycles of initialization and 5,000 cycles for collecting statistics. Each cycle consists of *N* steps, with *N* the number of molecules in the system. We set the temperature to 300 K for all simulations and simulated pressures ranging from 0.1 MPa to 10 MPa. We made a unit cell with the same area of our graphene sheet (21.3 Å x 17.2 Å) and the desired height of the pore (*H* = 8, 10, 12, 15, 20 Å). We then made the simulation box the size of 2 x 2 x 3 unit cells, resulting in 3-slit pores with graphene sheets the size of 42.6 Å x 34.4 Å x 3 *H* (see **Figure 3.7**) Interactions were cut off at 12 Å similarl to our MD simulations. Periodic boundary conditions (PBC) extended the simulation system "infinitely" in all directions.

For the graphene and GO, we used the same atom types and parameters as in our MD simulations, however we chose the TraPPE unified model for CH_4[99] to improve simulation speed and because its parameters are highly validated. For the same reasons of matching our MD results and being made for adsorption, we used the TraPPE force field definitions for CO_2[100]. **Figure 3.8**. compares adsorption isotherms for CH_4 and CO_2 in a slit-shaped pore using both MD and GCMC simulations; the results validate the two methods as essentially equivalent.

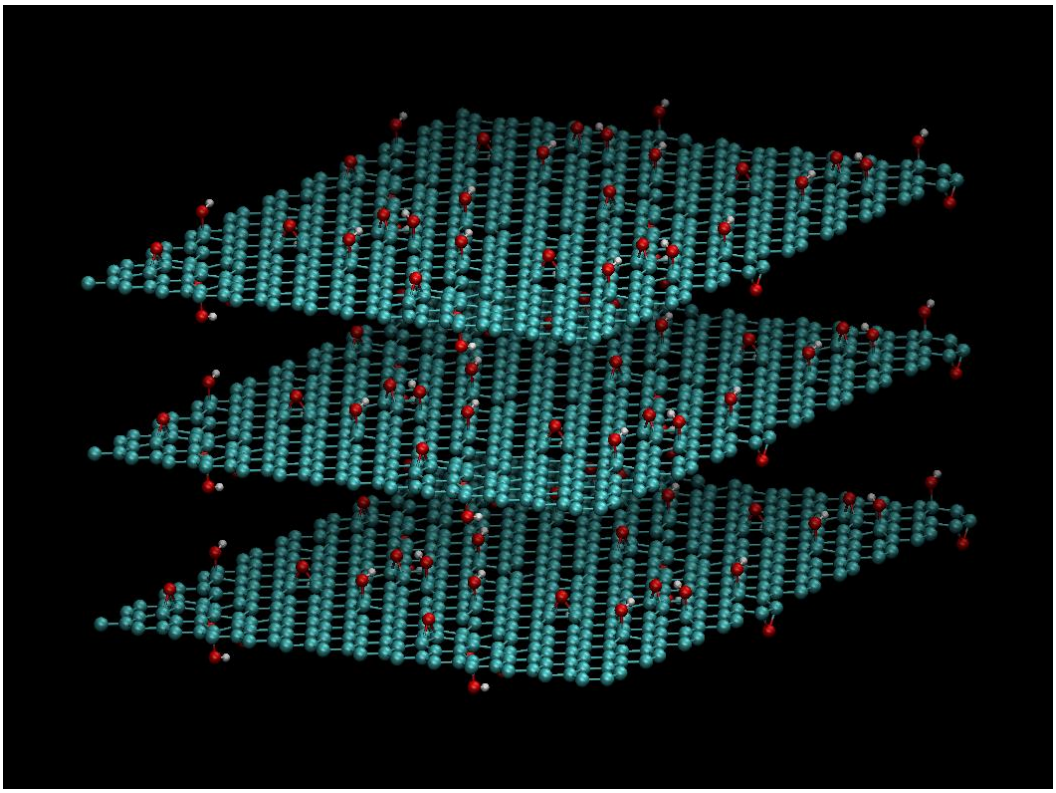

Figure 3.7 - A representation of the GCMC simulation box for GO bonded with epoxy and OH groups. The box dimensions are 42.6 Å x 34.4 Å x 3 *H*, where *H* is the pore size (10 Å in this example). PBCs are used in all directions.

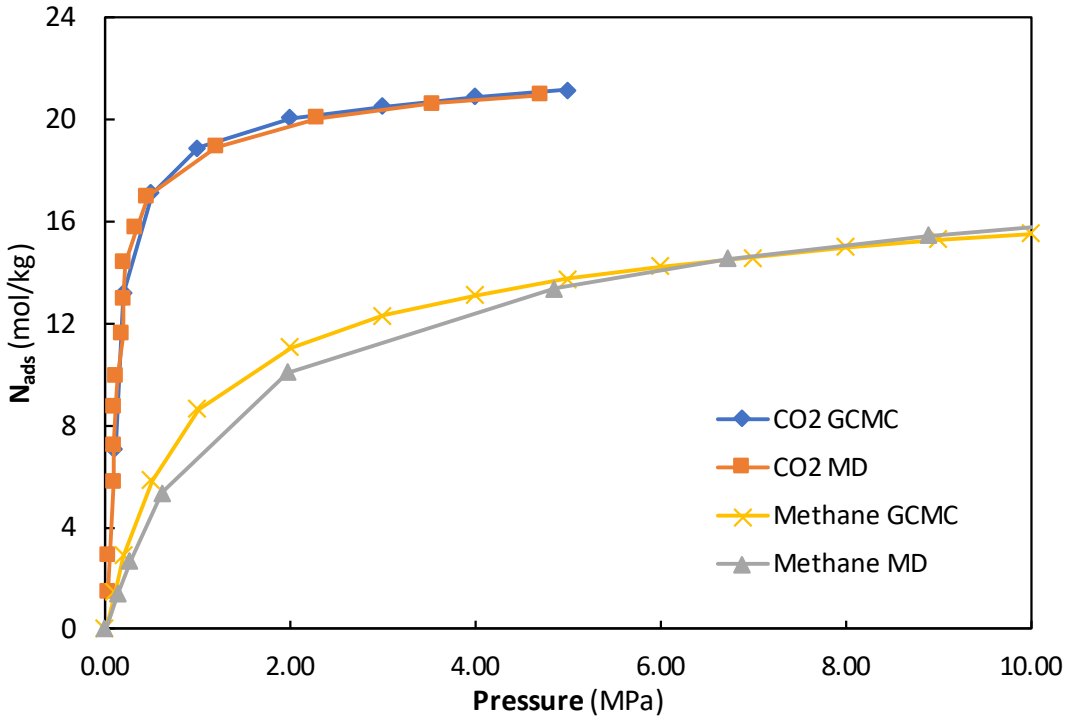

Figure 3.8 – Adsorption isotherms from GCMC and MD results in 10 Å graphene pores.

For the GCMC simulations we used the same non-bonded terms seen in **Eqs. 2.5** and **2.6** for MD. However, the pore structure was kept rigid in the GCMC simulations, besides a short MD simulation of the start to allow GO epoxy and OH groups, as well as their C neighbors to settle in to realistic configurations before the GCMC simulations started.

3.5 Grand Canonical Monte Carlo Results and Discussion

We simulated CO_2 and CH_4 mixtures adsorbing into GO and graphene at feed-gas concentrations/partial pressures ranging between 0-100% CH_4 (corresponding to 100-0% CO_2) with pores of sizes 8, 10, 12, 15, and 20Å. **Figure 3.9**-**Figure 3.12** depict adsorption isotherms of various feed-gas concentration for CH_4 and CO_2 in the l2Å pore. Each figure shows the adsorption of one molecule type, in a single type of pore for a sequence of concentrations and pressures. Adsorption isotherms for other pore sizes can be found in **Appendix B**, **Figures B11-B15**. A realistic AC, with a distribution of pore sizes, may be accurately simulated by combining simulations of single pore sizes[31].

The first observation that stands out is how CO_2 adsorption in both graphene and GO is largely unaffected by small increases in CH_4's partial pressure (**Figure 3.11** and **Figure 3.12**). On the other hand, even small increases in CO_2 partial pressure, reduces the uptake of CH_4 significantly (**Figure 3.9** and **Figure 3.10**).

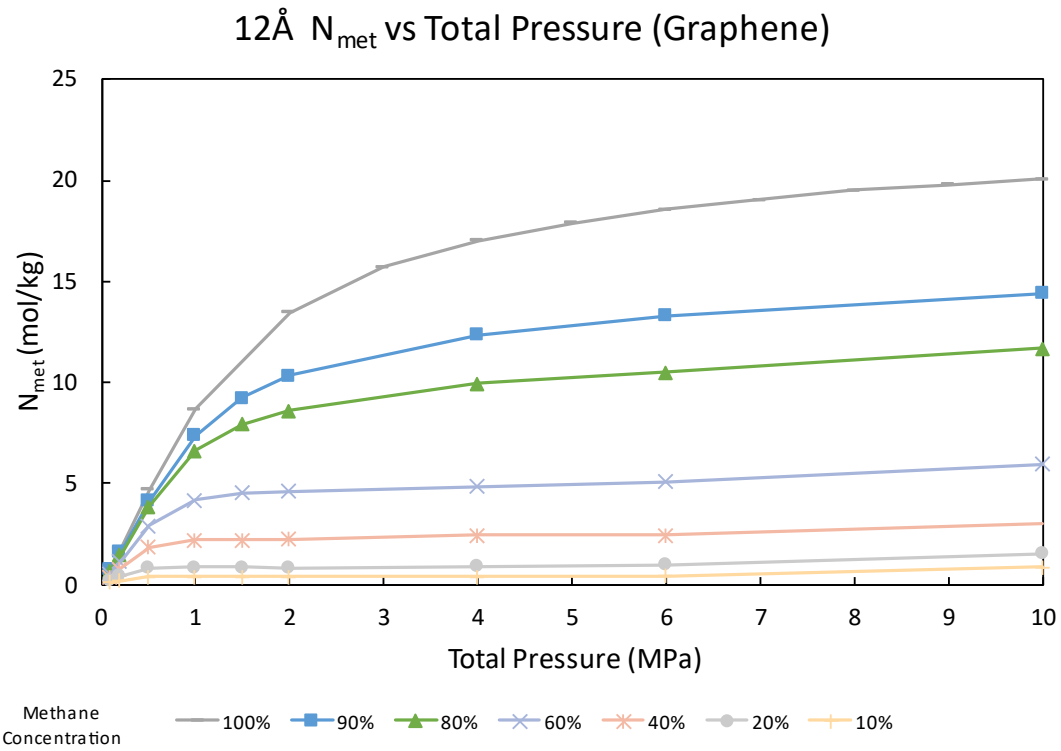

Figure 3.9 – Adsorption isotherm for CH_4 in a 12Å graphene pore. Notice how the introduction of just 10% CO_2 in the feed gas dramatically reduces CH_4 uptake. The colors of this graph and **Figure 3.10**-**Figure 3.12** are coordinated to reflect the same CH_4 concentrations, i.e., 90% CH_4 corresponds to 10% CO_2.

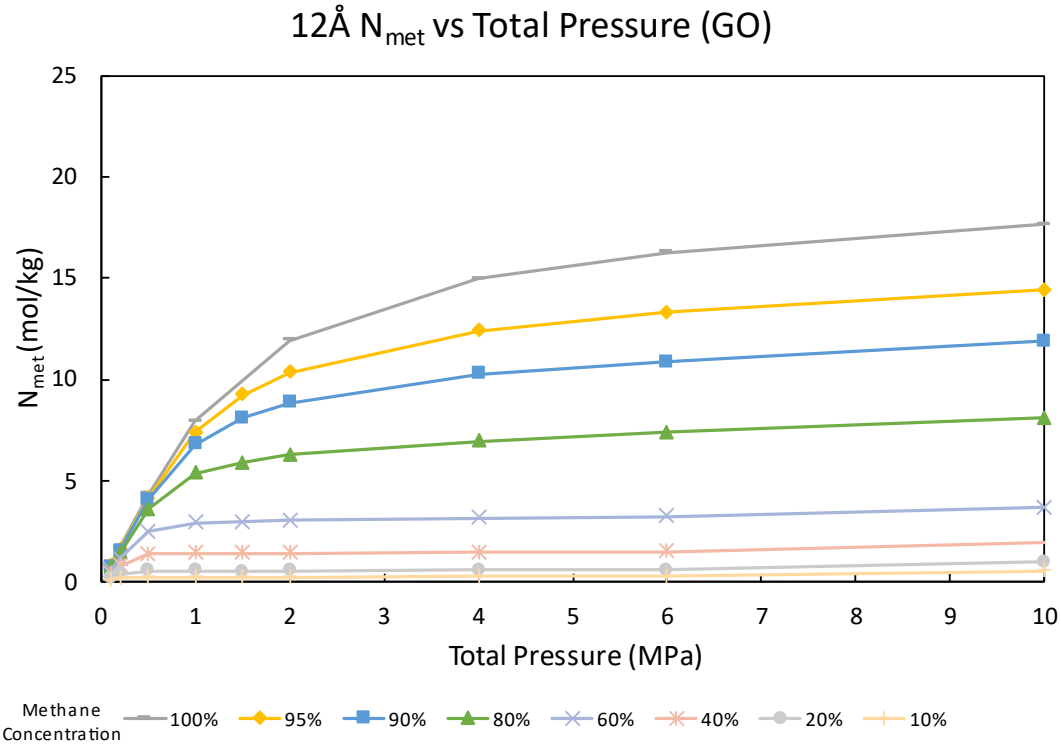

Figure 3.10 - Adsorption isotherm for CH_4 in a 12Å GO pore (color scheme: **Fig. 3.9**).

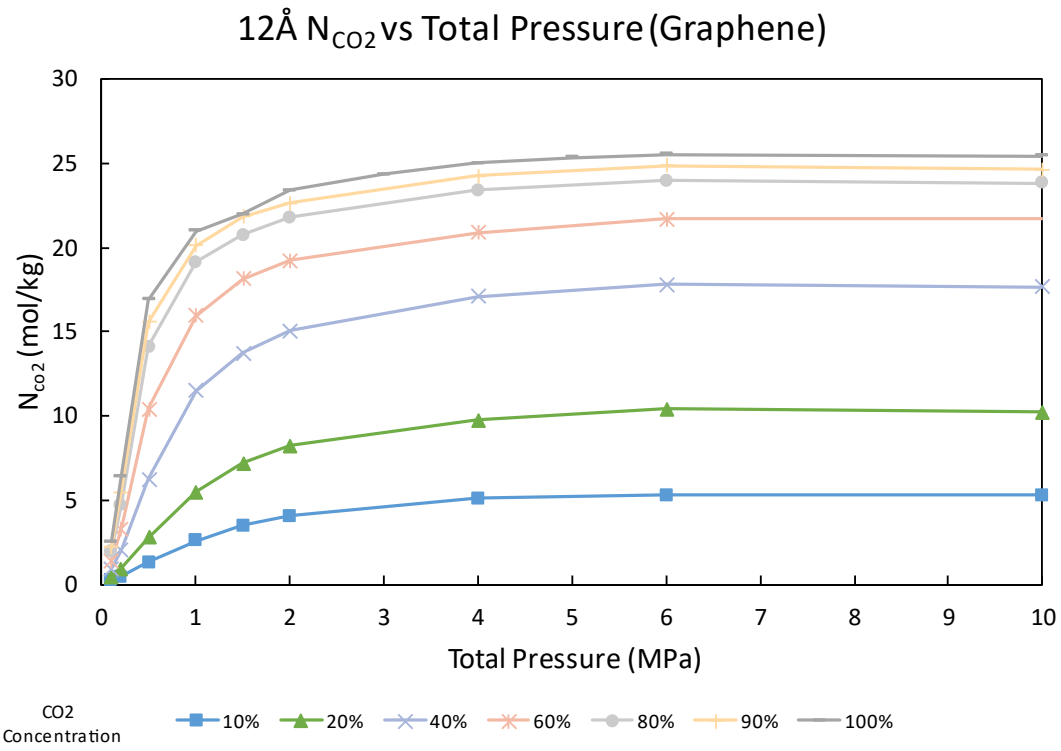

Figure 3.11 - Adsorption isotherm for CO_2 in a 12Å graphene pore (color scheme: **Fig. 3.9**).

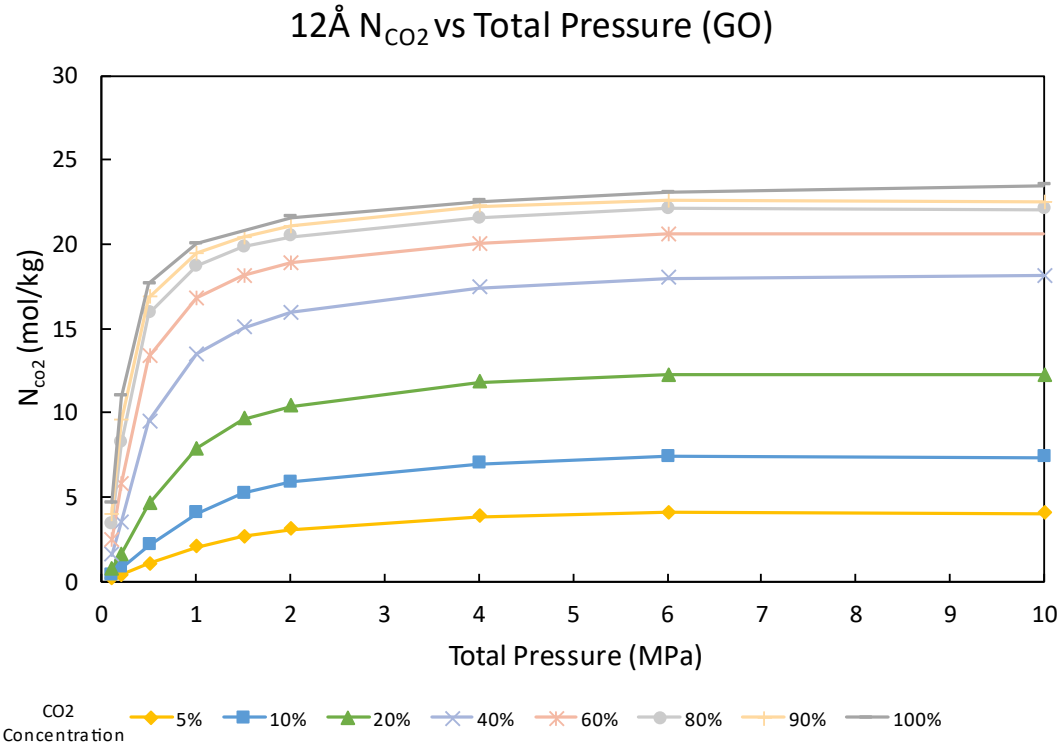

Figure 3.12 - Adsorption isotherm for CO_2 in a 12Å GO pore (color scheme: **Fig. 3.9**).

Figure 3.13-Figure 3.15 show in a more global form the adsorption isotherms for mixtures of CH_4 and CO_2 in a 10Å graphene pore for varying total pressures and gas composition (CH_4 molar fraction, or partial pressure). Looking at the surfaces created by these isotherms, one can visualize how even just a 20% decrease in the fraction of CH_4 pressure results in a 50% decrease in CH_4 molecules inside of the pores. These figures also help visualize that when CO_2 and CH_4 adsorbed into these pores, it was not in ratio to their partial pressures. Therefore, upon discharge, what comes out of the pores when desorbing is not what was put in. As an example, we placed yellow arrows in **Figure 3.13-Figure 3.15** to show that when feed gas was comprised of 60% CH_4, releasing the gas from 6MPa down to 1MPa results in a concentration that is approximately 30% CH_4. However, once this happens care must be taken since the desorbed gas composition varies from the original. Thus, if the process is done in equilibrium, one should migrate to a new composition isotherm, as shown by the dotted lines. In reality, of course, the process from start to end points follows process-dependent curves along the isotherm surfaces. See **Appendix B Figures B1-B5.2** for the rest of the adsorption isotherm surface plots and contour plots.

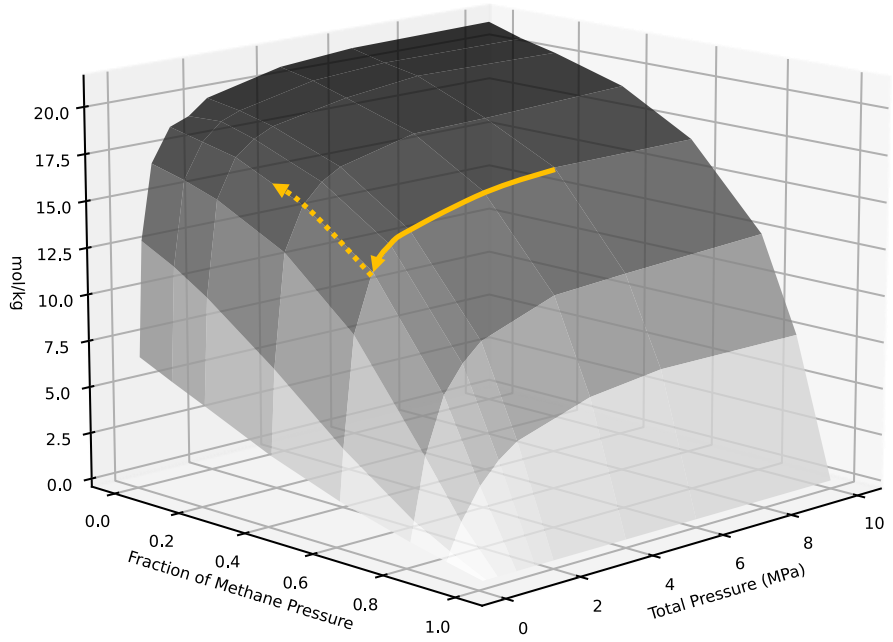

Figure 3.13 – CO_2 adsorption in 10Å graphene pores for various feed gas pressure and compositions. The solid yellow line corresponds to a desorption isotherm at fixed concentration. In most practical cases, during desorption the composition of the outflow gas changes, this is reflected by the dotted yellow line. In reality, the process would follow a process-dependent curve connecting initial and final points.

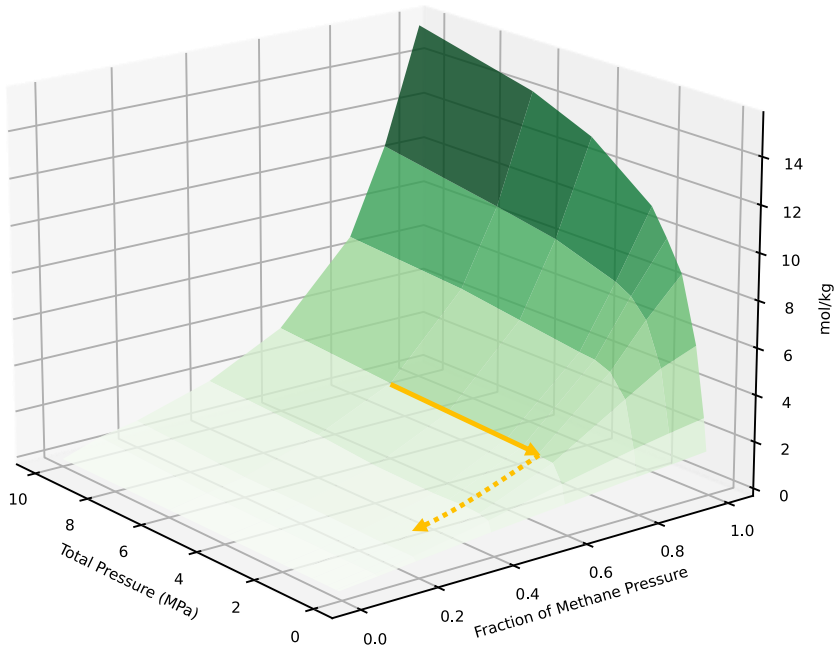

Figure 3.14 - CH_4 adsorption in 10Å graphene pores for various feed gas pressure and compositions. See **Figure 3.13** for information on the arrows.

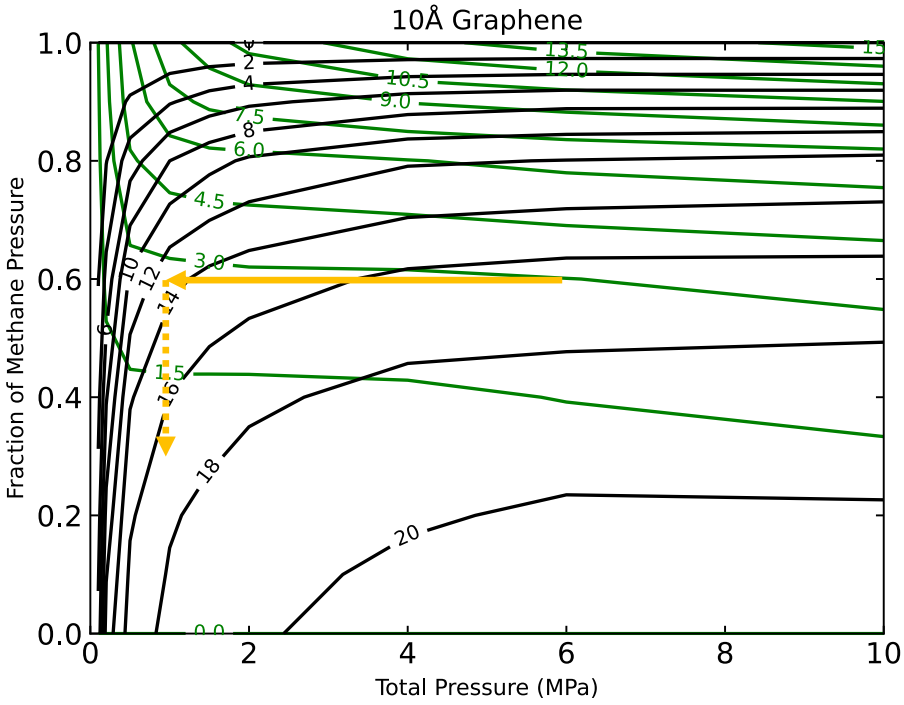

Figure 3.15 – Contour plot of the of the same data from **Figure 3.13** and **Figure 3.14**. See **Figure 3.13** for information on the arrows.

3.5.1 Selectivity of Adsorbents

Can the preferential adsorption of graphene and GOs be used to select one of the two gases from a feed gas? We explore this idea in this section. Here we can investigate how pore type, pore size, and initial and final pressures affect the ratio of the desorbed gas mixture, i.e., the selectivity of the material and process. **Figure 3.16**-**Figure 3.20** show how CO_2 concentrations of the desorbed gas depend on the various initial and final pressures given a starting feed-gas mixture of 60% CH_4- 40% CO_2. To get the final CO_2 concentration, we took the difference between N_{ads} for two pressures for both CH_4 and CO_2 and calculated what percentage CO_2 the released gas was. The amount of gas released would be the same as the change along the z-axis (mol/kg) depicted in **Figure 3.13** and **Figure 3.14**.

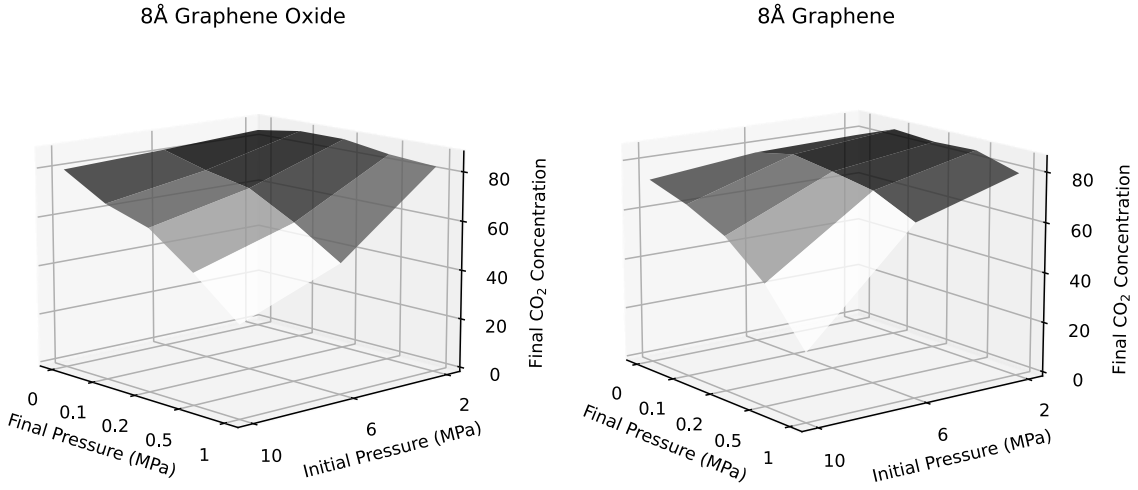

Figure 3.16 – Molar concentration of CO2 in the desorbed gas for various pressure changes for an 8Å graphene or GO pore when the feed gas during adsorption had a 40% CO_2, 60% CH_4 molar composition. Highest and lowest values are: GO maximum (max) = 87%, minimum (min) = 34%, Graphene 85%, min = 25%.

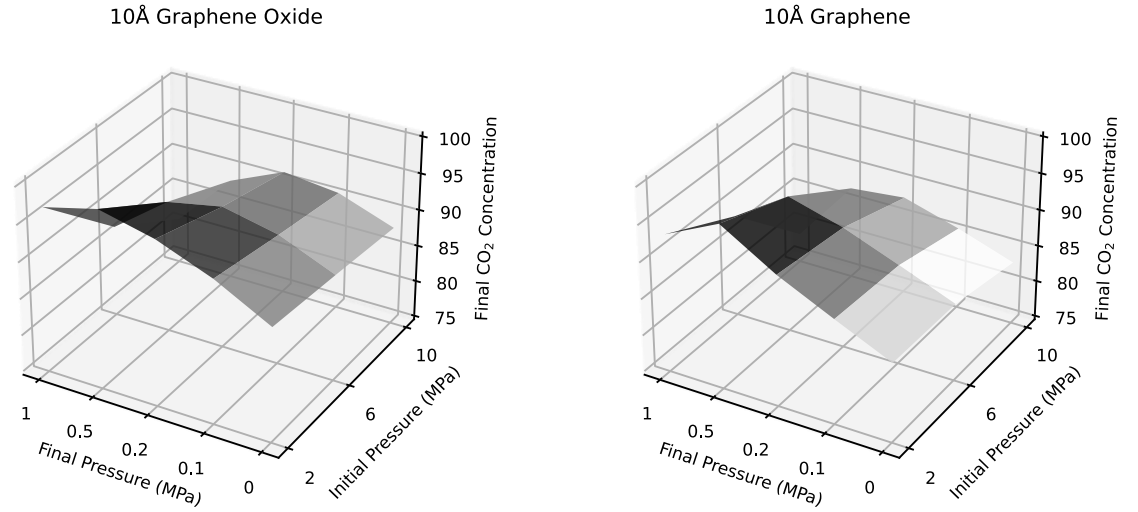

Figure 3.17 - Molar concentration of CO_2 in the desorbed gas for various pressure changes for an 10Å graphene or GO pore when the feed gas during adsorption had a 40% CO_2, 60% CH_4 molar composition. For GO max = 98.8%, and min = 80.1%. For Graphene max = 96.8%, and min = 78.5%

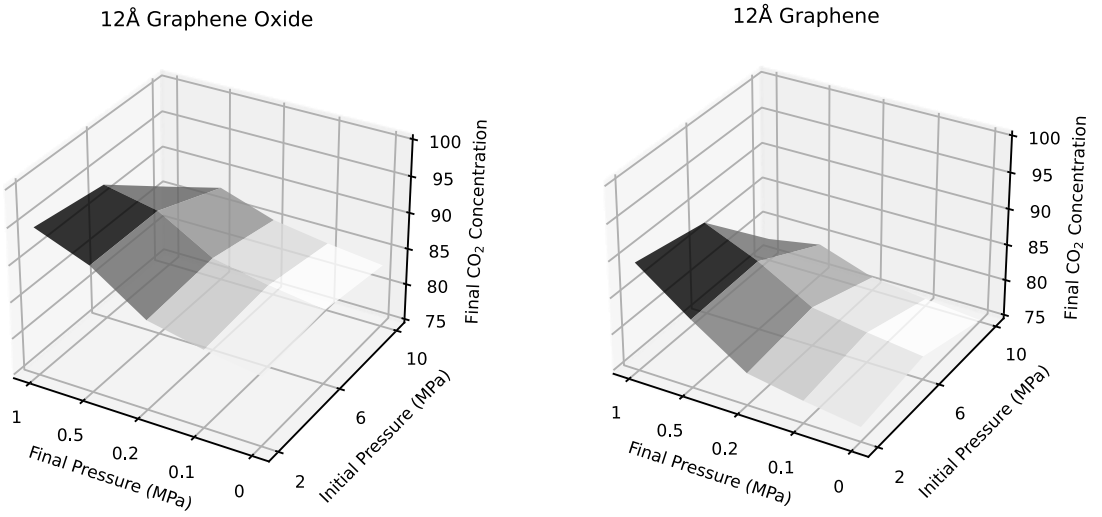

Figure 3.18 - Molar concentration of CO_2 in the desorbed gas for various pressure changes for a 12Å graphene or GO pore when the feed gas during adsorption had a 40% CO_2, 60% CH_4 molar composition. For GO max = 94.7%, and min = 83.1%. For Graphene max = 89.4%, and min = 74.7%.

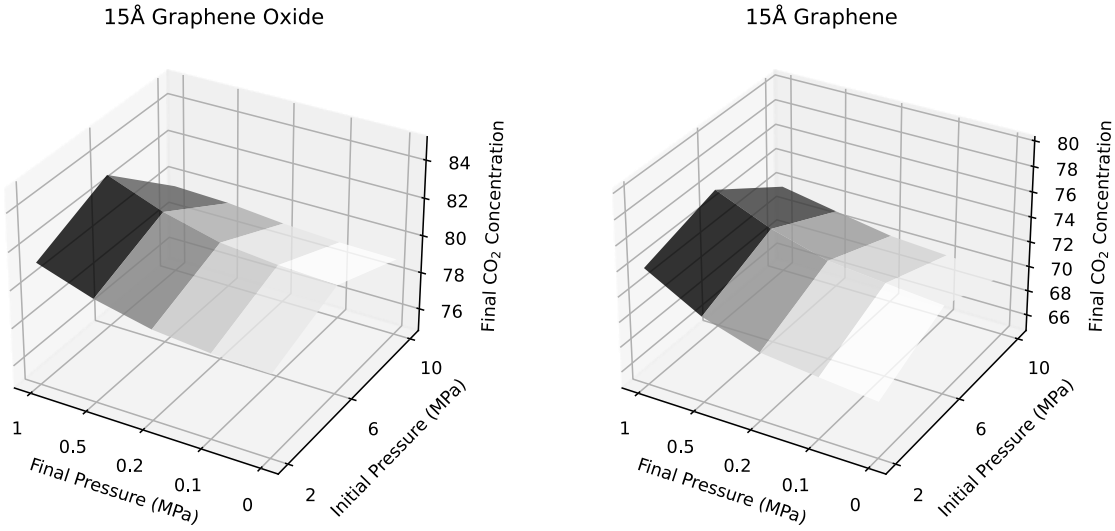

Figure 3.19 - Molar concentration of CO_2 in the desorbed gas of various pressure changes for a 15Å graphene or GO pore when the feed gas during adsorption had a 40% CO_2, 60% CH_4 molar composition. For GO max = 83%, and min = 78.8%. For Graphene max = 75.8%, and min = 69.1%.

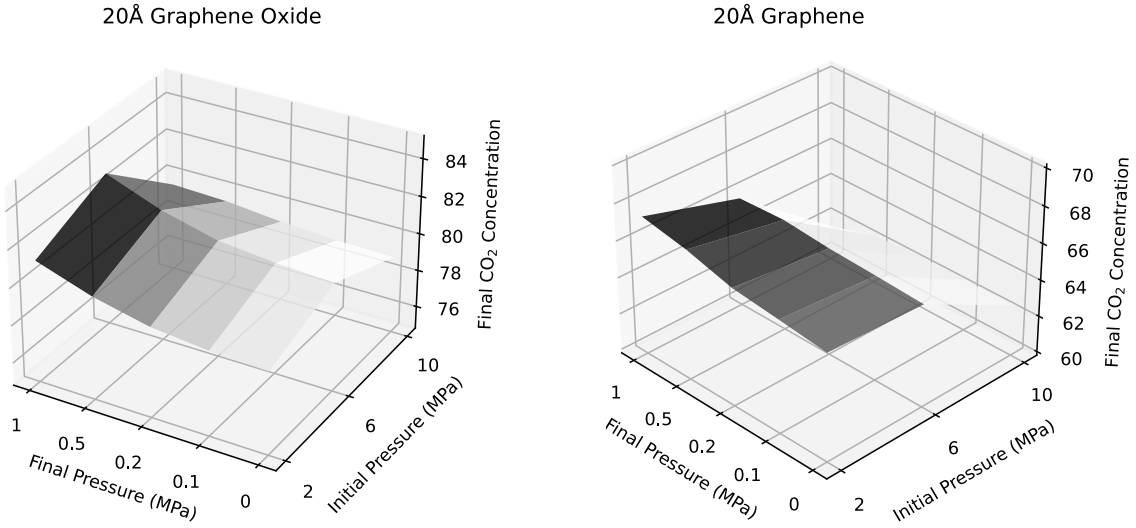

Figure 3.20 - Molar concentration of CO_2 in the desorbed gas of various pressure changes for an 20Å graphene or GO pore when the feed gas during adsorption had a 40% CO_2, 60% CH_4 molar composition. For GO max = 73.7%, and min = 65.4%. For Graphene max = 67.3%, and min = 61.7%.

When looking at which pore types are best at concentrating CO_2, it is evident that in all pore sizes GO performs better at concentrating CO_2. Also, the best initial pressures for concentrating CO_2 varies between 6MPa and 2MPa depending on the pore size, and the best final pressure for concentrating CO_2 is 1MPa for the pores 12Å or larger, 0.5MPa for the 10Å pore, and a total release down to 0MPa for the 8Å pore.

Figure 3.21-**Figure 3.23** show final CO_2 concentrations for various final pressures in different pore sizes with fixed initial pressures. What is immediately evident is that the 10Å pore consistently yields the highest concentration of CO_2 in desorbed gases in every scenario, meaning that regardless of initial pressure, final pressure, or pore type, the 10Å pore performs the best at concentrating CO_2.

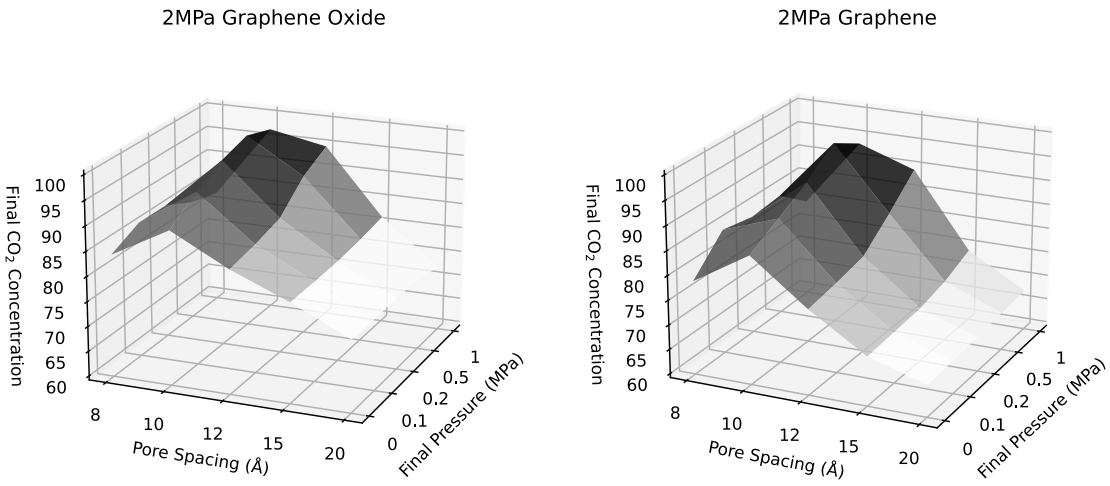

Figure 3.21 - Molar concentration of CO_2 in the desorbed gas for various pore spacings and 2MPa initial pressure which had a 40% CO_2, 60% CH_4 molar composition.

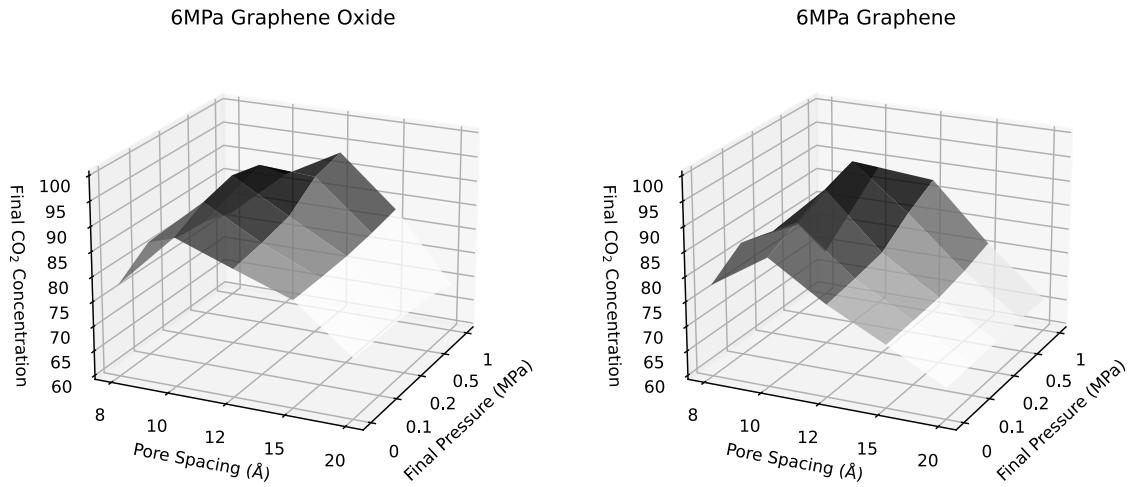

Figure 3.22 - Molar concentration of CO_2 in the desorbed gas for various pore spacings and 6MPa initial pressure which had a 40% CO_2, 60% CH_4 molar composition.

Figure 3.24-**Figure 3.26** depict the amount of CO_2 released during the desorption process. As expected, for higher initial pressures there is a positive correlation between pore size and CO_2 released. But at a lower initial pressure (2MPa), the 10 Å pore stands out by releasing the most amount of CO_2 in the graphene pore when desorbing all gas in

the pore, while the 12Å pore performs the best in GO. This is due to its sharper uptake at low pressures, as seen in **Figure 3.9**-**Figure 3.12** and **Figures B11-B15** in **Appendix B**.

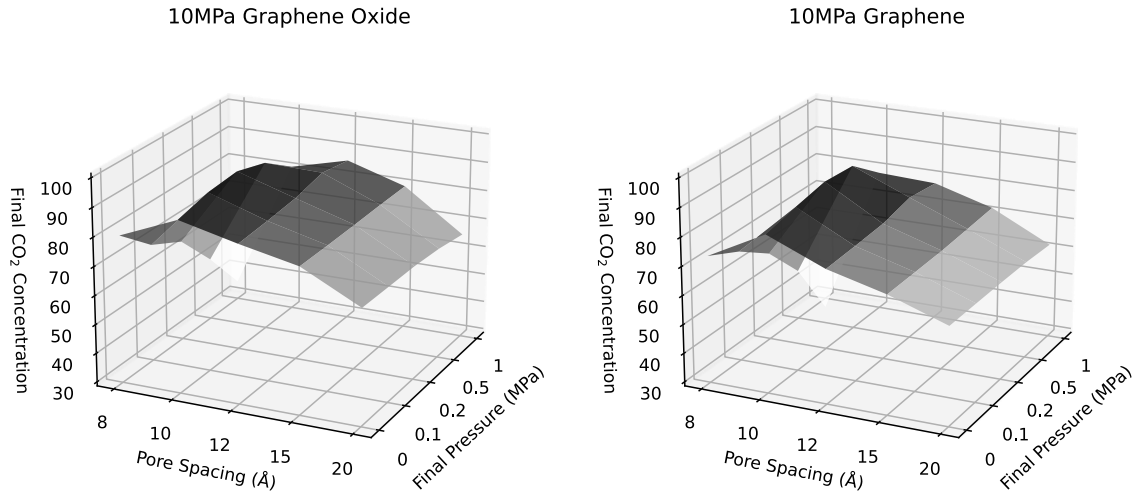

Figure 3.23 - Molar concentration of CO_2 in the desorbed gas for various pore spacings and 10MPa initial pressure which had a 40% CO_2, 60% CH_4 molar composition.

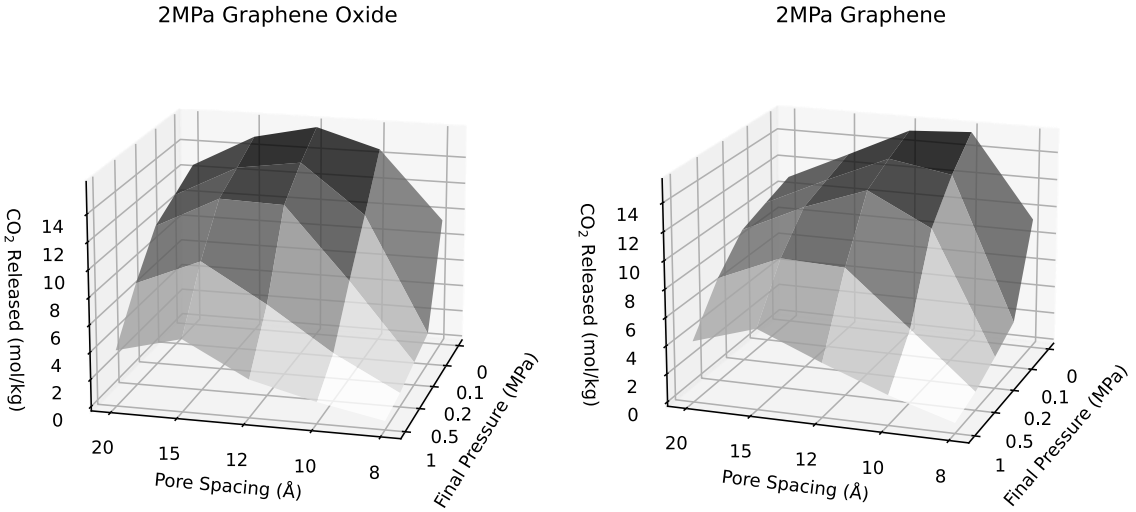

Figure 3.24 – Moles of CO_2 released in the desorbed gas for various pore spacings and 2MPa initial pressure which had a 40% CO_2, 60% CH_4 molar composition.

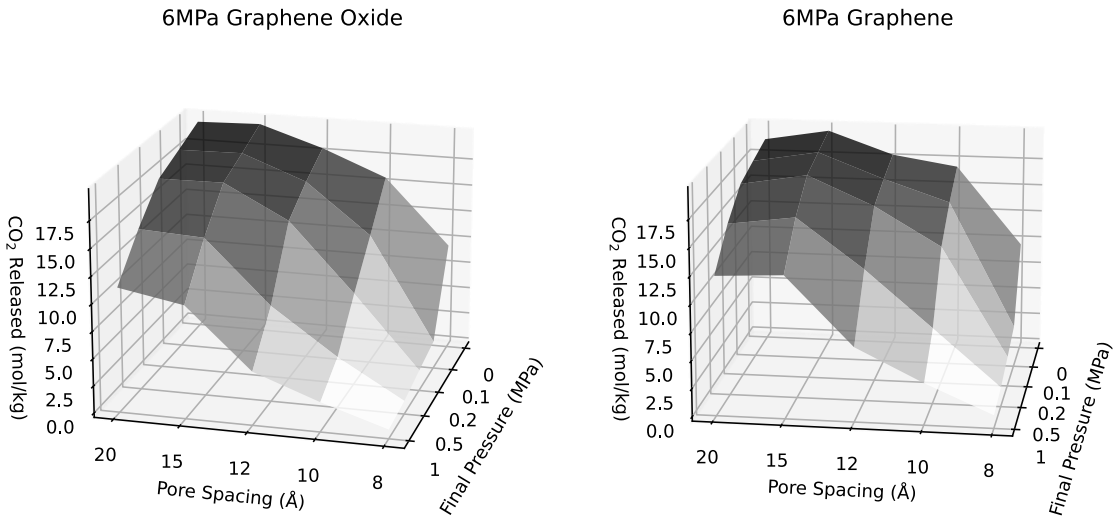

Figure 3.25 - Moles of CO_2 released in the desorbed gas for various pore spacings and 6MPa initial pressure which had a 40% CO_2, 60% CH_4 molar composition.

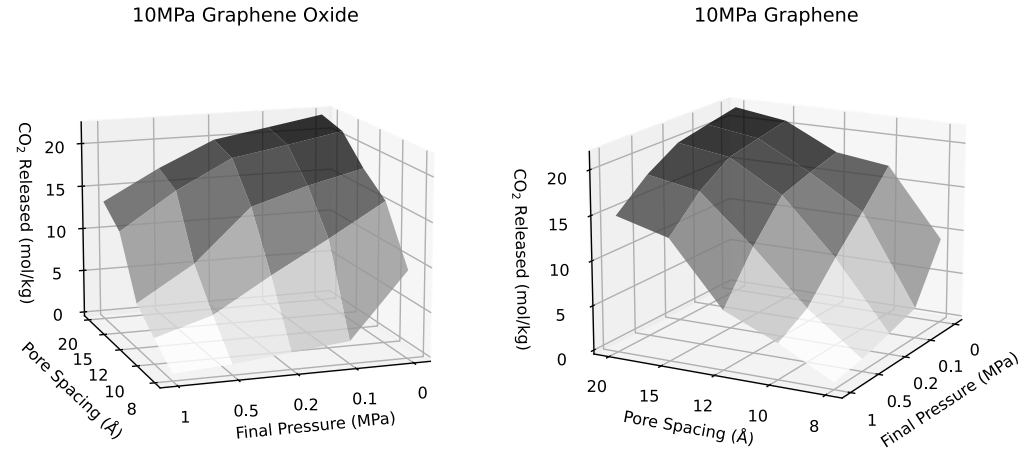

Figure 3.26 - Moles of CO_2 released in the desorbed gas for various pore spacings and 10MPa initial pressure which had a 40% CO_2, 60% CH_4 molar composition.

From inspecting pore size, oxidization, and pressure change, one can identify which combinations of pore types and pressure changes are optimal for concentrating CO_2. We need to keep in mind though that the desorption we discuss here is in equilibrium which consistently leads all pores to be CO_2 concentrators. It is possible that

in practical applications one could rapidly release gas from these pores and achieve higher concentrations of CH_4 due to its higher mobility and lower enthalpy of adsorption. First, GO is seen to be a better CO_2 concentrator in all cases. This is not surprising given CO_2's affinity for GO related to the added polar interactions. As for the optimal pressure change to concentrate CO_2, starting from 2MPa and stopping at 0.5MPa seems to work best. This is partly because CO_2 does most of its adsorption between 0 and 0.5MPa and plateaus at around 2MPa, whereas CH_4 continues to steadily adsorb more with pressure up to 10MPa. So going beyond 2MPa only favors concentrating CH_4 more.

Pore size also heavily influences the final CO_2 concentration. 10Å appears to be the optimal size for concentration of CO_2. This is likely because the non-negligible interaction range of polar groups in GO reaches a larger percentage of the 10Å pore. It appears that 8Å loses too much space with the addition of the surface groups, which lowers its performance.

3.6 Conclusions

We can see from the MD simulation's adsorption isotherms, enthalpy of adsorption, and pair-correlation functions that CO_2 has a strong affinity for GO such that it pushes out CH_4. The GCMC results corroborate with this fact in that it is found that GO is the best CO_2 concentrator in every pore size and pressure differential. In the end, after exploring various mechanisms for CO_2's adsorption in graphene and GO, we have found that a 10Å GO pore desorbing from 2MPa to 0.5MPa will yield the highest concentration of CO_2.

Although experimental results are yet to be published, our partners at Missouri S&T have found that when loading their ANG fuel tanks with approximately equal molar

ratios of CH_4 and CO_2, the gas that comes out always has a CH_4:CO_2 ratio between 1:3 and 1:4. This is quite consistent with our findings of final CO_2 concentrations between 60-80% for an initial 40% concentration.

Chapter 4: Helium Adsorption on Graphene Lattices

4.1 Introduction

Helium-4 (^{4}He) atoms deposited on solid substrates have been long identified as a possible manifestation of bosonic many-body quantum effects with a possibly very rich phase diagram[101–110]. Graphite, and more recently graphene, being almost homogeneous and atomically flat two-dimensional (2D) substrates[111], have been recognized as ideal scenarios for He to exhibit quantum many-body effects that are not present in other settings. In fact, experimental and theoretical studies have shown that a thin *superfluid* He film can form on graphite[102,112–116]. Although not many studies exist for *graphene*[107,117–120], it offers some significant advantages over graphite, namely a larger specific surface area (i.e., fewer C atoms that could interfere with scattering experiments) and the added possibilities to distort the substrate itself, e.g., by putting graphene on some dielectric substrate thus inducing strain, potentially leading to novel exotic phases[118].

He atoms being neutral and having the smallest electric polarizability in nature due to their closed 1s shell interact with the substrate and with other atoms via only very weak van der Waals (vdW) forces. These weak, but relatively long-range, interactions raise the possibility of superfluidity (note that in total *absence* of interactions, a Bose gas may form a condensate, but not a superfluid phase, something that is often confused). Under what conditions (i.e., temperature, film coverage) do different phases manifest themselves in this context? This will depend on the competition between the two-body He-He interactions, that of He-graphene, and temperature excitations. Since (as we shall see) the He-graphene interaction is the strongest, graphene also has a modest advantage

over graphite here: its adsorption potential is ca. 10% weaker, which helps reduce localization effects due to corrugation[121].

Thus, the problem of ^{4}He on graphene has become a relevant contemporary area of study, due to its potential to produce purely 2D collective bosonic phases. Existing theoretical studies[119,120,122–124] indicate that the first layer of He adsorbed on graphene forms an insulating solid commensurate state where all He atoms occupy 1/3 the graphene hexagon centers (which is the energetically preferred location, as will be explained later), see **Figure 4.1**.

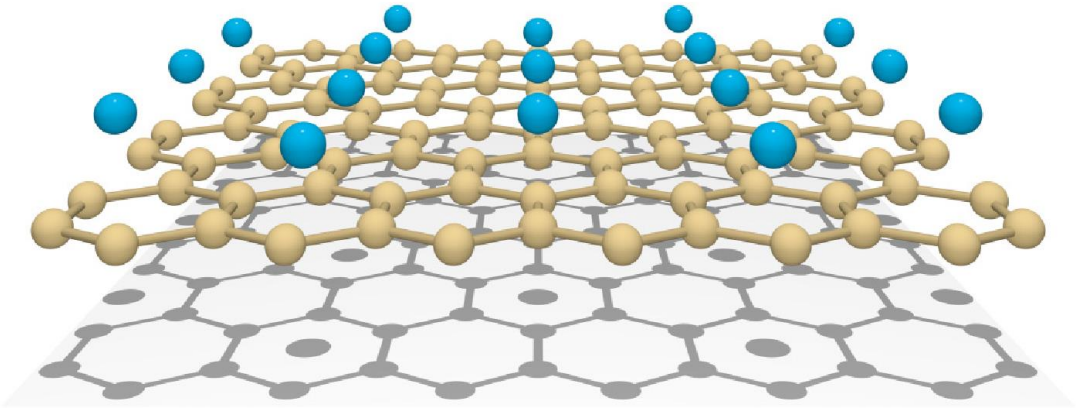

Figure 4.1 – A depiction of ^{4}He (blue atoms) in its $\sqrt{3} \times \sqrt{3}$ commensurate phases over graphene (gold atoms) which corresponds to 1/3 of the graphene center sites being occupied[125].

There are, however, some hints of the possible existence of a competing liquid-phase at zero temperature[122], and the existence of these phases depend crucially on the exact form and intensity of the He-graphene and He-He potentials. This is because the He solid shown in **Figure 4.1** is not due to *only* the corrugation of the He-graphene potential, but also (crucially sometimes) to the He-He repulsion itself, which is strongly repulsive for small distances (again a reflection of the closed 1s shells of He). Overall, the possible

emergence of superfluidity and other complex quantum phases is delicate and depends on the balance of weak vdW forces.

4.2 The Bose Hubbard Model

In work performed in collaboration with teams at the University of Vermont and University of Tennessee, Knoxville, we sought to elucidate how an effective low-energy description of a system comprised of a single layer of helium adsorbed on graphene can be studied using a Bose-Hubbard model (BHM). Application of such simplified model is desirable for the following reasons: **(a)** The results cited above regarding the existence of the 1/3 insulating solid state were determined by computationally expensive quantum Monte Carlo (QMC) methods, which although highly reliable, do not allow for the development of a deeper understanding of what interactions are important. **(b)** The development of an effective BHM is highly non-trivial under the conditions of He on graphite. The reason is the closeness of the characteristic He lattice (a few Å) to the range of both the long-range attractive (several Å) and short-range (~ 2 Å) repulsive vdW forces. This means that before performing the calculations, it is not clear whether a BHM is even appropriate to this system (it is). **(c)** Finally, by having a BHM developed for He-graphene, variations such as the application of strain, or even consideration of different substrates may be considered without the need to perform full QMC simulations. In other words, the development of a reliable BHM description has the potential to permit in the future the prediction of interesting quantum phases in other systems.

Our main contribution to this effort was the determination of the effective parameters of such BHM from *ab initio* 2nd order Møller–Plesset perturbation theory

(MP2)[72] calculations, i.e., the determination of the parameters of the generalized BHM[126–128] which is employed to determine the phase characteristics of the system. For the first He layer considered here, the low-energy BH Hamiltonian is given by:

$$H = -t \sum_{<i,j>} \left(\hat{c}_i^\dagger \hat{c}_j + h.c.\right) + U \sum_i \hat{c}_i^\dagger \hat{c}_i^\dagger \hat{c}_i \hat{c}_i$$

$$+V \sum_{<i,j>} \hat{c}_i^\dagger \hat{c}_i \hat{c}_j^\dagger \hat{c}_j + V' \sum_{\ll i,j \gg} \hat{c}_i^\dagger \hat{c}_i \hat{c}_j^\dagger \hat{c}_j + \cdots \quad (4.1)$$

where $\hat{c}_i^\dagger, \hat{c}_i$ are creation and annihilation operators for ^{4}He bosons at lattice site *i*, *h.c.* denotes the Hermitian conjugate, <*i*,*j*> denotes summation over nearest neighbor sites, <<*i*,*j*>> next nearest neighbor sites, etc. The first term corresponds to the transport or hopping term (*t* is calculated from the He-graphene potentials determined below). The second term is an on-site attraction or repulsion potential, and for ^{4}He atoms on graphene it can be safely taken to be ∞ given that hard-core He-He repulsion for distances smaller than ~ 2.7 Å (**Figure 4.10**). In the case of ^{4}He on graphene, the third term is also repulsive for two occupied, neighboring sites (at distances ~ 2.42 Å, see **Figure 4.3** and **Figure 4.10**). We calculate *V explicitly* from *ab initio* simulations of two ^{4}He on graphene. Subsequent terms for next-nearest-neighbors are generally neglected due to the smallness of He-He interactions at distances > 4 Å, but our calculations (below) show a significant enhancement of this interaction on graphene relative to the vacuum He-He interaction (**Figure 4.10** and **Table 4.2**), and these terms may in fact generate additional important physics (e.g., if the lattice is tuned such that the *V* terms is made to vanish).

In the past the calculations of ^{4}He potentials on graphene or graphite were exclusively performed using density functional theory (DFT) methods. However, DFT is known to be inaccurate for the determination of weak vdW (dispersion) interactions[129].

DFT yields reasonable He-graphene interactions only because the parameters in the simulations are semi-empirically tuned to yield the correct response. However, by tuning for the He-graphene interaction, one generally fails to tune for the He-He interaction, especially as it is modified by the presence of the graphene substrate itself[130,131]. In fact, two studies (Burganova *et al.*[102] and Ambrosetti *et al.*[103]) show that in the case of DFT calculations, the results depend heavily on the functional and basis set chosen as well as the computational power available (cut-off ranges, etc.).

Here we want to consider the He-He interactions on graphene at a higher precision level, which will help provide the parameters for a complete extended BH model of the system. This is the reason we employ the much more computationally expensive 2nd order Møller–Plesset perturbation theory (MP2)[72], with a high complexity 6-31++G(d,3p) basis set[129,134,135]. The MP2 method is generally believed to be highly reliable for determination of weak vdW forces[129], and is generally considered a "silver standard", second only to the even more expensive (and prohibitively so in our calculations) "gold standard" Coupled Cluster (CC) method[129].

4.3 Quantum Phase Diagram

The BHM (**Eq. 4.1**), in the limit of infinite on-site repulsion (infinite U) and neglecting next-nearest interactions and beyond ($V' = 0, \ldots$) has been analyzed in mean-field theory[136,137]. **Figure 4.2** shows the phase diagram, which is in qualitative agreement with calculations performed using lattice Monte Carlo approaches for hard-core bosons[138–140]. At low filling of the layer (i.e., small chemical potential μ) three phases are observed: the commensurate C1/3 solid, a supersolid, and a homogeneous superfluid.

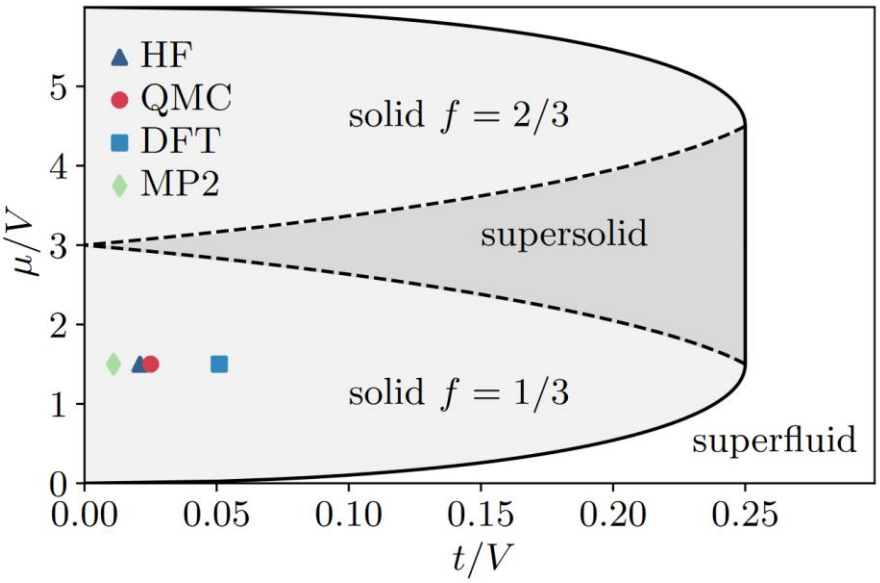

Figure 4.2 -Zero temperature mean-field phase diagram for hard-core bosons (infinite on-site repulsion *U*) on a triangular lattice with nearest-neighbor interactions *V* and hopping *t*. The density is controlled by the chemical potential μ. At filling factors $f = 1/3$ and 2/3 a commensurate solid is observed, whereas a superfluid and supersolid exist for various combinations of the hopping parameter *t* and chemical potential μ. Solid lines are 1st order transitions (discontinuity in various quantities such as density), dotted lines are 2nd order transitions. The data points shown in the lower left corner correspond to calculations of the BHM interaction parameters by various techniques[125] indicating that the ground state of a single layer of ^{4}He on graphene resides deep in the commensurate solid phase at 1/3 filling. The MP2 data point is part of the results we will be presenting below. For these points, μ has been tuned to coincide to the value at the tip of the first lobe[125].

4.4. 2nd Order Møller–Plesset Calculations

For our studies we first calculated the interaction potential between a single He atom and graphene by using sequentially larger aromatic carbons (benzene, coronene, hexabenzocoronene, circumcoronene)[118] using Møller–Plesset 2nd perturbation theory (MP2)[72] in Gaussian 09[72] using a variety of Pople-type basis sets. We first performed an extensive basis-set study because He's closed shell requires large basis sets to be able to have enough degrees of freedom for a wavefunction optimization. In the end, we found that the 6-31++G(d,3p) basis set was generally adequate. These basis sets incorporate

polarization functions p for s orbitals (needed for He's 1s electrons), and diffusion (++), which mean they are excellent at representing the various orbitals. However, they require very long calculations since in MP2, the computational cost scales with the number of degrees of freedom of the wavefunctions to the 5th power[129]. The aromatic systems were taken with C-C bond lengths of 1.42 Å to simulate graphene.

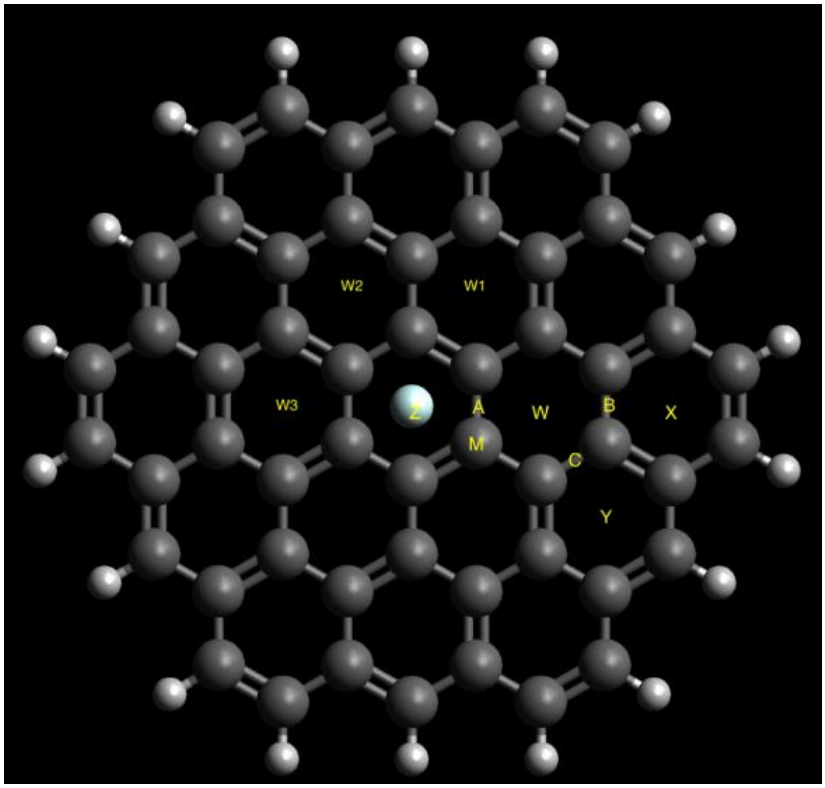

Figure 4.3 - The circumcoronene-graphene with the sites labeled where the He potentials were initially calculated.

To determine the various potentials for adsorption (the He-graphene interaction) we considered the adsorption of a single ^{4}He atom over various sites as depicted in **Figure 4.3** by bringing an atom from infinity perpendicularly to the circumcoronene and above the final adsorption site (Z, W$_i$, Y, X, A, M, see **Figure 4.3**):

$$E(z) = U(z) - U_{\infty} \quad (4.2)$$

where $U(z)$ is the MP2 energy of the moving He at height z, U_∞ is the energy of the configuration with the moving He atom at $z = \infty$. **Figure 4.4** shows $E(z)$ at the center points of benzene, coronene, hexabenzocoronene, and circumcoronene ("Z"), and for the first out-of-center local minimum ("W") for circumcoronene. The small difference between the curves for hexabenzocoronene and circumcoronene centers, and the small difference between the minimum at W and at Z indicate that circumcoronene achieved, to a reasonable accuracy, convergence to the graphene (infinite plane). Although it would certainly be desirable to consider larger aromatic molecules, the computational cost (5-20 CPU-days per point) is already quite high and would become prohibitive for larger systems since the cost scaling for MP2 methods is N^5, as mentioned above[129].

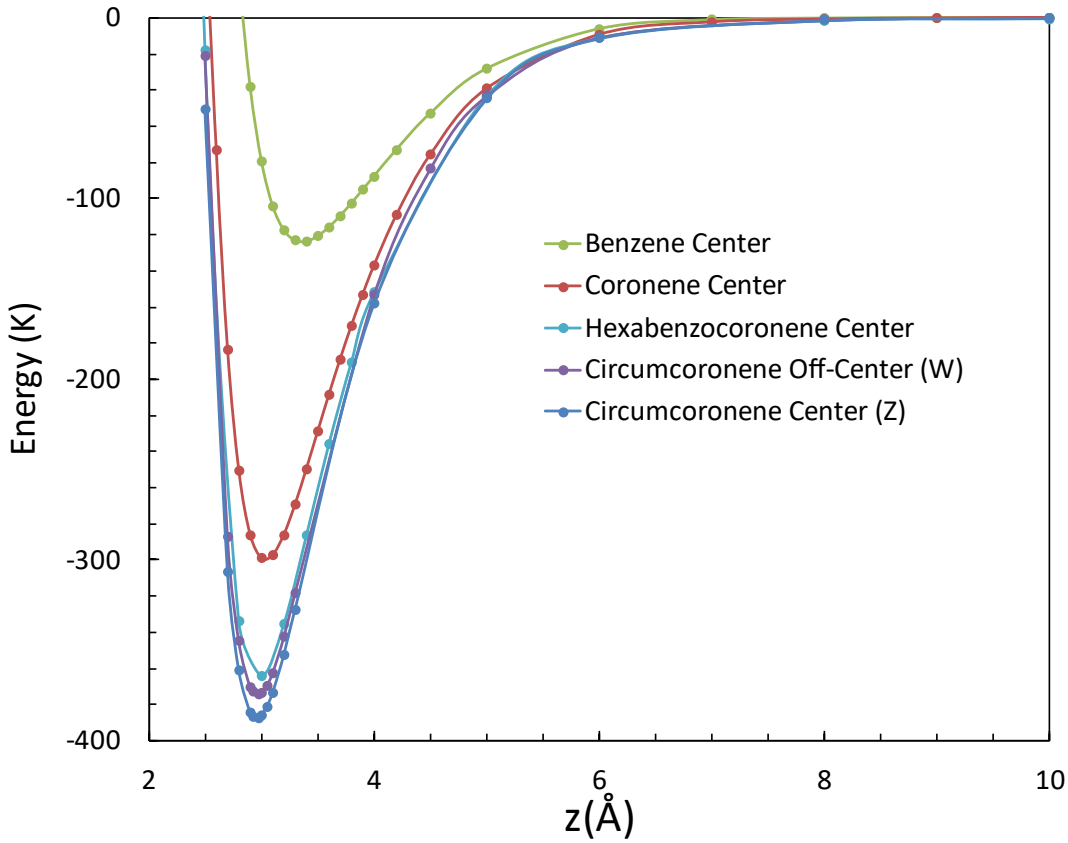

Figure 4.4 - A comparison between the adsorption potentials found over the centers of benzene, coronene, hexabenzocoronene and circumcoronene (Z) and the next local minimum for circumcoronene (W) calculated in MP2 using the 6-31++G(d,p) basis set.

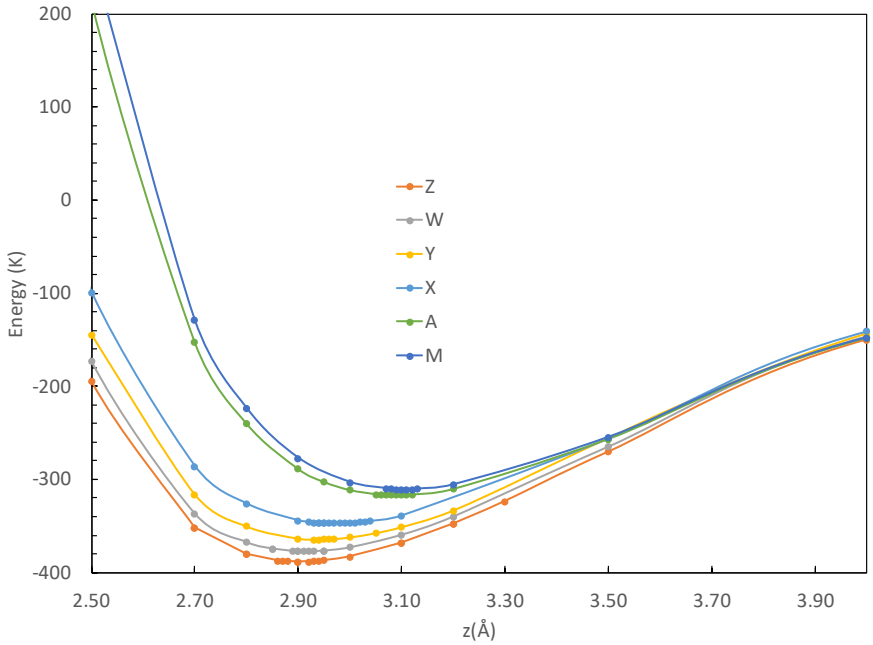

Figure 4.5 – Adsorption energy as function of height over various circumcoronene sites labeled in **Figure 4.3.**

Similar $E(z)$ scans were performed for circumcoronene above other important geometrically important points (Z, W_i, Y, X, A, M, see **Figure 4.3**): see **Figure 4.5**. The location of the minima (z_0, U_0) of $E(z)$ are listed in **Table 4.1**. **Figure 4.6** shows the dependence of z_0 and U_0 on the distance to the circumcoronene center point Z. Besides the saddle valley A and peak M (see **Figure 4.7-Figure 4.8**), there is a relatively small variation from Z to W, but this rapidly increases as we reach regions closer to the edges (beyond W's R = 2.46 Å) as the He "sees" less "graphene" around.

Table 4.1 - Distance to center R, height of minimum over site z_0, minimum adsorption potential U_0, and curvature of the potential around the minimum d^2E/dz^2.

Site	***R*** **(Å)**	z_0 **(Å)**	U_0 **(K)**	d^2E/dz^2 **(K/Å^2)**
Z	0	2.90	-388	665
A	1.23	3.09	-317	596
M	1.42	3.11	-310	506
W	2.46	2.92	-377	645
Y	4.26	2.94	-364	566
X	4.92	2.97	-347	576

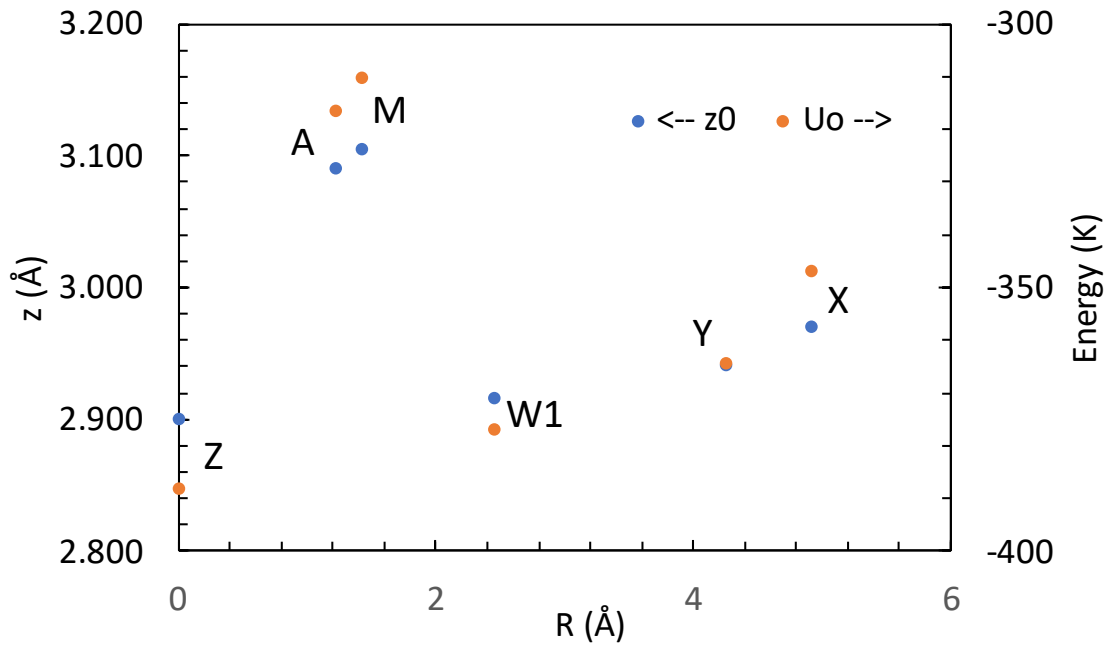

Figure 4.6 – Comparison of the height of He (z_0) at its lowest energy level (U_0).

Next, we performed scans that connected the two minima Z and W through the saddle point A, and from the saddle point A to the maximum M. After compensating for the overall "radial" dependence of the potential (**Table 4.1, Figure 4.6**), the lateral variation potentials are shown in **Figure 4.7** and **Figure 4.8**. The potentials are then fit to the standard Steele's 10-4-3 Lennard-Jones Potential for a particle on a periodic lattice[141,142]:

$$\mathcal{V}_{\mathrm{He-G}}(\boldsymbol{r}_i) = \varepsilon\sigma^2\frac{4\pi}{A}\left\{\left[\frac{2}{5}\left(\frac{\sigma}{z_i}\right)^{10} - \left(\frac{\sigma}{z_i}\right)^{4}\right] + \sum_{\mathbf{g}\neq 0}\sum_{\ell=1}^{2} e^{i\mathbf{g}\cdot(\boldsymbol{r}_i - \boldsymbol{b}_\ell)}\left[\frac{1}{60}\left(\frac{g\sigma^2}{2z_i}\right)^{5} K_5(gz_i) - \left(\frac{g\sigma^2}{2z_i}\right)^{2} K_2(gz_i)\right]\right\}.$$

Here $\boldsymbol{r} = (x, y)$ is the coordinate of a He atom in the xy plane, $\boldsymbol{b}_l$ are the basis vectors and $\mathbf{g}$ the reciprocal lattice vectors of the graphene (see ref.[125] for details). Most importantly, from the fits of **Figure 4.7** and **Figure 4.8** one can reliably map the entire effective potential for He-graphene interaction. Since the summation over $\mathbf{g}$'s is dominated by the

terms of smallest |g| (due to the exponential decay of the modified Bessel functions), one can approximate the *xy* dependent part of the potential as

$$V(r) = V_0 + c_{g1} \sum_{|\boldsymbol{g}|=g_1} e^{i\boldsymbol{g}\cdot\boldsymbol{r}} + c_{g2} \sum_{|\boldsymbol{g}|=g_2} e^{i\boldsymbol{g}\cdot\boldsymbol{r}} \qquad (4.3)$$

where $g_1 = 4\pi/(3a_0)$ and $g_2 = \sqrt{3}g_1$ make up the smallest set of reciprocal vectors. c_{g1} and c_{g2} can be calculated using the formulas:

$$c_{g1} = -\frac{1}{9}(V_{\max} - V_{\min}) \qquad (4.4)$$

$$c_{g2} = \frac{1}{8}\left(V_{\text{saddle point}} - V_{\min}\right) - \frac{1}{9}(V_{\max} - V_{\min}) \qquad (4.5)$$

A summary of the parameters calculated from the MP2 potentials, along with those of different methods[125] can be found in **Table 4.2**. **Figure 4.9** shows the *xy* dependence of the minimum energy of interaction between He and graphene.

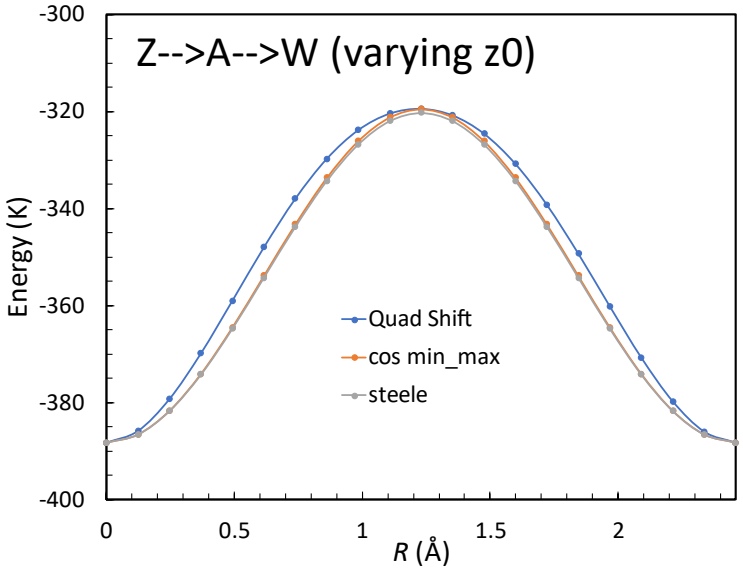

Figure 4.7 - Minimum adsorption potential for a Helium atom as it travels from site Z to site W.

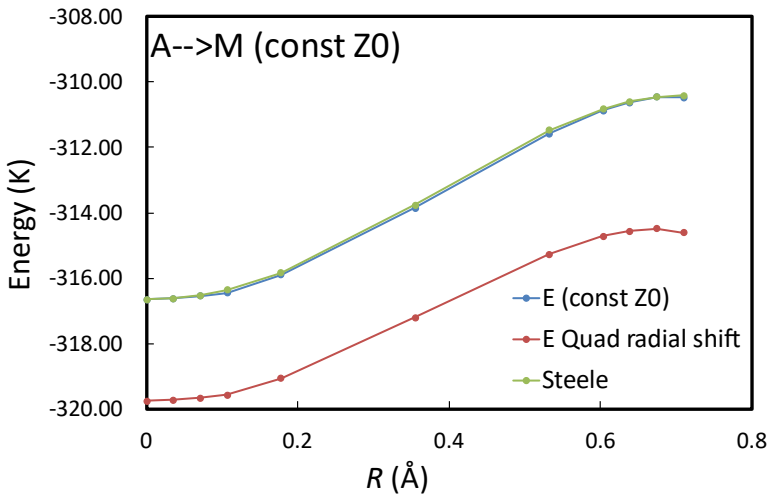

Figure 4.8 - The minimum energies along the path from site A to site M.

Table 4.2 –Parameters taken from the adsorption potentials for four different methods at high symmetry points corresponding to the minima, maxima and saddle points, used to calculate the coefficients of the Steele potential[125]. Hartree-Fock, Quantum Monte Carlo, Density Functional Theory calculations were completed by colleagues at the University of Vermont and the University of Tennessee, Knoxville.

Method	$V_{max} - V_{min}$ (K)	$V_{saddle\ point} - V_{min}$ (K)
Hartree-Fock	21.2	17.5
Quantum Monte Carlo	24.7	21.7
Density Functional Theory	39.2	36.1
Møller–Plesset second-order	72.2	66.0

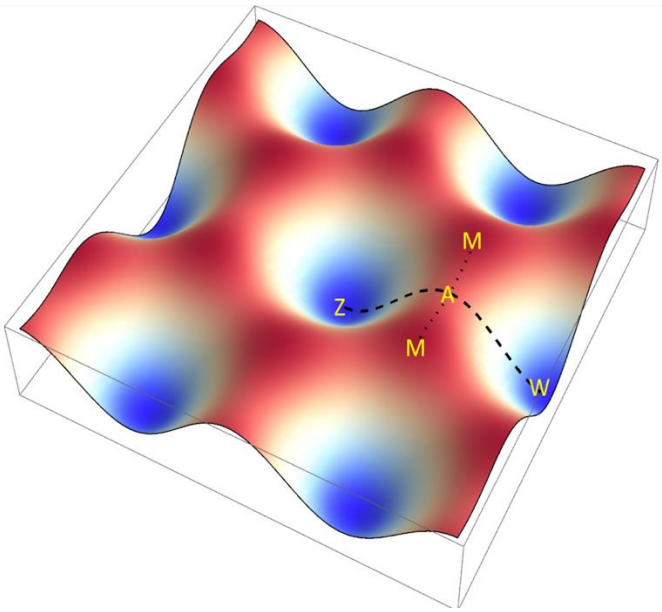

Figure 4.9- The *xy* dependence of the mimimum of the effective potential of He-graphene interaction The dashed and dotted lines shows the paths Z→A→W and A→M respectively (**Figure 4.7** and **Figure 4.8**).

From this *xy* dependent minimum potential energy, our colleagues calculated approximations of the tunneling parameter *t* based on each potential using quasiclassical WKB approximations. First, using a 1D WKB approximation, we can have at least an idea of how the tunneling parameter should look for a ^{4}He atom travelling from site Z to A to W. Based on pre-established quasiclassical hopping models[143,144]:

$$t = \frac{\hbar\omega_0}{2\pi}\exp\left(-\frac{1}{\hbar}\int_{x_{c1}}^{x_{c2}}\sqrt{2mV(x)-\frac{\hbar\omega_0}{2}}\,dx\right) \qquad (4.6)$$

where x_{c2} and x_{c1} are the classical turning points $V(x_c) = \hbar\omega_0/2$ and the integral is over the barrier interior. Here ω_0 is the frequency of small oscillations in the wells. **Table 4.3** shows the oscillation frequency, hopping parameter *t* and ratio to He-He nearest neighbor interaction *t*/*V* (*V* is calculated in the next section) for the MP2 calculations and those of different methods[125]. It is clear that the results from this simple approximation provide reasonable estimates as they provide similar results as those obtained from full two-dimensional methods as done in Ref.[125], see **Table 4.4**.

Table 4.3 – The column for $\hbar\omega_0/2$ is the ground state energy of the wells using a harmonic approximation and the tunneling parameter *t* is found using **Eq. 4.6**[125].

Method	$\hbar\omega_0/2$(K)	*t* (K)	*t*/*V*
Quantum Monte Carlo	15.1	2.22	0.041
Density Functional Theory	19.6	1.25	0.058
Møller–Plesset second-order	27.3	0.43	0.008

Table 4.4 – Tunneling parameter *t*, nearest neighbor interaction *V*, next-nearest neighbor *V*', and the ratio the effective Bose-Hubbard model *t*/*V* calculated numerically using 2D tunneling[125].

Method	*t*(K)	*V* (K)	*V'*	*t*/*V*
Wannier	1.45	7540	638	0.0002
HF	1.45	69.7	-2.08	0.021
QMC	1.38	54.3	-2.76	0.025

DFT	1.10	21.4	-1.36	0.051
MP2	0.59	51.5	-1.97	0.011

To complete the determination of the BHM parameters for He on graphene, we applied the same processes to find potentials of two and three ^{4}He atom systems on center sites, where one ^{4}He atom was moved along the normal of the surface while the rest of the ^{4}He atoms were held at the minima calculated from the single He calculations. The calculations with two ^{4}He atoms are pertinent to the determination of the BHM *V*, *V'*, etc. parameters (see **Eq. 4.1**). For systems with more than one He atom, we first placed one He at the optimal *z* for a particular site (e.g., Z) and brought another He atom from infinity over another site (e.g., W). **Table 4.5** shows the interaction energy for pairs of He atoms over the various local minima in absence of a circumcoronene adsorbent ($U_{\mathrm{HeHe,vacuum}}$) and when both atoms are adsorbed ($U_{\mathrm{HeHe,ads}}$). **Figure 4.10** graphically compares this interaction: the blue dots representing the He-He interaction over circumcoronene, and the continuous red curve the He-He potential in vacuum, both calculated at the MP2 level.

Table 4.5 – Comparison of the in vacuo He-He interaction to that of two He atoms adsorbed over various local minima (see **Figure 4.3**) of the He-circumcoronene potential. The repulsive parameter *V* of the BHM (**Eq. 4.1**) is best described by the potential ($U_{\mathrm{HeHe,ads}}$) of the pair ZW (highlighted in red), and the attractive parameter *V'* is best described by the pair W0W2 (highlighted in blue). Next nearest neighbor *V"* is best described by W3X (highlighted in green).

Pair	d_{HeHe}	$U_{\mathrm{HeHe,vacuum}}$ (K)	$U_{\mathrm{HeHe,ads}}$ (K)
ZW	2.46	44.6	51.52
W0W1	2.46	44.6	51.53
WX	2.46	44.6	48.03
WY	2.46	44.6	48.72
XY	2.46	44.6	50.37
W0W2	4.26	-1.94	-3.02
ZY	4.26	-1.94	-2.14
ZX	4.92	-0.63	-1.97
W3X	7.38	-0.06	-0.80

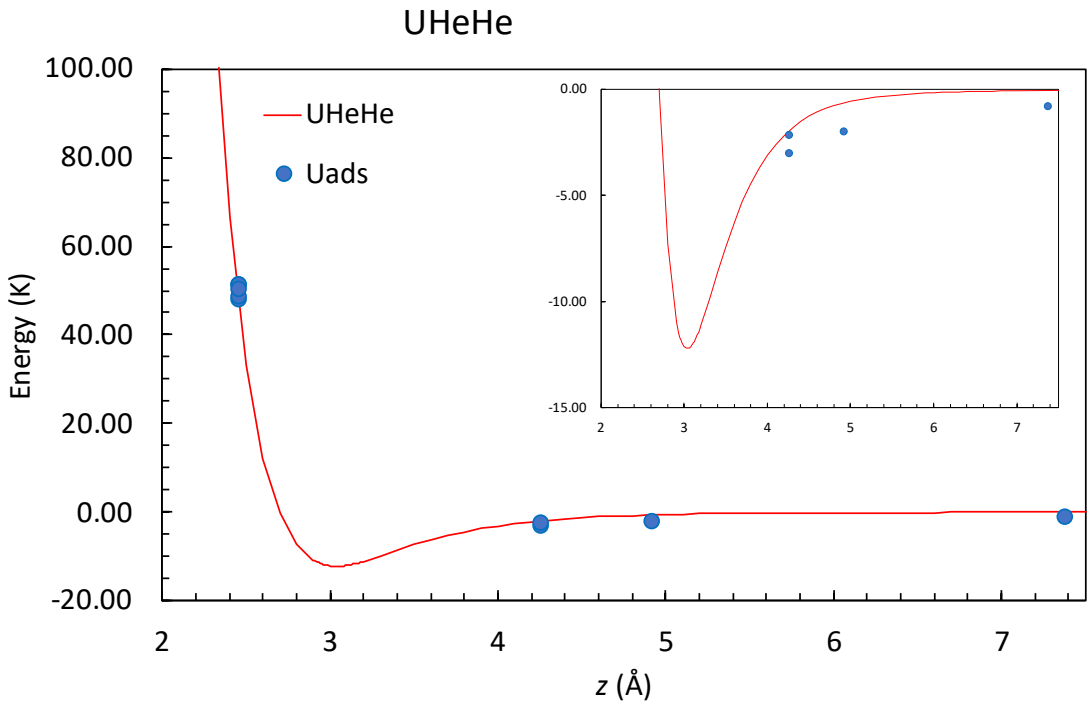

Figure 4.10 - The interaction potential between two ^{4}He atoms in a vacuum as a function of the distance between them (red line) compared to the that of two He atoms on circumcoronene (blue dots). The inset zooms in to the attractive part of the potential.

The repulsive parameter V of the BHM (**Eq. 4.1**) is best described by the potential ($U_{\mathrm{HeHe,ads}}$) of the pair ZW, and the attractive parameter V' is best described by the pair W0W2. Next nearest neighbor V'' would be best described by W0W3. The most remarkable feature of this calculation is the substantial enhancement of the V' and V'' attractive interactions for He adsorbed on the surface. Although these next and further nearest neighbor terms are ignored in the standard treatment of the BHM (**Eq. 4.1**), they may become important under conditions when the nearest neighbor repulsive interaction vanishes, i.e., if the adsorbent material results in a distance between nearest neighbors close to 2.7 Å. We leave the exploration of these cases for future work.

Appendix A: Parameters and Equations

Tables A1-A4. Bonded potential parameters. See **Eqs. 2.1-2.4** in the main body of the manuscript. These parameters were used in the MD studies of **Chapter 2** and **Chapter 3**.

Table A1 - Bonded Potential Parameters Used in the Pore-Expansion Simulations

Angles	**θ_0**[°]	**K_θ** [kcal/(mol rad^2)]	**S_0**[Å]	**K_{UB}** [kcal/(mol Å^2)]
CG2R61 - CG2R61 - CG2R61	120.000	40.000	2.41967	35.00
CA - CA - CA	120.000	40.000	2.41967	35.00
HGR61 - CG2R61 - CG2R61	120.000	30.000	2.15265	22.00
HGR61 - CA - CA	120.000	30.000	2.15265	22.00
CG321 - CG321 - CG321	113.4205	58.350	2.54834	11.16
CG331 - CG321 - CG321	112.904	34.600	2.15265	8.00
CG331 - CG321 - CG331	114.000	53.350	2.561	8.00
HB - CG321 - CG321	109.350	26.500	2.14993	22.53
HB - CG321 - CG331	109.612	34.600	2.15265	22.53
HB - CG331 - CG321	111.025	34.600	2.16745	22.53
HB - CG341 - HB	109.500	35.500		
HB - CG331 - HB	107.770	35.500	1.74399	1.74399
HB - CG321 - HB	106.400	35.500	1.76593	1.76593
CG3C31 - CG3C31 - OG3C31	56.50	30.00		
CG3C31 - OG3C31 - CG3C31	67.00	50.35		
CA - CG3C31 - OG3C31	115.10	30.00		
CSP23 - CG3C31 - OG3C31	115.10	30.00		
CA - CG3C31 - CG3C31	117.20	30.00		
CSP23 - CG3C31 - CG3C31	117.20	30.00		
CG3C31 - CG3C31 - CG3C31	117.20	30.00		
CSP23 - CG3C31 - CSP23	118.40	30.00		
CA - CA - CG3C31	119.70	30.00		
CG3C31 - CA - CG3C31	119.70	30.00		
CSP23 - CA - CG3C31	119.70	30.00		
CA - CSP23 - CG3C31	119.70	30.00		
CSP23 - CSP23 - CG3C31	119.70	30.00		
CG3C31 - CSP23 - CG3C31	119.70	30.00		
CA - CG3C31 - CA	119.70	30.00		
CA - CG3C31 - CSP23	119.70	30.00		
CSP23 - CSP23 - CA	115.700	40.000		
CSP23 - CA - CA	118.975	40.000		
CA - CSP23 - CA	111.900	40.000		
CA - CSP23 - OXGN	104.000	45.000		
CSP23 - CSP23 - OXGN	103.700	45.000		
CSP23 - OXGN - BCOO	109.600	85.000		
OXGN - BCOO - OXGN	120.000	76.0300		
OXGN - BCOO - CG2R61	120.000	57.300		
BCOO - CG2R61 - CG2R61	120.000	48.00		
BCOO - OXGN - HCP1	114.700	50.00		
CSP23 - OXGN - HCP1	109.000	60.0		
CSP23 - CSP23 - CSP23	110.000	40.000		
CSP23 - CA - CSP23	110.000	40.000		
CG3C31 - CSP23 - OXGN	103.700	45.000		

Bonds	**b_0[Å]**	**K_b [kcal/(mol Å^2)]**
CG2R61 - CG2R61	1.402	305.000
CA - CA	1.4220	305.000
CA - CSP23	1.520	305.000
CSP23 - CSP23	1.586	305.000
CG3C31 - CG3C31	1.501	240.00
CG3C31 - CA	1.468	240.00
CSP23 - CG3C31	1.53	222.500
CG2R61 - HGR61	1.087	340.000

HB - CG341	1.089	322.000
CSP23 - OXGN	1.473	320.000
OXGN - BCOO	1.370	424.000
CG2R61 - BCOO	1.556	314.000
CG3C31 - OG3C31	1.4310	220.00
OXGN - HCP1	0.986	560.00

Dihedrals	***N***	**δ [°]**	**K_χ [kcal/mol]**
CG2R61 - CG2R61 - CG2R61 - CG2R61	2	180.00	3.1000
HGR61 - CG2R61 - CG2R61 - CG2R61	2	180.00	4.2000
HGR61 - CA - CA - CA	2	180.00	4.2000
HGR61 - CG2R61 - CG2R61 - HGR61	2	180.00	2.4000
HGR61 - CA - CA - HGR61	2	180.00	2.4000
CA - CG3C31 - CG3C31 - OG3C31	4	0.00	2.000
CG3C31 - CG3C31 - CG3C31 - OG3C31	4	0.00	2.000
CSP23 - CG3C31 - CG3C31 - OG3C31	4	0.00	2.000
CA - CSP23 - CG3C31 - OG3C31	4	0.00	2.000
CG3C31 - CSP23 - CG3C31 - OG3C31	4	0.00	2.000
CSP23 - CSP23 - CG3C31 - OG3C31	4	0.00	2.000
CA - CA - CG3C31 - OG3C31	4	0.00	2.000
CG3C31 - CA - CG3C31 - OG3C31	4	0.00	2.000
CSP23 - CA - CG3C31 - OG3C31	4	0.00	2.000
OG3C31 - CG3C31 - CG3C31 - OG3C31	2	180.00	2.000
OXGN - CSP23 - CG3C31 - OG3C31	2	180.00	2.000
CA - CG3C31 - OG3C31 - CG3C31	2	0.00	2.000
CG3C31 - CG3C31 - OG3C31 - CG3C31	2	0.00	2.000
CSP23 - CG3C31 - OG3C31 - CG3C31	2	0.00	2.000
CA - CA - CG3C31 - CG3C31	2	180.00	2.000
CG3C31 - CA - CG3C31 - CG3C31	2	180.00	2.000
CSP23 - CA - CG3C31 - CG3C31	2	180.00	2.000
CA - CSP23 - CG3C31 - CG3C31	2	180.00	2.000
CG3C31 - CSP23 - CG3C31 - CG3C31	2	180.00	2.000
CSP23 - CSP23 - CG3C31 - CG3C31	2	180.00	2.000
CA - CG3C31 - CG3C31 - CG3C31	2	180.00	2.000
CG3C31 - CG3C31 - CG3C31 - CG3C31	2	180.00	2.000
CSP23 - CG3C31 - CG3C31 - CG3C31	2	180.00	2.000
CA - CA - CSP23 - CG3C31	2	180.00	2.000
CG3C31 - CA - CSP23 - CG3C31	2	180.00	2.000
CSP23 - CA - CSP23 - CG3C31	2	180.00	2.000
CA - CSP23 - CSP23 - CG3C31	2	180.00	2.000
CG3C31 - CSP23 - CSP23 - CG3C31	2	180.00	2.000
CSP23 - CSP23 - CSP23 - CG3C31	2	180.00	2.000
CA - CG3C31 - CSP23 - CG3C31	2	180.00	2.000
CG3C31 - CG3C31 - CSP23 - CG3C31	2	180.00	2.000
CSP23 - CG3C31 - CSP23 - CG3C31	2	180.00	2.000
CA - CA - CA - CG3C31	2	180.00	2.000
CG3C31 - CA - CA - CG3C31	2	180.00	2.000
CSP23 - CA - CA - CG3C31	2	180.00	2.000
CA - CSP23 - CA - CG3C31	2	180.00	2.000
CG3C31 - CSP23 - CA - CG3C31	2	180.00	2.000
CSP23 - CSP23 - CA - CG3C31	2	180.00	2.000
CA - CG3C31 - CA - CG3C31	2	180.00	2.000
CG3C31 - CG3C31 - CA - CG3C31	2	180.00	2.000
CSP23 - CG3C31 - CA - CG3C31	2	180.00	2.000
CG2R61 - CG2R61 - CG2R61 - OXGN	2	180.00	3.1000
CA - CA - CA - OXGN	2	180.00	3.1000
CA - CA - CSP23 OXGN	2	0.00	2.0000
CSP2 - CA - CSP23 - OXGN	2	0.00	2.0000
CG3C31 - CA - CSP23 - OXGN	2	0.00	2.0000
CA - CSP23 - CSP23 - OXGN	2	0.00	2.0000
CSP23 - CSP23 - CSP23 - OXGN	2	0.00	2.0000
CG3C31 - CSP23 - CSP23 - OXGN	2	0.00	2.0000
CA - CG3C31 - CSP23 - OXGN	2	0.00	2.0000

CSP23 - CG3C31 - CSP23 - OXGN	2	0.00	2.0000
CG3C31 - CG3C31 - CSP23 - OXGN	2	0.00	2.0000
CA - CSP23 - OXGN - HCP1	3	0.00	0.207
CSP23 - CSP23 - OXGN - HCP1	3	0.00	0.207
CG3C31 - CSP23 - OXGN - HCP1	3	0.00	0.207
OXGN - CSP23 - CSP23 - OXGN	2	180.00	3.1000
CA - CA - CA - CA	2	180.00	3.1000
CA - CA - CA - CSP23	2	180.00	2.0000
CA - CA - CSP23 - CA	2	180.00	2.0000
CA - CA - CSP23 - CSP23	2	180.00	2.0000
CA - CSP23 - CSP23 - CA	2	180.00	2.0000
CA - CSP23 - CSP23 - CSP23	2	180.00	2.0000
CA - CSP23 - CA - CSP23	2	180.00	2.0000
CSP23 - CA - CA - CSP23	2	180.00	2.0000
CSP23 - CSP23 - CSP23 - CSP23	2	180.00	2.0000
CSP23 - CSP23 - CA - CSP23	2	180.00	2.0000
CA - CA - CA - CG3C31	2	180.00	2.0000
CA - CA - CG3C31 - CA	2	180.00	2.0000
CA - CA - CG3C31 - CG3C31	2	180.00	2.0000
CA - CG3C31 - CG3C31 - CA	2	180.00	2.0000
CA - CG3C31 - CG3C31 - CA	2	180.00	2.0000
CA - CG3C31 - CG3C31 - CG3C31	2	180.00	2.0000
CA - CG3C31 - CA - CG3C31	2	180.00	2.0000
CG3C31 - CA - CA - CG3C31	2	180.00	2.0000
CG3C31 - CG3C31 - CG3C31 - CG3C31	2	180.00	2.0000
CSP23 - CSP23 - CSP23 - CG3C31	2	180.00	2.0000
CSP23 - CSP23 - CG3C31 - CSP23	2	180.00	2.0000
CSP23 - CSP23 - CG3C31 - CG3C31	2	180.00	2.0000
CSP23 - CG3C31 - CG3C31 - CSP23	2	180.00	2.0000
CSP23 - CG3C31 - CG3C31 - CG3C31	2	180.00	2.0000
CSP23 - CG3C31 - CG3C31 - CG3C31	2	180.00	2.0000
CSP23 - CG3C31 - CSP23 - CG3C31	2	180.00	2.0000
CG3C31 - CSP23 - CSP23 - CG3C31	2	180.00	2.0000
CA - CA - CSP23 - CG3C31	2	180.00	2.0000
CA - CA - CG3C31 - CSP23	2	180.00	2.0000
CA - CSP23 - CG3C31 - CA	2	180.00	2.0000
CA - CSP23 - CA - CG3C31	2	180.00	2.0000
CA - CG3C31 - CA - CSP23	2	180.00	2.0000
CSP23 - CA - CA - CG3C31	2	180.00	2.0000
CA - CSP23 - CSP23 - CG3C31	2	180.00	2.0000
CA - CSP23 - CG3C31 - CSP23	2	180.00	2.0000
CA - CG3C31 - CSP23 - CSP23	2	180.00	2.0000
CA - CG3C31 - CG3C31 - CSP23	2	180.00	2.0000
CA - CG3C31 - CSP23 - CG3C31	2	180.00	2.0000
CA - CSP23 - CG3C31 - CG3C31	2	180.00	2.0000
CSP23 - CA - CSP23 - CG3C31	2	180.00	2.0000
CSP23 - CA - CG3C31 - CSP23	2	180.00	2.0000
CG3C31 - CA - CSP23 - CSP23	2	180.00	2.0000
CG3C31 - CA - CG3C31 - CSP23	2	180.00	2.0000
CG3C31 - CA - CSP23 - CG3C31	2	180.00	2.0000
CSP23 - CA - CG3C31 - CG3C31	2	180.00	2.0000
CA - CSP23 - OXGN - BCOO	3	180.00	3.1000
CSP23 - CSP23 - OXGN - BCOO	3	180.00	3.1000
CG3C31 - CSP23 - OXGN - BCOO	3	180.00	3.1000
CSP23 - OXGN - BCOO - OXGN	1	180.00	3.1000
CSP23 - OXGN - BCOO - CG2R61	1	0.00	2.590
OXGN - BCOO - CG2R61 - CG2R61	2	180.00	2.590
CSP23 - CA - CA - HGR61	2	180.00	4.2000
BCOO - CG2R61 - CG2R61 - CG2R61	1	0.00	2.590
BCOO - CG2R61 - CG2R61 - HGR61	1	180.00	2.590
HCP1 - OXGN - BCOO - OXGN	1	180.00	0.20
HCP1 - OXGN - BCOO - CG2R61	2	180.00	2.34

Table A2 – Bonding parameters for modeling CH_4, CO_2, and Propane adsorption on GOF

bonds	b_0[Å]	k_b[kcal/(mol Å^2)]
CA—CA	1.42	305.0
CA—CSP23	1.52	305.0
CSP23—CSP23	1.586	305.0
CG3C31—CA	1.468	240.00
CG3C31—CSP23	1.520	250.000
CG3C31—CG3C31	1.501	240.00
CG3C31—OG3C31	1.4310	220.00
OXGN--HCP1	0.968	560.00
OXGN—CSP23	1.473	320.000
HB—CG341	1.089	322.000
CG331—CG321	1.5280	222.500
HB—CG331	1.089	322.000
CG321—HB	1.093	309.000
CG2O7--OG2D5	1.1600	986.00

Table A3 – Angle parameters for modeling CH_4, CO_2, and Propane adsorption on GOF

angles	θ_0[°]	k_θ[kcal/(mol rad^2)]	S_0[Å]	K_{UB}[kcal/(mol Å^2)]
CA-CA-CA	120.000	40.000	2.41967	35.00
CSP23-CA-CA	118.975	40.000		
CA-CSP23-CA	111.900	40.000		
CSP23-CSP23-CA	115.700	40.000		
CSP23-CA-CSP23	110.000	40.000		
CSP23-CSP23-CSP23	110.000	40.000		
CA-CA-CG3C31	119.70	30.00		
CA-CG3C31-CA	119.70	30.00		
CG3C31-CA-CG3C31	119.70	30.00		
CG3C31-CG3C31-CA	117.20	30.00		
CG3C31-CG3C31-CG3C31	117.20	30.00		
CSP23-CSP23-CG3C31	119.70	30.00		
CSP23-CG3C31-CSP23	118.40	30.00		
CG3C31-CSP23-CG3C31	119.70	30.00		
CSP23-CG3C31-CG3C31	117.20	30.00		
CSP23-CA-CG3C31	119.70	30.00		
CA-CSP23-CG3C31	119.70	30.00		
CA-CG3C31-CSP23	119.70	30.00		
CA-CSP23-OXGN	104.000	45.000		
CSP23-CSP23-OXGN	103.700	45.000		
CSP23-OXGN-HCP1	109.000	60.0		
CG3C31-CSP23-OXGN	103.700	45.000		
HB-CG341-HB	35.500	109.500		
OG2D5-CG2O7-OG2D5	45.00	180.00		
HB-CG331-CG321	111.025	34.600	2.16745	22.53
HB-CG321-CG331	109.612	34.600	2.15265	22.53
CG331-CG321-CG331	114.000	53.350	2.561	8.00
HB-CG331-HB	107.770	35.500	1.74399	1.74399
HB-CG321-HB	106.400	35.500	1.76593	1.76593

Table A4 – Dihedral parameters for modeling CH_4, CO_2, and Propane adsorption on GOF

dihedrals	n	δ [°]	K_χ [kcal/mol]
CA - CA - CA - CA	2	180.00	3.1000
CA - CA - CA - CSP23	2	180.00	2.0000
CA - CA - CSP23 - CA	2	180.00	2.0000
CA - CA - CSP23 - CSP23	2	180.00	2.0000
CA - CSP23 - CSP23 - CA	2	180.00	2.0000
CA - CSP23 - CSP23 - CSP23	2	180.00	2.0000
CA - CSP23 - CA - CSP23	2	180.00	2.0000
CSP23 - CA - CA - CSP23	2	180.00	2.0000

CSP23 - CSP23 - CSP23 - CSP23	2	180.00	2.0000
CSP23 - CSP23 - CA - CSP23	2	180.00	2.0000
CA - CA - CA - CG3C31	2	180.00	2.0000
CA - CA - CG3C31 - CA	2	180.00	2.0000
CA - CA - CG3C31 - CG3C31	2	180.00	2.0000
CA - CG3C31 - CG3C31 - CA	2	180.00	2.0000
CA - CG3C31 - CG3C31 - CA	2	180.00	2.0000
CA - CG3C31 - CG3C31 - CG3C31	2	180.00	2.0000
CA - CG3C31 - CA - CG3C31	2	180.00	2.0000
CG3C31 - CA - CA - CG3C31	2	180.00	2.0000
CG3C31 - CG3C31 - CG3C31 - CG3C31	2	180.00	2.0000
CSP23 - CSP23 - CSP23 - CG3C31	2	180.00	2.0000
CSP23 - CSP23 - CG3C31 - CSP23	2	180.00	2.0000
CSP23 - CSP23 - CG3C31 - CG3C31	2	180.00	2.0000
CSP23 - CG3C31 - CG3C31 - CSP23	2	180.00	2.0000
CSP23 - CG3C31 - CG3C31 - CG3C31	2	180.00	2.0000
CSP23 - CG3C31 - CG3C31 - CG3C31	2	180.00	2.0000
CSP23 - CG3C31 - CSP23 - CG3C31	2	180.00	2.0000
CG3C31 - CSP23 - CSP23 - CG3C31	2	180.00	2.0000
CA - CA - CSP23 - CG3C31	2	180.00	2.0000
CA - CA - CG3C31 - CSP23	2	180.00	2.0000
CA - CSP23 - CG3C31 - CA	2	180.00	2.0000
CA - CSP23 - CA - CG3C31	2	180.00	2.0000
CA - CG3C31 - CA - CSP23	2	180.00	2.0000
CSP23 - CA - CA - CG3C31	2	180.00	2.0000
CA - CSP23 - CSP23 - CG3C31	2	180.00	2.0000
CA - CSP23 - CG3C31 - CSP23	2	180.00	2.0000
CA - CG3C31 - CSP23 - CSP23	2	180.00	2.0000
CA - CG3C31 - CG3C31 - CSP23	2	180.00	2.0000
CA - CG3C31 - CSP23 - CG3C31	2	180.00	2.0000
CA - CSP23 - CG3C31 - CG3C31	2	180.00	2.0000
CSP23 - CA - CSP23 - CG3C31	2	180.00	2.0000
CSP23 - CA - CG3C31 - CSP23	2	180.00	2.0000
CG3C31 - CA - CSP23 - CSP23	2	180.00	2.0000
CG3C31 - CA - CG3C31 - CSP23	2	180.00	2.0000
CG3C31 - CA - CSP23 - CG3C31	2	180.00	2.0000
CSP23 - CA - CG3C31 - CG3C31	2	180.00	2.0000
CA - CG3C31 - CG3C31 - OG3C31	4	0.00	2.000
CG3C31 - CG3C31 - CG3C31 - OG3C31	4	0.00	2.000
CSP23 - CG3C31 - CG3C31 - OG3C31	4	0.00	2.000
CA - CSP23 - CG3C31 - OG3C31	4	0.00	2.000
CG3C31 - CSP23 - CG3C31 - OG3C31	4	0.00	2.000
CSP23 - CSP23 - CG3C31 - OG3C31	4	0.00	2.000
CA - CA - CG3C31 - OG3C31	4	0.00	2.000
CG3C31 - CA - CG3C31 - OG3C31	4	0.00	2.000
CSP23 - CA - CG3C31 - OG3C31	4	0.00	2.000
OG3C31 - CG3C31 - CG3C31 - OG3C31	2	180.00	2.000
OXGN - CSP23 - CG3C31 - OG3C31	2	180.00	2.000
CA - CG3C31 - OG3C31 - CG3C31	2	0.00	2.000
CG3C31 - CG3C31 - OG3C31 - CG3C31	2	0.00	2.000
CSP23 - CG3C31 - OG3C31 - CG3C31	2	0.00	2.000
CA - CA - CG3C31 - CG3C31	2	180.00	2.000
CG3C31 - CA - CG3C31 - CG3C31	2	180.00	2.000
CSP23 - CA - CG3C31 - CG3C31	2	180.00	2.000
CA - CSP23 - CG3C31 - CG3C31	2	180.00	2.000
CG3C31 - CSP23 - CG3C31 - CG3C31	2	180.00	2.000
CSP23 - CSP23 - CG3C31 - CG3C31	2	180.00	2.000
CA - CG3C31 - CG3C31 - CG3C31	2	180.00	2.000
CG3C31 - CG3C31 - CG3C31 - CG3C31	2	180.00	2.000
CSP23 - CG3C31 - CG3C31 - CG3C31	2	180.00	2.000
CA - CA - CSP23 - CG3C31	2	180.00	2.000
CG3C31 - CA - CSP23 - CG3C31	2	180.00	2.000

CSP23 - CA - CSP23 - CG3C31	2	180.00	2.000
CA - CSP23 - CSP23 - CG3C31	2	180.00	2.000
CG3C31 - CSP23 - CSP23 - CG3C31	2	180.00	2.000
CSP23 - CSP23 - CSP23 - CG3C31	2	180.00	2.000
CA - CG3C31 - CSP23 - CG3C31	2	180.00	2.000
CG3C31 - CG3C31 - CSP23 - CG3C31	2	180.00	2.000
CSP23 - CG3C31 - CSP23 - CG3C31	2	180.00	2.000
CA - CA - CA - CG3C31	2	180.00	2.000
CG3C31 - CA - CA - CG3C31	2	180.00	2.000
CSP23 - CA - CA - CG3C31	2	180.00	2.000
CA - CSP23 - CA - CG3C31	2	180.00	2.000
CG3C31 - CSP23 - CA - CG3C31	2	180.00	2.000
CSP23 - CSP23 - CA - CG3C31	2	180.00	2.000
CA - CG3C31 - CA - CG3C31	2	180.00	2.000
CG3C31 - CG3C31 - CA - CG3C31	2	180.00	2.000
CSP23 - CG3C31 - CA - CG3C31	2	180.00	2.000
CA - CA - CA - OXGN	2	180.00	3.1000
CA - CA - CSP23 - OXGN	2	0.00	2.0000
CSP2 - CA - CSP23 - OXGN	2	0.00	2.0000
CG3C31 - CA - CSP23 - OXGN	2	0.00	2.0000
CA - CSP23 - CSP23 - OXGN	2	0.00	2.0000
CSP23 - CSP23 - CSP23 - OXGN	2	0.00	2.0000
CG3C31 - CSP23 - CSP23 - OXGN	2	0.00	2.0000
CA - CG3C31 - CSP23 - OXGN	2	0.00	2.0000
CSP23 - CG3C31 - CSP23 - OXGN	2	0.00	2.0000
CG3C31 - CG3C31 - CSP23 - OXGN	2	0.00	2.0000
CA - CSP23 - OXGN - HCP1	3	0.00	0.207
CSP23 - CSP23 - OXGN - HCP1	3	0.00	0.207
CG3C31 - CSP23 - OXGN - HCP1	3	0.00	0.207
OXGN - CSP23 - CSP23 - OXGN	2	180.00	3.1000

Non-bonded potential parameters. See **Eq. 2.5** and **2.6**. These parameters were used in the MD studies of **Chapter 2** and **Chapter 3** as well as the GCMC study of **Chapter 3**.

Table A5 –Nonbonded parameters for modeling Xenon CH_4, CO_2, and Propane adsorption on GOF

	q [esu]	ε [kcal/mol]	r_{min} [Å]	Notes
CG341	-0.360	-0.078	2.050	Methane C
HB	+0.090	-0.022	1.320	Methane H, Propane H
XE	0.000	-0.4990	2.284	Xenon
CA	0.000	-0.070	1.992	Graphene/GO/GOF C
CSP23	+0.238	-0.070	1.992	GOF C bonded to DBA O
	+0.215			GO/GOF C bonding to hydroxyl O
OXGN	-0.418	-0.100	1.650	DBA O bonded to C
	-0.561			O in fluid DBA, bonded to H
	-0.561			O in VdW DBA, free end, bonded to H
	-0.443			O in VdW DBA, bonded end, bonded to C
	-0.601			OH Group
BCOO	+0.453	-0.200	2.290	DBA Boron, bonded side
	+0.365			Fluid DBA
	+0.367			DBA Boron, non-bonded side
CG2RG1	-0.189	-0.070	1.992	Covalent DBA Benzene Ring bonded to H
	+0.062			DBA Benzene Ring bonded to Boron,

				bonded side
	-0.190			VdW DBA Benzene Ring bonded to H
	+0.367			DBA Benzene Ring bonded to B, unbonded side
	-0.184			Fluid DBA Benzene Ring bonded to H
	+0.071			Fluid DBA Benzene Ring bonded to B
HGR1	+0.140	-0.030	1.358	H bonded to Covalent and VdW benzene ring
	+0.137			H bonded to Fluid benzene ring
HCP1	+0.390	-0.046	0.2245	H bonded to end of DBA
	+0.386			H in OH Group
CG3C31	+0.221	-0.056	2.010	GO C bonded to Epoxy
OG3C31	-0.4422	-0.100	1.650	Epoxy O
CG331	-0.0780	-0.27	2.0500	Propane edge carbon
CG321	-0.0560	-0.18	2.0100	Propane center carbon
CG2O7	-0.0580	0.60	1.5630	Carbon-Dioxide carbon
OG2D5	-0.1650	-0.30	1.6920	Carbon-Dioxide carbon

From the study in **Chapter 2**, the CH_4 density to pressure formula used is:

$$P(\rho) = 0.318\rho^3 - 3.13\rho^2 + 41.3\rho \qquad (C.1)$$

Where P is in bar and ρ is molecule density in molecule/nm^3.

From the study in **Chapter 2**, the xenon density to pressure formula used is:

$$P(\rho) = 0.664\rho^3 - 8.93\rho^2 + 41.4\rho \qquad (C.2)$$

Where P is in bar and ρ is molecule density in molecule/nm^3.

Appendix B: Supplementary Information and Data Sets

Below are contour plots of the co-adsorption of CH_4 and CO_2. This data was collected from GCMC simulations run in RASPA using the methods described in Chapter 3.

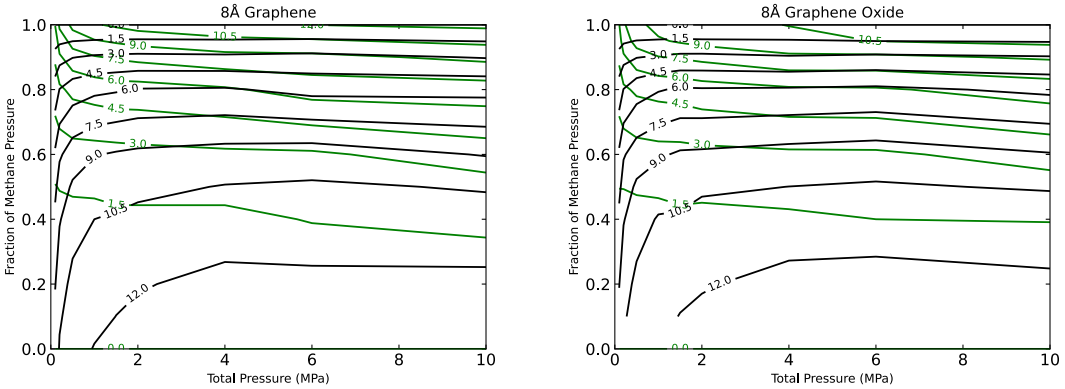

Figure B1 – Coadsorption of CH_4 (green) and CO_2 (black) for graphene and GO for 8Å pores. The contour lines labels are in mol/kg.

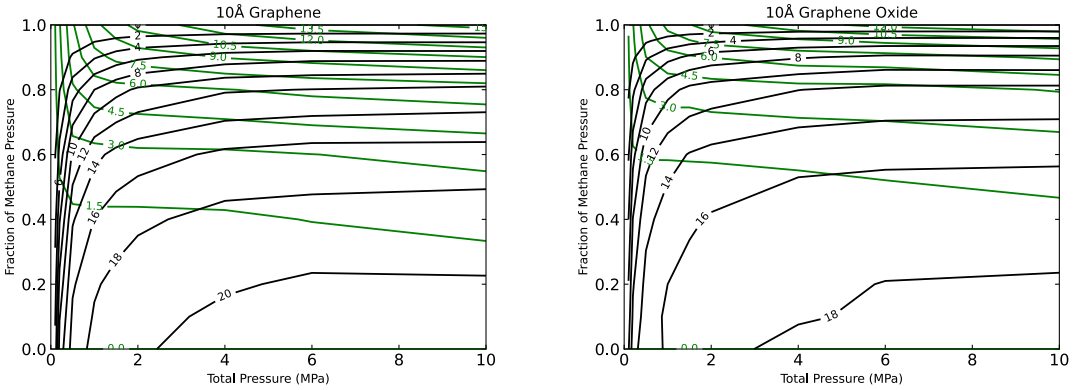

Figure B2 – 10Å pores.

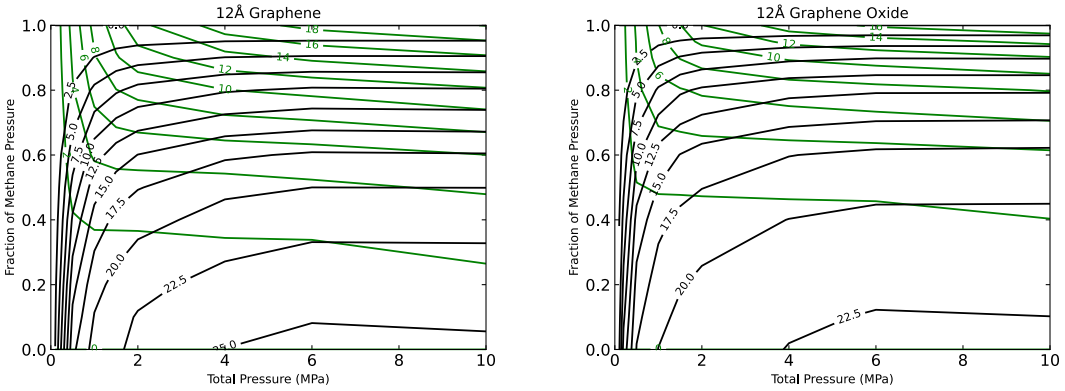

Figure B3 –12Å pores.

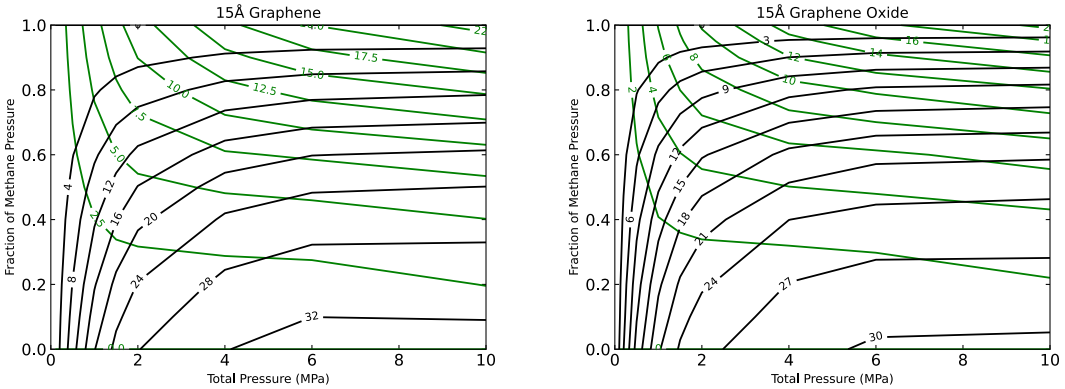

Figure B4 –15Å pores.

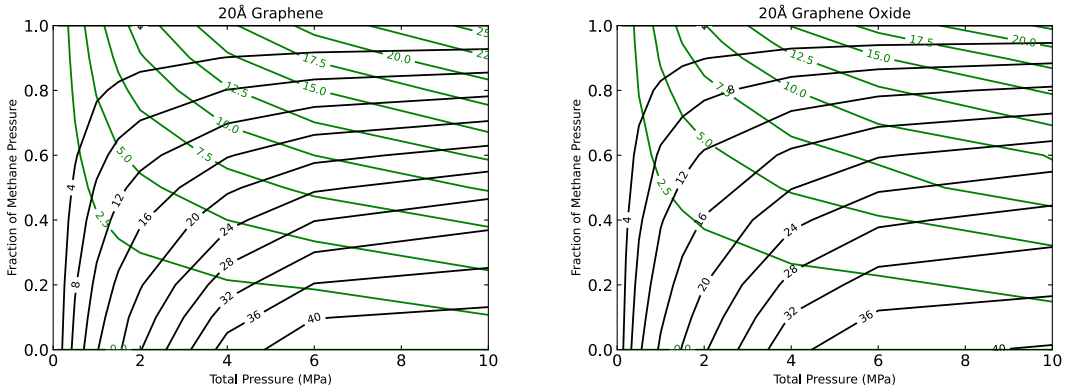

Figure B5 –20Å pores.

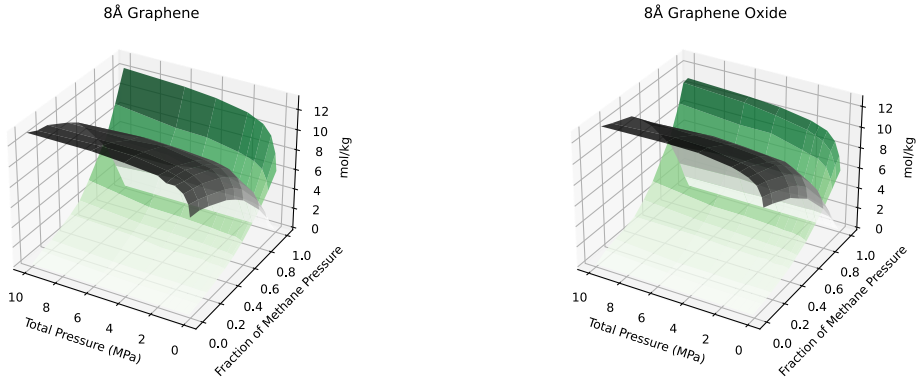

Figure B1.2 – Surface plots using the same data as **Fig B1** above.

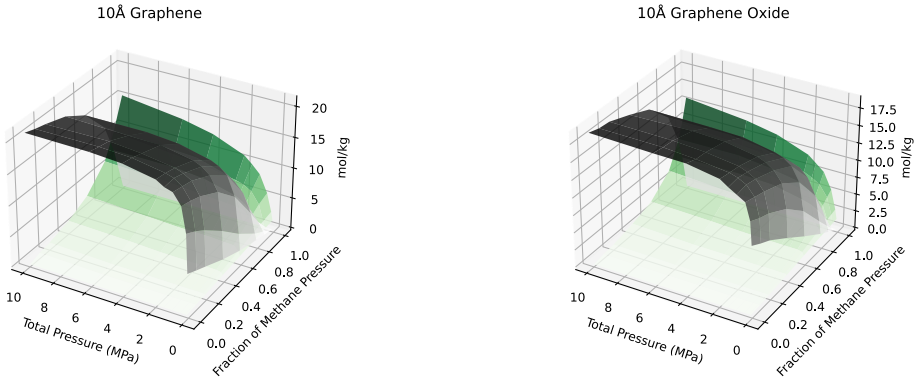

Figure B2.2 – Surface plots using the same data as **Fig B2** above.

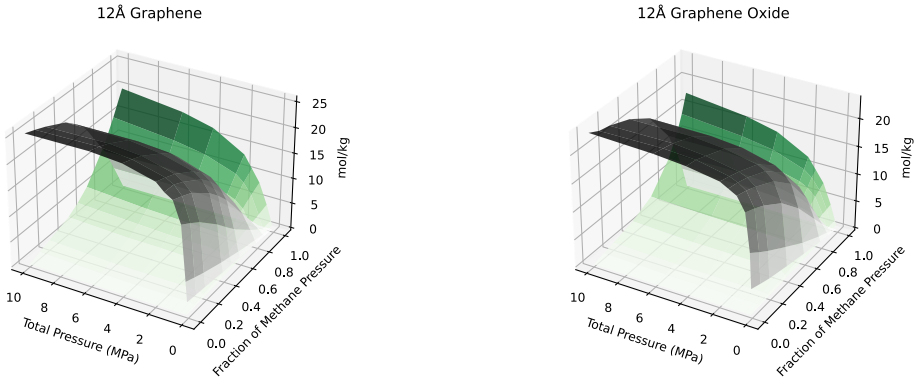

Figure B3.2 – Surface plots using the same data as **Fig B3** above.

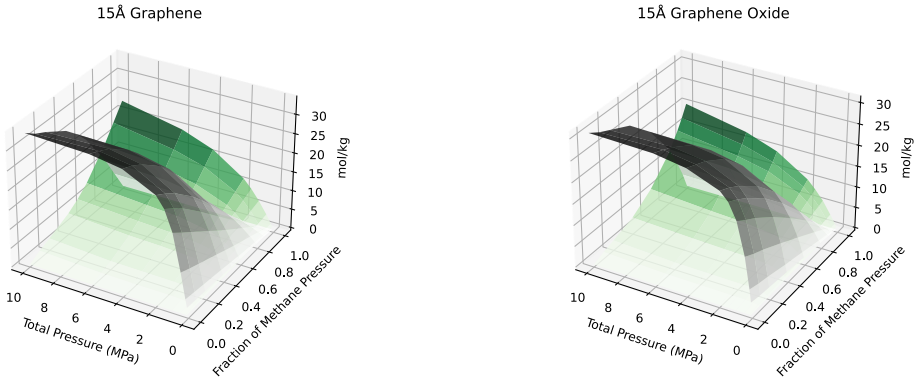

Figure B4.2 -– Surface plots using the same data as **Fig B4** above.

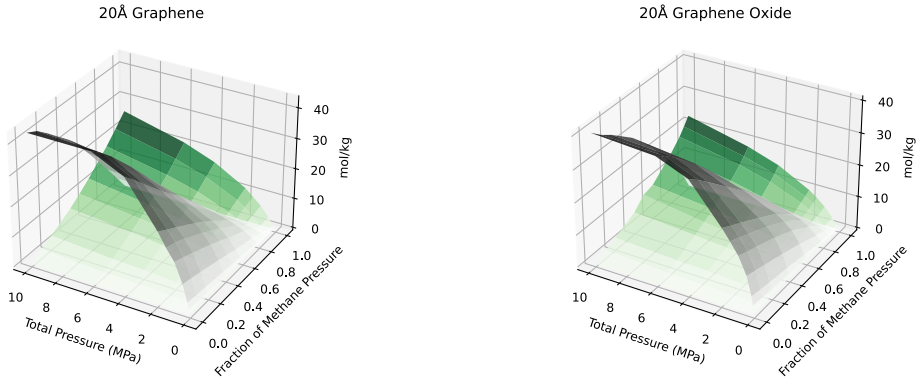

Figure B5.2 – Surface plots using the same data as **Fig B5** above.

Below are adsorption isotherms from GCMC studies of the coadsorption of CH_4 and CO_2. The first five sets of plots compare how oxidization affects adsorption (**Fig B6-10**). The second sets of plots contain all of the adsorption isotherms for every pore size of graphene and GO (**Fig B11-15**).

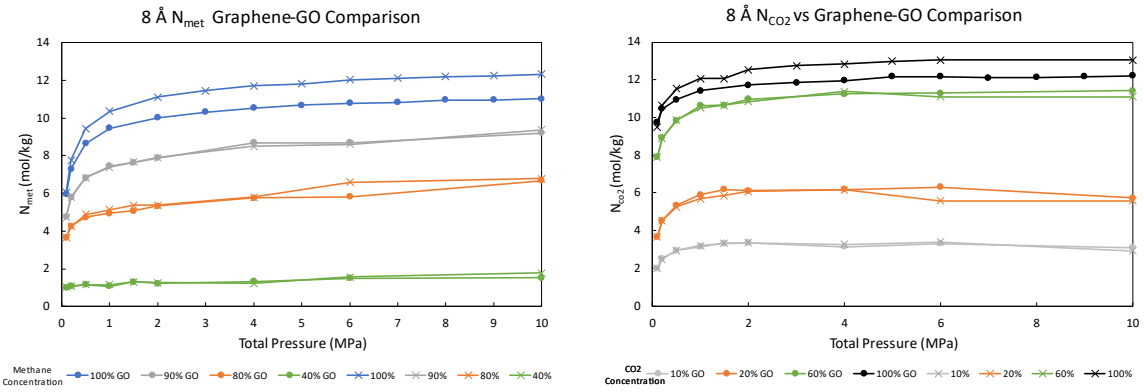

Figure B6 – Adsorption at select partial pressures for CH_4 and CO_2 in graphene and GO in 8Å pores comparing GO and graphene in the same plots.

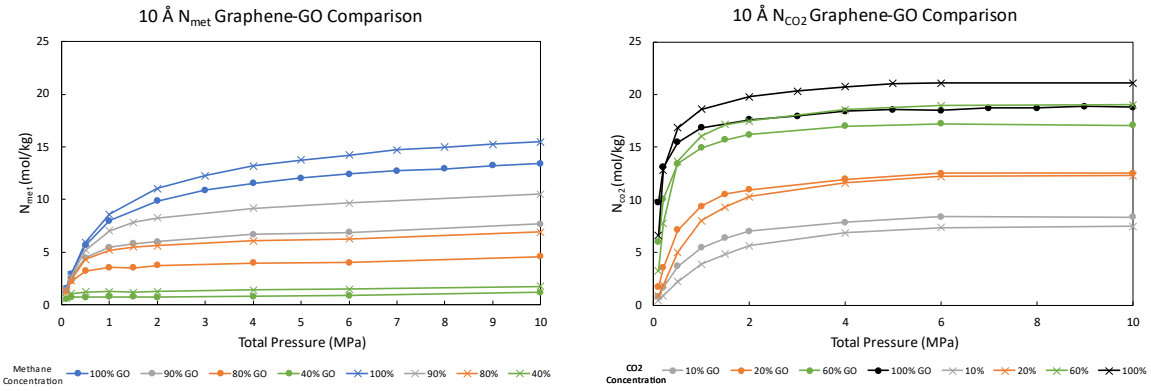

Figure B7 – Adsorption at select partial pressures for CH_4 and CO_2 in graphene and GO in 10Å pores comparing GO and graphene in the same plots.

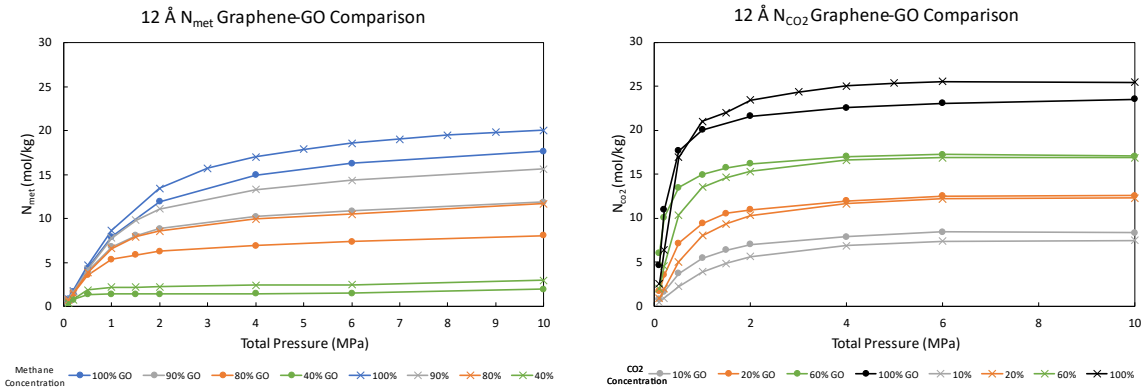

Figure B8 - Adsorption at select partial pressures for CH_4 and CO_2 in graphene and GO in 12Å pores comparing GO and graphene in the same plots.

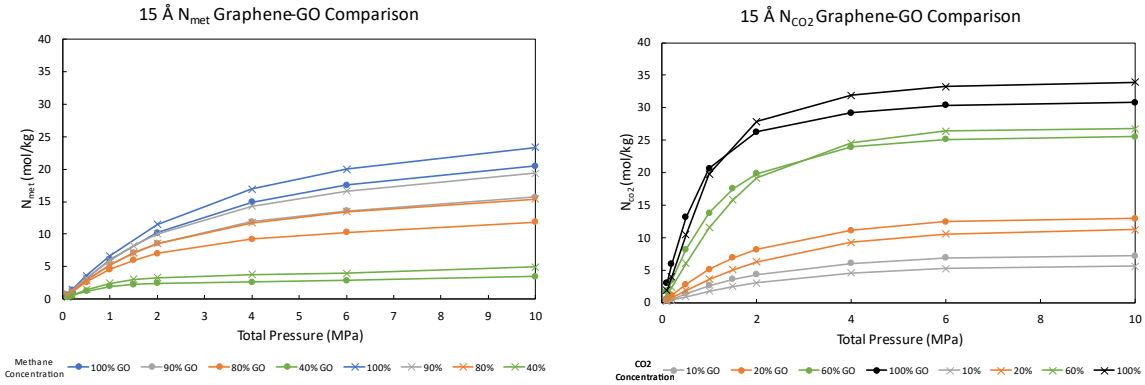

Figure B9 - Adsorption at select partial pressures for CH_4 and CO_2 in graphene and GO in 15Å pores comparing GO and graphene in the same plots.

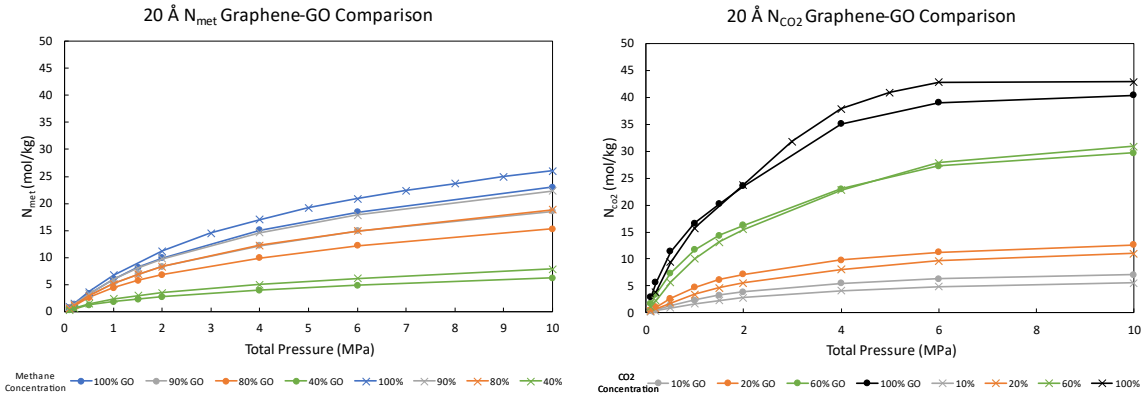

Figure B10 - Adsorption at select partial pressures for CH_4 and CO_2 in graphene and GO in 20Å pores comparing GO and graphene in the same plots.

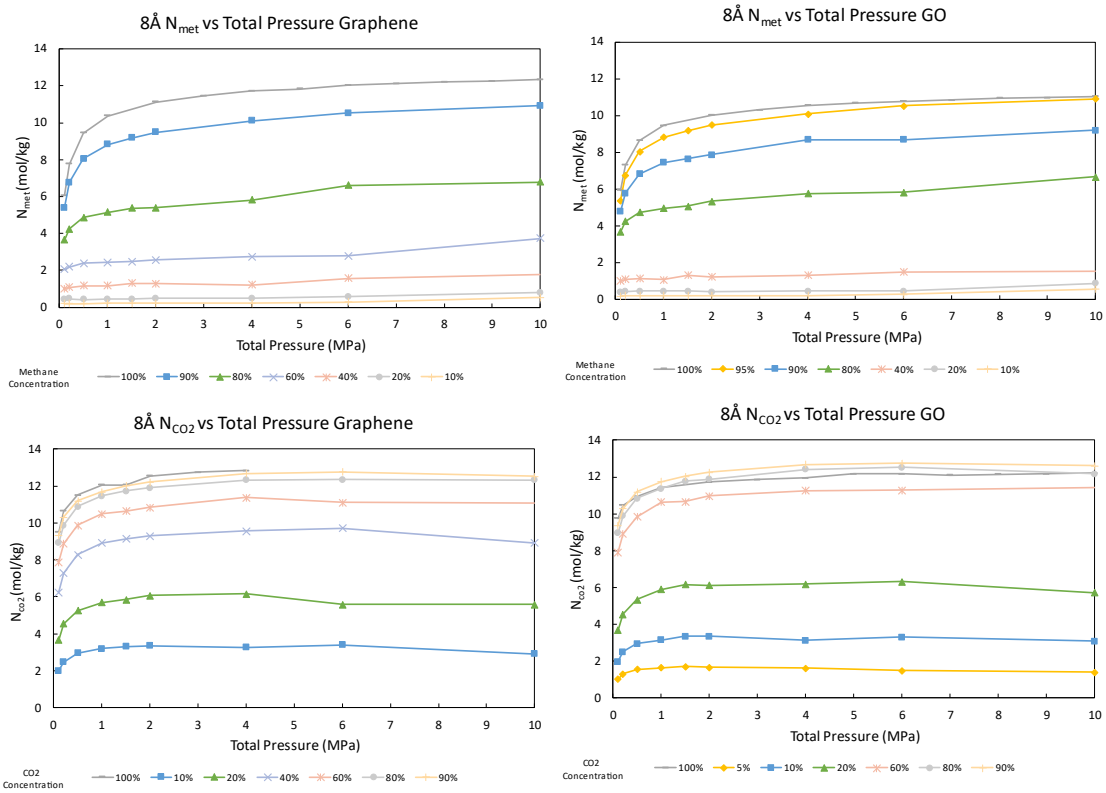

Figure B11 – Adsorption isotherms for CH_4 and CO_2 in GO and graphene in 8Å pores. The y-axes of the graphene and GO graphs were made the same to compare adsorption. The lines are color coded such that the color of the 90% CH_4 line is the same as the 10% CO_2 line.

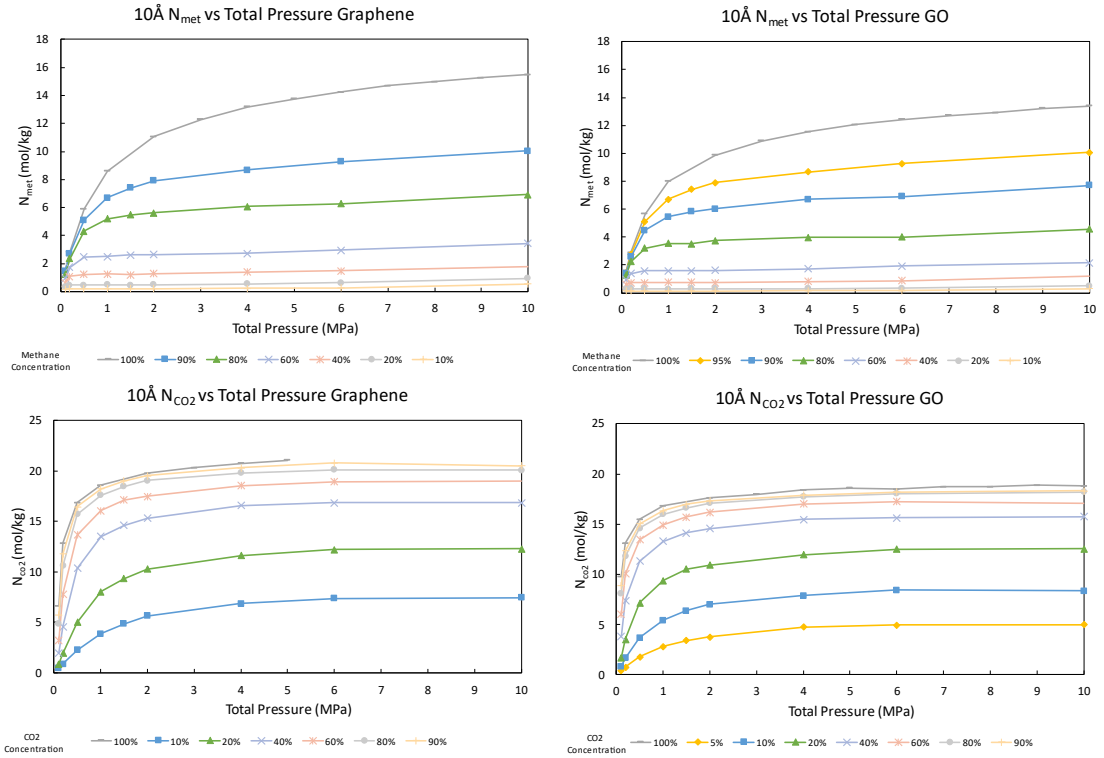

Figure B12 -– Adsorption isotherms for CH_4 and CO_2 in GO and graphene in 10Å pores.

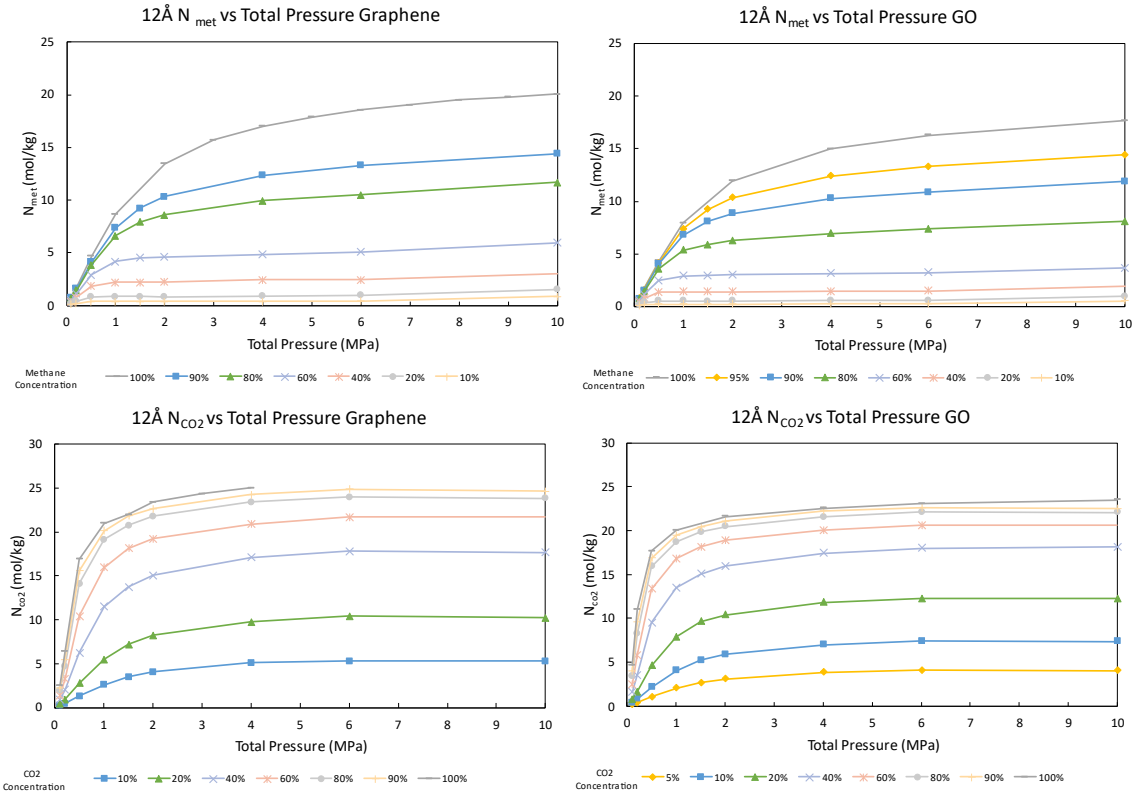

Figure B13 – Adsorption isotherms for CH_4 and CO_2 in GO and graphene in 12Å pores.

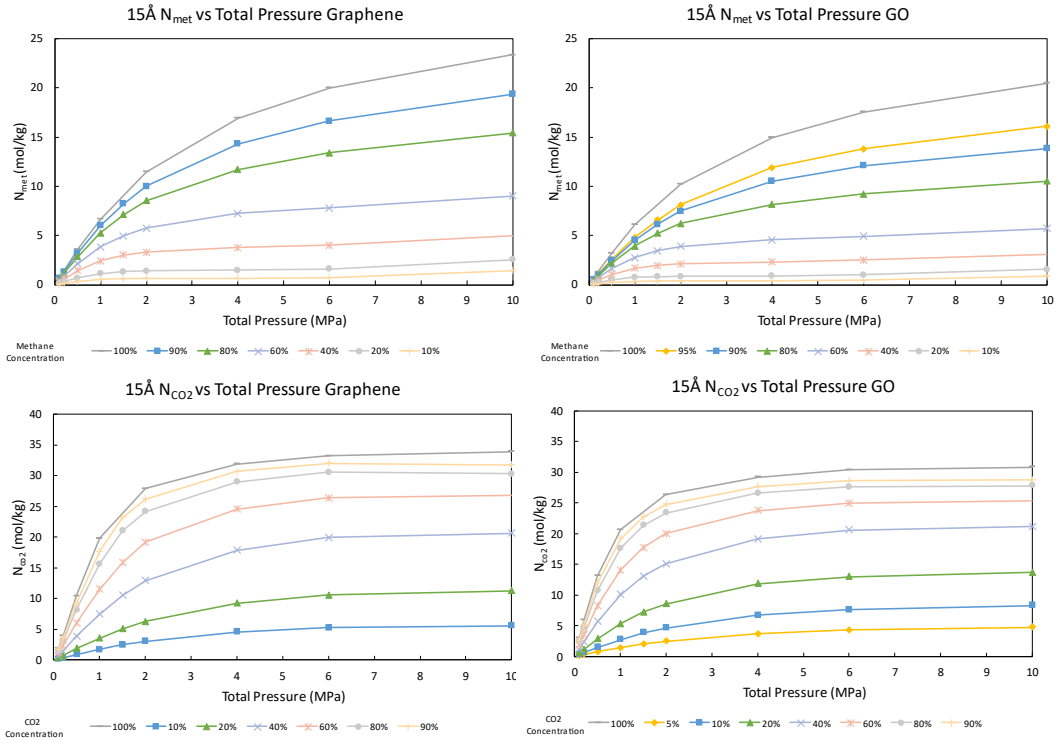

Figure B14 – Adsorption isotherms for CH_4 and CO_2 in GO and graphene in 15Å pores.

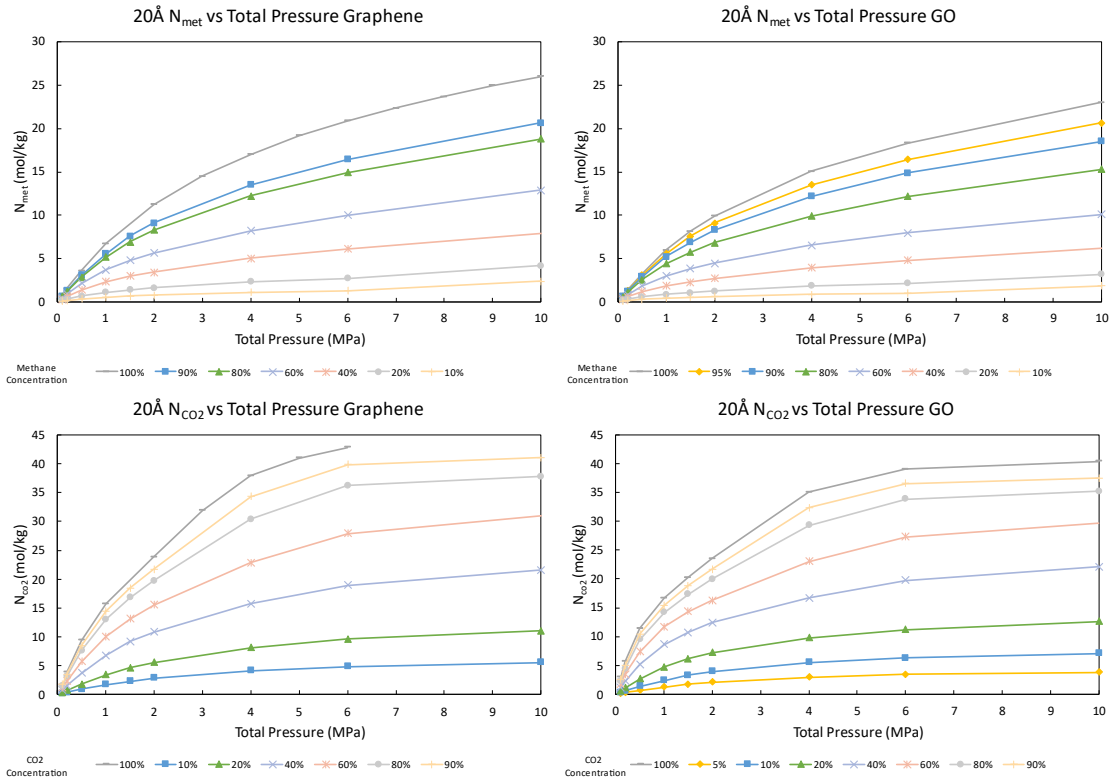

Figure B15 – Adsorption isotherms for CH_4 and CO_2 in GO and graphene in 20Å pores.

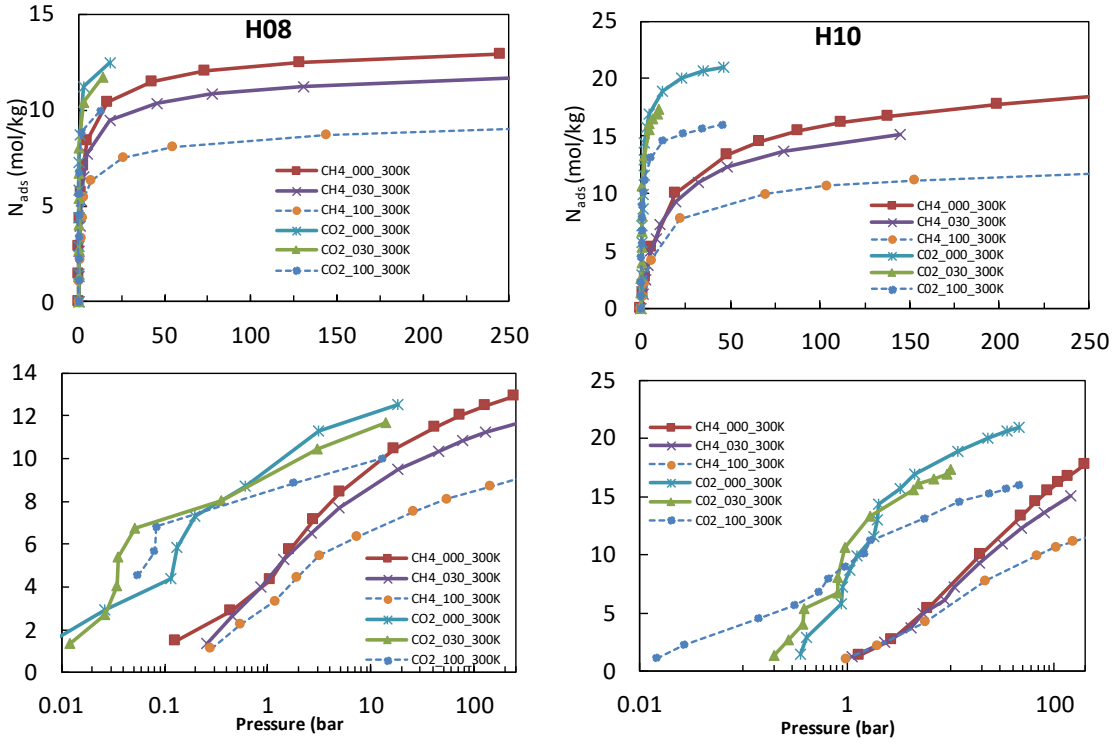

Figure B16 – Adsorption isotherms for homogeneous runs of CH_4 and CO_2 in graphene (000), GO (030), and heavily oxidized GO (100) run in MD. The 8Å pore isotherms are on the left and

the 10Å pore isotherms are on the right. The bottom row of isotherms are log plots of the top row plots.

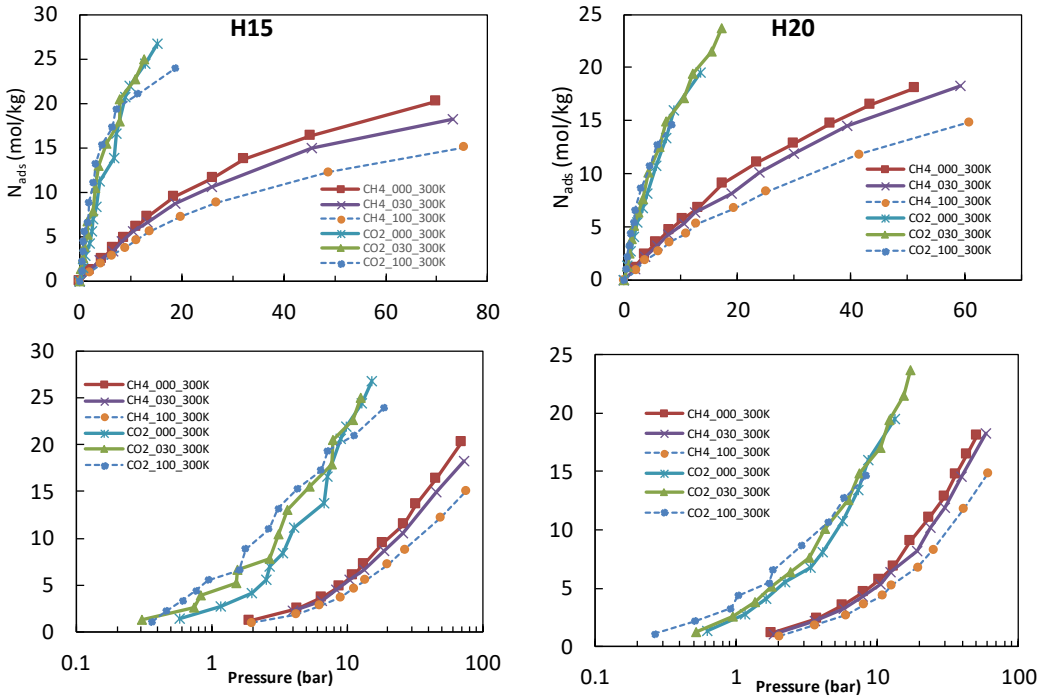

Figure B17 – Sames as **Fig B16** above except for 15Å and 20Å

Appendix C: Program Files

Program C1 - Cov-DBA-Placer6.py.

This program randomly places 3 DBA molecules in a graphene slit-shaped pore by randomly selecting a bond in the bottom layer and top layer that are between 5-7Å laterally apart. It places the DBA in perpendicular to the pore above the bottom bond with no regard for the top bond, allowing the DBA to find its optimal position on its own. This was used in the MD study for **Chapter 1**.

```
import numpy as np
import math
import fileinput
from collections import OrderedDict

################################################################################
#DEFINING VARIABLES
DMIN=5
DMAX=7
OD01Z = 1.473   # shift up the z coordinate of the HO (vs. OH)

CD0Z = 1.562
CD12Z = 2.289
CD1234Y = 1.204
CD34Z = 3.681
CD5Z = 4.408

HD3456Y = 2.148
HD34Z = 1.749
HD56Z = 4.221

BD1Z = 5.970

OD23Y = 1.212
OD23Z = 6.621

HD78Z = 7.587
HD78Y = 1.152
#choice to have input values of number of Epoxy and number of OH groups
#Epoxy_count=input('Enter number of Epoxy: ')
#OH_count=input('Enter number of OH groups: ')

#variables of quantity and charge
BD0_charge='0.4529'
CDB0_charge='0.0623'
CDB1_charge='0.0716'
CDH_charge='-0.188655'
HC_charge='0.1400'
HO_charge='0.3896'
BD1_charge='0.3668'
CSP23_charge='0.232749'
OH_charge='-0.5609'
OC_charge='-0.4176'
DBA_count = 3


#file input and output names
top_psf_file_in = 'graphene2-PBCx.psf'
top_pdb_file_in = 'graphene2-PBCx.pdb'
bottom_psf_file_in = 'graphene1-PBCx.psf'
bottom_pdb_file_in = 'graphene1-PBCx.pdb'
```

```
psf_file_out = 'GO-PBCx.psf'
pdb_file_out= 'GO-PBCx.pdb'

################################################################################
#EMPTY ARRAYS
top_Atom_array=np.array([])
top_Bond_array=np.array([])
top_csp23_atom_array=np.array([])
top_csp23_bond_array=np.array([])
top_atom_key=([])
top_pdb_array=([])

bottom_Atom_array=np.array([])
bottom_Bond_array=np.array([])
bottom_csp23_atom_array=np.array([])
bottom_csp23_bond_array=np.array([])
bottom_atom_key=([])
bottom_pdb_array=([])

################################################################################
#OPEN AND CREATE FILES
#open psf file to read, create new psf file to write
#open pdb file to read, create new pdb file to write
top_psf_file=open(top_psf_file_in, 'r')
top_pdb_file=open(top_pdb_file_in, 'r')

bottom_psf_file=open(bottom_psf_file_in, 'r')
bottom_pdb_file=open(bottom_pdb_file_in, 'r')

new_psf=open(psf_file_out, 'w+')
new_pdb=open(pdb_file_out, 'w+')

################################################################################
#reading psf_file line by line to obtain Atom array and Bond array separately
for line in top_psf_file:
    if line[:4]=='ATOM':
        line=line.strip()
        top_Atom_array=np.append(top_Atom_array, line)

    elif line[:4]=='BOND':
        line=line.strip()
        top_Bond_array=np.append(top_Bond_array, line)

    else:
        continue

for line in bottom_psf_file:
    if line[:4]=='ATOM':
        line=line.strip()
        bottom_Atom_array=np.append(bottom_Atom_array, line)

    elif line[:4]=='BOND':
        line=line.strip()
        bottom_Bond_array=np.append(bottom_Bond_array, line)

    else:
        continue
#getting keys from Atom_array for dictionary (this is the original Atom_array)
for idx in top_Atom_array:
    x=idx[5:9].split()
    top_atom_key=np.append(top_atom_key, x)

for idx in bottom_Atom_array:
    x=idx[5:9].split()
    bottom_atom_key=np.append(bottom_atom_key, x)

#getting array from pdb file
for line in top_pdb_file:
```

```
    if line[0:6]=='ATOM  ':
        line=line.strip()
        top_pdb_array=np.append(top_pdb_array, line)
    else:
        continue

for line in bottom_pdb_file:
    if line[0:6]=='ATOM  ':
        line=line.strip()
        bottom_pdb_array=np.append(bottom_pdb_array, line)
    else:
        continue

#create an array of just atom coordinates from pdb_array
#these are in list form to keep their structure
top_atom_coor=[]
bottom_atom_coor=[]

for idx in top_pdb_array:
    a=idx[30:54].split()
    top_atom_coor.append(a)

for idx in bottom_pdb_array:
    a=idx[30:54].split()
    bottom_atom_coor.append(a)

#take atom numbers and their coordinates and make a dictionary
top_dictionary=dict(zip(top_atom_key, top_atom_coor))
bottom_dictionary=dict(zip(bottom_atom_key, bottom_atom_coor))

###############################################################################
#RANDOM PLACE
#loop through to obtain 10 random bonds for the Epoxy while changing charge and type in atom array
for chosen bonded atoms
#changing atom types to CEPX in order to separate atoms already used in previous bonds
#this is done first by selecting the first bond, comparing to atoms in atom array,
#then taking the second bond, comparing to atoms in atom array,
#and collectively seeing if either of the two have been selected before
#if not, the atoms in the bonds are edited in the atom array and added to a cepx_bond_array for
later use

str_id="{:8s}".format('CSP23') #string id for cepx, comparing to other atom ids
str_id2="{:8s}".format('CSP23') #string id for csp23

#Find a bond for the DBA in the bottom
i=0
while i<DBA_count:
    if i==0:
        bottom_random_bond=np.random.choice(bottom_Bond_array, size=(1,), replace=False)
        for bond_idx in bottom_random_bond:
            new_key=bond_idx[12:19]
            x='{}'.format(bond_idx[6:10])
            ref_atom_coor1=bottom_dictionary.get(x)
            xpos=float(ref_atom_coor1[0])
            ypos=float(ref_atom_coor1[1])
            if ypos < -7:
                for atom_idx1 in bottom_Atom_array:
                    if bond_idx[6:10]==atom_idx1[5:9]:
                        x1=atom_idx1
                        str1=atom_idx1[11:19]
                        charge1=atom_idx1[19:23]
                for atom_idx2 in bottom_Atom_array:
                    if bond_idx[12:16]==atom_idx2[5:9]:
                        x2=atom_idx2
                        str2=atom_idx2[11:19]
                        charge2=atom_idx2[19:23]
                        if str1==str_id or str2==str_id:
                            continue
                        else:
```

```
                            s1=x1.replace(str1, str_id)
                            t1=s1.replace(charge1, CSP23_charge)
                            s2=x2.replace(str2, str_id)
                            t2=s2.replace(charge2, CSP23_charge)
                            bottom_Atom_array=np.core.defchararray.replace(bottom_Atom_array, x1,
t1)
                            bottom_Atom_array=np.core.defchararray.replace(bottom_Atom_array, x2,
t2)
                            bottom_csp23_bond_array=np.append(bottom_csp23_bond_array,
bottom_random_bond)
                            i=i+1

    elif i==1:
        bottom_random_bond=np.random.choice(bottom_Bond_array, size=(1,), replace=False)
        for bond_idx in bottom_random_bond:
            new_key=bond_idx[12:19]
            x='{}'.format(bond_idx[6:10])
            ref_atom_coor1=bottom_dictionary.get(x)
            xpos=float(ref_atom_coor1[0])
            ypos=float(ref_atom_coor1[1])
            if ypos > 7:
                for atom_idx1 in bottom_Atom_array:
                    if bond_idx[6:10]==atom_idx1[5:9]:
                        x1=atom_idx1
                        str1=atom_idx1[11:19]
                        charge1=atom_idx1[19:23]
                for atom_idx2 in bottom_Atom_array:
                    if bond_idx[12:16]==atom_idx2[5:9]:
                        x2=atom_idx2
                        str2=atom_idx2[11:19]
                        charge2=atom_idx2[19:23]
                        if str1==str_id or str2==str_id:
                            continue
                        else:
                            s1=x1.replace(str1, str_id)
                            t1=s1.replace(charge1, CSP23_charge)
                            s2=x2.replace(str2, str_id)
                            t2=s2.replace(charge2, CSP23_charge)
                            bottom_Atom_array=np.core.defchararray.replace(bottom_Atom_array, x1,
t1)
                            bottom_Atom_array=np.core.defchararray.replace(bottom_Atom_array, x2,
t2)
                            bottom_csp23_bond_array=np.append(bottom_csp23_bond_array,
bottom_random_bond)
                            i=i+1
    else:
        bottom_random_bond=np.random.choice(bottom_Bond_array, size=(1,), replace=False)
        for bond_idx in bottom_random_bond:
            for atom_idx1 in bottom_Atom_array:
                if bond_idx[6:10]==atom_idx1[5:9]:
                    x1=atom_idx1
                    str1=atom_idx1[11:19]
                    charge1=atom_idx1[19:23]
            for atom_idx2 in bottom_Atom_array:
                if bond_idx[12:16]==atom_idx2[5:9]:
                    x2=atom_idx2
                    str2=atom_idx2[11:19]
                    charge2=atom_idx2[19:23]
                    if str1==str_id or str2==str_id:
                        continue
                    else:
                        s1=x1.replace(str1, str_id)
                        t1=s1.replace(charge1, CSP23_charge)
                        s2=x2.replace(str2, str_id)
                        t2=s2.replace(charge2, CSP23_charge)
                        bottom_Atom_array=np.core.defchararray.replace(bottom_Atom_array, x1, t1)
                        bottom_Atom_array=np.core.defchararray.replace(bottom_Atom_array, x2, t2)
                        bottom_csp23_bond_array=np.append(bottom_csp23_bond_array,
bottom_random_bond)
```

```
                        i=i+1

print bottom_csp23_atom_array
print bottom_csp23_bond_array
###############################################################################
#NEW ATOM AND BOND ADDITIONS
#add new atoms to atom array while also adding new bonds to bond array and adding bonds to
cepx_bond_array
# first Epoxy

#DBA additions to the array
numb=0
addition=[]
bond_add=[]
while numb<DBA_count:
    if numb<10:
         addition.append('ATOM OD0{}  OXGN    {}'.format(numb, OC_charge))#0
         addition.append('ATOM OD1{}  OXGN    {}'.format(numb, OC_charge))
         addition.append('ATOM BD0{}  BCOO    {}'.format(numb, BD0_charge))
         addition.append('ATOM CD0{}  CG2R61  {}'.format(numb, CDB0_charge))
         addition.append('ATOM CD1{}  CG2R61  {}'.format(numb, CDH_charge))
         addition.append('ATOM CD2{}  CG2R61  {}'.format(numb, CDH_charge))
         addition.append('ATOM CD3{}  CG2R61  {}'.format(numb, CDH_charge))
         addition.append('ATOM CD4{}  CG2R61  {}'.format(numb, CDH_charge))
         addition.append('ATOM CD5{}  CG2R61  {}'.format(numb, CDB1_charge))
         addition.append('ATOM HD3{}  HGR61   {}'.format(numb, HC_charge))
         addition.append('ATOM HD4{}  HGR61   {}'.format(numb, HC_charge))
         addition.append('ATOM HD5{}  HGR61   {}'.format(numb, HC_charge))
         addition.append('ATOM HD6{}  HGR61   {}'.format(numb, HC_charge))
         addition.append('ATOM BD1{}  BCOO    {}'.format(numb, BD1_charge))
         addition.append('ATOM OD2{}  OXGN    {}'.format(numb, OC_charge))
         addition.append('ATOM OD3{}  OXGN    {}'.format(numb, OC_charge))#15
         csp23_bond_idx=bottom_csp23_bond_array[numb]
         bond_add.append('BOND  {}  OD0{}'.format(csp23_bond_idx[6:10], numb))#0
         bond_add.append('BOND  {}  OD1{}'.format(csp23_bond_idx[12:16], numb))
         bond_add.append('BOND  OD0{}  BD0{}'.format(numb, numb))
         bond_add.append('BOND  OD1{}  BD0{}'.format(numb, numb))
         bond_add.append('BOND  BD0{}  CD0{}'.format(numb, numb))
         bond_add.append('BOND  CD0{}  CD1{}'.format(numb, numb))
         bond_add.append('BOND  CD0{}  CD2{}'.format(numb, numb))
         bond_add.append('BOND  CD1{}  HD3{}'.format(numb, numb))
         bond_add.append('BOND  CD1{}  CD3{}'.format(numb, numb))
         bond_add.append('BOND  CD3{}  HD5{}'.format(numb, numb))
         bond_add.append('BOND  CD3{}  CD5{}'.format(numb, numb))
         bond_add.append('BOND  CD5{}  CD4{}'.format(numb, numb))
         bond_add.append('BOND  CD4{}  HD6{}'.format(numb, numb))
         bond_add.append('BOND  CD4{}  CD2{}'.format(numb, numb))
         bond_add.append('BOND  CD2{}  HD4{}'.format(numb, numb))
         bond_add.append('BOND  CD5{}  BD1{}'.format(numb, numb))
         bond_add.append('BOND  BD1{}  OD2{}'.format(numb, numb))
         bond_add.append('BOND  BD1{}  OD3{}'.format(numb, numb))#17
    else:
        print "***** too many DBA molecules ***********"

    for i in range(0,16):
        bottom_Atom_array=np.append(bottom_Atom_array, addition[i])

    for i in range(0,18):
        bottom_Bond_array=np.append(bottom_Bond_array, bond_add[i])

    bottom_csp23_bond_array=np.append(bottom_csp23_bond_array, bond_add[0])
    bottom_csp23_bond_array=np.append(bottom_csp23_bond_array, bond_add[1])

    addition=[]
    bond_add=[]

    numb=numb+1

print bottom_csp23_bond_array
```

```
###############################################################################
#GETTING COORDINATES FOR OEPX, HO, and OH
# creating cepx dictionaries, modyfying the coordinates from the atoms that they are bonded to;
#iterating loop over two indices as O** is bonded to two atoms--which are in order
# averaging x and y coordinates, 1 is added to z coordinate


### Placing DBA molecules ATOM by ATOM
## have to do it separately: first for the OH** then the HO**
## adding 1 to z coordinate for OH** then adding another 1 to z coordinate for HO**
## x,y stay the same
bottom_csp23_dict={}
for bond in bottom_csp23_bond_array:
    if bond[12:14]=='OD':
        new_key=bond[12:19]
        x='{}'.format(bond[6:10])
        ref_atom_coor1=bottom_dictionary.get(x)
        z=format(OD01Z+0.20*(np.random.rand()-0.5), '.3f')
        z_coor=float(ref_atom_coor1[2])+float(z)
        new_val=[ref_atom_coor1[0],ref_atom_coor1[1], z_coor]
        bottom_csp23_dict[new_key]=new_val
    else:
        continue


###############################################################################
#WRITING TO THE NEW PDB
#first the old entries; also getting variables from the old pdb such as segname, mol_number
entries=0
bottom_pdb_file.seek(0)
for line in bottom_pdb_file:

    if line[:4]=='ATOM':
        mol_numb = line[25:26]
        segname = line[17:21]
        seg = segname
        lay = line[72:76]
        new_pdb.write(line)
        entries=entries+1
    else:
        continue

print segname
print lay

#taking new dictionaries, combining them, sorting them, and converting into list
keylist=bottom_csp23_dict.keys()
keylist.sort()

###now writing the new entries with correct form; added_entries continues the tally of atoms
## key is the atom name, coordinates of respected key are retrieved from cepx_dict--the new
combined dictionary
added_entries=entries+1
numb=0
j=0
bond_add=[]
for i in range(0,DBA_count):
        for key in keylist:
                if key=='OD0{}'.format(numb):
                        coor1=bottom_csp23_dict[key]
                        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}
{:6s}{:6s}    {:3s}{:>2s}".format('ATOM', added_entries, key, segname, mol_numb, float(coor1[0]),
float(coor1[1]), float(coor1[2]), '1.00', '0.00', lay, 'O')
                        new_pdb.write(line+'\n')
                        added_entries+=1
                if key=='OD1{}'.format(numb):
                        coor2=bottom_csp23_dict[key]
```

```
                        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}
{:6s}{:6s}    {:3s}{:>2s}".format('ATOM', added_entries, key, segname, mol_numb, float(coor2[0]),
float(coor2[1]), float(coor2[2]), '1.00', '0.00', lay, 'O')
                        new_pdb.write(line+'\n')
                        added_entries+=1

#coordinates for the Boron which is the zero
        Bcoor=[(float(coor1[0])+float(coor2[0]))/2,(float(coor1[1])+float(coor2[1]))/2,float(coor1
[2])+0.966]

#placing the Boron in the pdb file
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'BD0{}'.format(numb), segname, mol_numb,
float(Bcoor[0]), float(Bcoor[1]), float(Bcoor[2]), '1.00', '0.00', lay, 'B')
        new_pdb.write(line+'\n')
        added_entries+=1

#placing the Carbon ring in the pdb file
#CD0
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'CD0{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1], Bcoor[2]+CD0Z, '1.00', '0.00', lay, 'C')
        new_pdb.write(line+'\n')
        added_entries+=1
#CD1
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'CD1{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1]+CD1234Y, Bcoor[2]+CD12Z, '1.00', '0.00', lay, 'C')
        new_pdb.write(line+'\n')
        added_entries+=1
#CD2
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'CD2{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1]-CD1234Y, Bcoor[2]+CD12Z, '1.00', '0.00', lay, 'C')
        new_pdb.write(line+'\n')
        added_entries+=1
#CD3
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'CD3{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1]+CD1234Y, Bcoor[2]+CD34Z, '1.00', '0.00', lay, 'C')
        new_pdb.write(line+'\n')
        added_entries+=1
#CD4
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'CD4{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1]-CD1234Y, Bcoor[2]+CD34Z, '1.00', '0.00', lay, 'C')
        new_pdb.write(line+'\n')
        added_entries+=1
#CD5
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'CD5{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1], Bcoor[2]+CD5Z, '1.00', '0.00', lay, 'C')
        new_pdb.write(line+'\n')
        added_entries+=1

#Attaching Hydrogens to the ring
#HD3
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'HD3{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1]+HD3456Y, Bcoor[2]+HD34Z, '1.00', '0.00', lay, 'H')
        new_pdb.write(line+'\n')
        added_entries+=1
#HD4
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'HD4{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1]-HD3456Y, Bcoor[2]+HD34Z, '1.00', '0.00', lay, 'H')
        new_pdb.write(line+'\n')
        added_entries+=1
#HD5
```

```
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'HD5{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1]+HD3456Y, Bcoor[2]+HD56Z, '1.00', '0.00', lay, 'H')
        new_pdb.write(line+'\n')
        added_entries+=1
#HD6
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'HD6{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1]-HD3456Y, Bcoor[2]+HD56Z, '1.00', '0.00', lay, 'H')
        new_pdb.write(line+'\n')
        added_entries+=1

#Finishing up the structure with the Boron, two Oxygens, and two Hyrdogens
#BD1
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'BD1{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1], Bcoor[2]+BD1Z, '1.00', '0.00', lay, 'B')
        new_pdb.write(line+'\n')
        added_entries+=1
#OD2
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'OD2{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1]+OD23Y, Bcoor[2]+OD23Z, '1.00', '0.00', lay, 'O')
        new_pdb.write(line+'\n')
        added_entries+=1
#OD3
        line="{:6s}{:5d} {:^4s} {:4s}    {:1s}    {:8.3f}{:8.3f}{:8.3f}  {:6s}{:6s}
{:3s}{:>2s}".format('ATOM', added_entries, 'OD3{}'.format(numb), segname, mol_numb, Bcoor[0],
Bcoor[1]-OD23Y, Bcoor[2]+OD23Z, '1.00', '0.00', lay, 'O')
        new_pdb.write(line+'\n')
        added_entries+=1

        #Find a bond for the DBA in the top
        found='FALSE'
        while found == 'FALSE':
                top_random_bond=np.random.choice(top_Bond_array, size=(1,), replace=False)
                for bond_idx in top_random_bond:
                        new_key=bond_idx[12:19]
                        x='{}'.format(bond_idx[6:10])
                        ref_atom_coor1=top_dictionary.get(x)
                        xpos=float(ref_atom_coor1[0])
                        ypos=float(ref_atom_coor1[1])
                        distance = math.sqrt((float(Bcoor[0])-xpos)**2 + (float(Bcoor[1])-
ypos)**2)
                       print "Bcoorx is {}, Bcoory is {}".format(Bcoor[0], Bcoor[1])
                       print "xpos is {}, ypos is {}, and distance is {}".format(xpos, ypos,
distance)
                       if distance < DMIN or distance > DMAX:
                               continue
                       else:
                               for atom_idx1 in top_Atom_array:
                                       if bond_idx[6:10]==atom_idx1[5:9]:
                                               x1=atom_idx1
                                               str1=atom_idx1[11:19]
                                                charge1=atom_idx1[19:23]
                               for atom_idx2 in top_Atom_array:
                                       if bond_idx[12:16]==atom_idx2[5:9]:
                                                x2=atom_idx2
                                                str2=atom_idx2[11:19]
                                                charge2=atom_idx2[19:23]
                                                if str1==str_id or str2==str_id:
                                                    continue
                                                else:
                                                    s1=x1.replace(str1, str_id)
                                                    t1=s1.replace(charge1, CSP23_charge)
                                                    s2=x2.replace(str2, str_id)
                                                    t2=s2.replace(charge2, CSP23_charge)

top_Atom_array=np.core.defchararray.replace(top_Atom_array, x1, t1)
```

```
top_Atom_array=np.core.defchararray.replace(top_Atom_array, x2, t2)

top_csp23_bond_array=np.append(top_csp23_bond_array, top_random_bond)
                                                    found = 'TRUE'

        #print top_csp23_atom_array
        print top_csp23_bond_array

        #-------------------------------------------------------------
        csp23_bond_idx=top_csp23_bond_array[j]
        bond_add.append('BOND  {}  OD3{}'.format(csp23_bond_idx[6:10], j))
        bond_add.append('BOND  {}  OD2{}'.format(csp23_bond_idx[12:16], j))

        top_Bond_array=np.append(top_Bond_array, bond_add[0])
        top_Bond_array=np.append(top_Bond_array, bond_add[1])

        print "This is bond_add {}    {}    ".format(bond_add[0],bond_add[1])

        bond_add=[]

        j=j+1
        #-------------------------------------------------------------

        #-------------------------------------------------------------
        numb=numb+1

top_pdb_file.seek(0)
for line in top_pdb_file:
    if line[:4]=='ATOM':
        mol_numb = line[25:26]
        segname = line[17:21]
        seg = segname
        lay = line[72:76]
        index=line[6:12]
        output_line = line[:6] + "{:5d} ".format(added_entries) + line[12:17] + "{:4s}
{:1s}".format('GRA1', '1') + line[26:72] + "LAY1 " + line[77:]
        new_pdb.write(output_line)
        added_entries=added_entries+1
    else:
        continue
print segname
print lay


#writing END to pdb file to complete the writing of the new pdb file
new_pdb.write('END')

################################################################################
#WRITING NEW PSF FILE
#going back to the top of the psf file; writing with the new atom and bond arrays

bottom_psf_file.seek(0)
top_psf_file.seek(0)

#this adds new MASS lines
#new_psf.write("MASS     1 CA     12.01100 C ! aromatic C"+'\n')
#new_psf.write("MASS     1 CSP23  12.01100 C ! aromatic C"+'\n')
#new_psf.write("MASS     1 CEPX   12.01100 C ! aromatic C"+'\n')
#new_psf.write("MASS     1 HO      1.00800 H !"+'\n')
#new_psf.write("MASS     1 OH     15.99940 O !"+'\n')
#new_psf.write("MASS     1 OEPX   15.99940 O !"+'\n')
#new_psf.write("MASS     1 BCOO   10.811   B ! B in OOBC, charge +0.45 (B3LYP/6-61G(d) C.W.
2017)"+'\n')

for line in bottom_psf_file:
    if line[:4]=="ATOM":
        break
    else:
```

```
        new_psf.write(line)

for line in top_psf_file:
    if line[:4]=="ATOM":
        break

b_idx=0
for a_idx in bottom_Atom_array:
    new_psf.write(a_idx+'\n')
    b_idx=b_idx+1

for a_idx in top_Atom_array:
    new_psf.write(a_idx+'\n')

for line in bottom_psf_file:
    if line[:4]=="ATOM":
        continue
    elif line[:4]=="BOND":
        break
    else:
        new_psf.write(line)

d_idx=0
for c_idx in bottom_Bond_array:
    if c_idx != 'GROUP':
        new_psf.write(c_idx+'\n')
        d_idx=d_idx+1

for line in top_psf_file:
    if line[:4]=="ATOM":
        continue
    elif line[:4]=="BOND":
        break
    else:
        new_psf.write(line)

for c_idx in top_Bond_array:
    new_psf.write(c_idx+'\n')

################################################################################
#close the files
bottom_psf_file.close()
top_psf_file.close()
new_psf.close()
bottom_pdb_file.close()
top_psf_file.close()
new_pdb.close()
```

Program C2 - com_pz_GO-201806.tcl:

This program records the z-position (height) of every carbon atom in the graphene layers for every frame in the MD study from **Chapter 1**.

```
#### To run this file, type 'vmd -dispdev none -e com-distro.tcl' in a terminal window
#### Alternatively, type 'source com-distro.tcl' into the TK Console of VMD.
####

#set BASEFILE graphene+fluidDBA+gas
#set BASEFILE STRUCTURE-OUT
#set NGAS 100
##############################################################################################################
#### BASIC PARAMETERS
####
set PZ [open "GO-PZ.dat" w]

set cutoff    12.0      ;#### NOTICE: USER PREFERENCE:  CUT-OFF (defines area on "edge" of pores)
#set frameSkip 200      ;#### NOTICE: USER PREFERENCE   Defines number of frames which will be
skipped in analysis.
set frameSkip 1000

##############################################################################################################
### LOAD MOLECULES
###
#mol new $BASEFILE-OUTPUT.psf                   ;#### Load file -OUT.psf
mol new STRUCTURE+GAS_IN.psf
#mol addfile $BASEFILE_OUTPUT.dcd waitfor all   ;#### Load file GO-OUT.dcd and wait
mol addfile STRUCTURE+GAS-OUT.dcd waitfor all
### READ SOME STUFF FROM RUN
###
set nFrames [molinfo top get numframes] ;#### Create a variable nFrames equal to the number of
frames observed



#########################################################
###Histogram of z coordinate

set structure [atomselect top "type CSP23 or type CA"]
puts "done with atomselect"
set structatomnumber [expr [$structure num]]
puts "done with structnumber"
for {set f $frameSkip} {$f < 1 + $nFrames } {incr f 1} {
puts $f
    for {set i 0} {$i < [expr $structatomnumber]} {incr i 1} {
#puts  $i
                set indxupdated [lindex [$structure list] $i]
        set sel [atomselect top "index $indxupdated" frame $f]
            set com [measure center $sel weight mass]
                set zcom [lindex $com 2]
                puts $PZ "$zcom"
        }
}


close $PZ
quit
```

Program C3 - vmd-com-distro-20190604.tcl:

The purpose of this program is to get CH_4 densities in specified regions including inside the pore, gas phase, and pore edge. It records the cartesian coordinates of the center-of-mass for every CH_4 (or xenon with a quick name change in the code) every frame. Based on predefined regions, it assigns the CH_4 molecules to one of the region from which the count is averaged over the total number frames. From there we can get a density to translate to pressure. This is used in the MD studies of both **Chapter 2** and **Chapter 3**.

```
#### This script can no longer be run from terminal window.
#### It must be run from TK console.
#### To run the script, type 'source vmd-com-distro-201711.tcl'
#### We need a fast way to export information.
#### NOTE: This program loads files $filename.psf $filename.dcd automatically.
####
package require pbctools
set filename STRUCTURE+GAS

puts "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
puts "Program begins..."
puts "will read $filename.psf and $filename.dcd "
puts "will write to $filename-DISTRO.log and $filename-PZ.dat"

##############################################################################################################
### LOAD MOLECULES
###
mol new STRUCTURE+GAS_IN.psf                                        ;#### Load file $filename.psf
mol addfile STRUCTURE+GAS-OUT.dcd waitfor all    ;#### Load file $filename.dcd and wait for
completion
puts "read files"
##############################################################################################################

set DISTRO [open "STRUCTURE+GAS-analysis-DISTRO.dat" w]
set datmet [open "STRUCTURE+GAS-analysis-PZ.dat" w]

#set TEST [open "TEST.log" w]
###set dat [open "NXXX-pz3.dat" w]       ;#### NOTICE: This was removed such that the filename is
more specific
##############################################################################################################
#### BASIC PARAMETERS
####
set cutoff    12.0       ;#### NOTICE: USER PREFERENCE:  CUT-OFF (defines area on "edge" of pores)
set frameSkip 1000       ;#### NOTICE: USER PREFERENCE    Defines number of frames which will be
skipped in analysis
set framelast 0          ;#### WARNING:  on 32 bit machines one may have to "split" the
calculations in parts
                                      ;#### SET frame last 0 to run to end of file
### READ SOME STUFF FROM RUN
###
set nFrames [molinfo top get numframes] ;#### Create a variable nFrames equal to the number of
frames observed

  # Create selection of first carbon atom in each layer, and get the z-coordinate to define region
boundary conditions
set z1 [expr 10]
set z2 [expr 1]

set cell [pbc get -namd]

set LxPBC [expr 21.3]    ;#### NOTICE:  We now read from DCD file.
```

```
set LyPBC [expr 37.22]  ;#### NOTICE:  This REQUIRES we run script in TK console!!!!
set LzPBC [expr 50]

set GasZBoundary [expr -16]
#########################################################################################################
### DETERMINE "Volumes"
###
  # Measure minmax of graphene layer, and define boundaries in y-direction (outside of slits)
  # skinDepth (in Angstroms) accounts for molecules just at the edge, not exactly in slit


### THIS IS INCORRECT, NEGLECTS SMALL "link" between edges of PBC
#set xmin [lindex [lindex $mm 0] 0]
#set xmax [lindex [lindex $mm 1] 0]

### ymin/maxPore is the size of the graphene, can be read from it
### the ymin/maxEdge defines the region on the edge (i.e., not inside the pore, but within cutoff)
set yminPore [expr -8.608]
set ymaxPore [expr 8.608 ]
set yminEdge [expr -8.608 - 2]
set ymaxEdge [expr 8.608 + 2]
set ymin [expr -$LyPBC/2]
set ymax [expr $LyPBC/2]
set xmin [expr -$LxPBC/2]
set xmax [expr $LxPBC/2]
set zmin [expr -$LzPBC/2]
set zmax [expr $LzPBC/2]

### NOT USED, MAY BE USEFUL AT SOME POINT
#set allatoms [atomselect top all]
#set mmall [measure minmax $allatoms]    ;#### The min-max for all the atoms
#set zmin [lindex [lindex $mmall 0] 2]
#set zmax [lindex [lindex $mmall 1] 2]

### CALCULATE VOLUMES
### Volume is computational cell
### Volume 1 is space exactly between graphene layers
### Volume 2 is the two Angstrom wide volume outside the pore.
### Volume 3 is the "side gas" region
### Volume 4 is the "bottom gas" region
### Volume 5 is the adsorption region above and below the graphene layers
set volume  [expr abs($xmin-$xmax)*abs($ymin-$ymax)*abs($zmax-$zmin)]
set volume1 [expr abs($xmin-$xmax)*abs($yminPore-$ymaxPore)*abs($z1-$z2)]
set volume2 [expr abs($xmin-$xmax)*2*abs($yminEdge-$yminPore)*abs($z1-$z2)]
set volume3 [expr abs($xmin-$xmax)*2*abs($yminEdge-$ymin)*abs($zmax-$GasZBoundary)]
set volume4 [expr $LxPBC*$LyPBC*abs($zmin-$GasZBoundary)]
set volume5 [expr $volume-$volume1-$volume2-$volume3-$volume4]

puts $DISTRO "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
puts $DISTRO ""
puts $DISTRO "xmin, xmax:        $xmin   $xmax"
puts $DISTRO "yminPore, ymaxPore:        $yminPore       $ymaxPore"
puts $DISTRO "yminEdge, ymaxEdge:        $yminEdge       $ymaxEdge"
puts $DISTRO "ymin, ymax:        $ymin   $ymax"
puts $DISTRO "z1, z2:    $z1     $z2"
puts $DISTRO "zmin, zmax:        $zmin   $zmax"
puts $DISTRO "GasZBoundary:    $GasZBoundary"
puts $DISTRO ""
puts $DISTRO "VOLUMES (total, pore, edge, side gas, bottom gas, adsorption region)"
puts $DISTRO "$volume    $volume1$volume2$volume3$volume4$volume5"
puts $DISTRO ""
puts $DISTRO "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"

puts "got all parameters"
```

```
#########################################################################################################
#########################################################################################################
#########################################################################################################
### CALCULATE NUMBER OF MOLECULES IN EACH VOLUME

#########################################################################################################
###############
### METHANE ###
###############
set methane [atomselect top "type CG341"]                ;#### Creates a selection of atoms which
includes only METHAENE types
                                ;#### Used in arithmetic, to determine the number of checks to be
conducted
set nMET [expr [$methane num]]   ;#### Number of atoms in all methane moleclues divided by number
in each moleclue

set countMET1 0.0                                        ;#### A counter, initially zero, to
count the number of moleclues in volume one
set countMET2 0.0                                        ;#### A counter, initially zero, to
count the number of moleclues in volume two
set countMET3 0.0                                        ;#### A counter, initially zero, to
count the number of moleclues in volume three
set countMET4 0.0                                        ;####
set countMET5 0.0                                        ;####

#####################################################  ADDED BECAUSE OF MEMORY ISSUE ON 32 BIT
if {$framelast > 0.0} {
        set framestop $framelast
} else {
        set framestop $nFrames
}


puts $DISTRO "framestart, framestop: $frameSkip  $framestop"
puts "framestart, framestop: $frameSkip  $framestop"

#####################################################
for {set f $frameSkip} {$f < $framestop} {incr f 1} {    ;####Cycle through frames
#puts $DISTRO "$f"

        for {set i 0} {$i < [expr $nMET]} {incr i} {     ;####Cycle through moleclues, making
each a new atomselect object
        set indxupdated [lindex [$methane list] $i]
        set sel [atomselect top "index $indxupdated" frame $f]
        set com [measure center $sel weight mass]
        set xcom [lindex $com 0]
        set ycom [lindex $com 1]
        set zcom [lindex $com 2]
        $sel delete

#########################################################################
### COUNTERS

        if {$zcom < -16} {
                set countMET4 [expr $countMET4+1]
        } elseif {$ycom > $ymaxEdge || $ycom < $yminEdge} {
                set countMET3 [expr $countMET3+1]
        } elseif {$zcom < $z1 && $zcom > $z2 && $ycom > $ymaxPore && $ycom < $ymaxEdge} {
                set countMET2 [expr $countMET2+1]
        } elseif {$zcom < $z1 && $zcom > $z2 && $ycom < $yminPore && $ycom > $yminEdge} {
                set countMET2 [expr $countMET2+1]
        }  elseif {$zcom < $z1 && $zcom > $z2} {
                set countMET1 [expr $countMET1+1]
```

```
        } else {set countMET5 [expr $countMET5+1]
          }

#########################################################################
### P(z)
        if {$ycom < $ymaxPore && $ycom > $yminPore} {
                if {
                                ($zcom < $z1) &&
                                ($zcom > $z2)           } {
                puts $datmet "$zcom"
                }
        }

}}                                                      ; ## close loops
puts $DISTRO "DONE WITH CALCULATION"
set countMET [expr $countMET1 + $countMET2 + $countMET3 + $countMET4 + $countMET5]

#########################################################################################################
### PRINT INFORMATION
puts $DISTRO ""
puts $DISTRO "METHANE counts: (total, pore, edge, side gas, bottom gas, adsorption region)"
puts $DISTRO "$countMET     $countMET1     $countMET2     $countMET3     $countMET4
        $countMET5"
puts $DISTRO ""
puts $DISTRO "METHANE counts (per frame): "
puts $DISTRO "[expr $countMET/($framestop-$frameSkip)]  [expr  $countMET1/($framestop-$frameSkip)]
        [expr $countMET2/($framestop-$frameSkip)]         [expr $countMET3/($framestop-
$frameSkip)]    [expr $countMET4/($framestop-$frameSkip)]        [expr $countMET5/($framestop-
$frameSkip)]"
puts $DISTRO " "
puts $DISTRO "METHANE average densities (nm^-3) (note: neglecting skin depths, need to adjust by
vdW skin depth for calculation of excess adsorption"
puts $DISTRO "(total, pore, edge, gas, upper edge)"

puts $DISTRO "[expr 1000*$countMET/($nFrames-$frameSkip)/$volume][expr 1000*$countMET1/($nFrames-
$frameSkip)/$volume1]   [expr 1000*$countMET2/($nFrames-$frameSkip)/$volume2]    [expr
1000*$countMET3/($nFrames-$frameSkip)/$volume3]  [expr 1000*$countMET4/($nFrames-
$frameSkip)/$volume4]   [expr 1000*$countMET5/($nFrames-$frameSkip)/$volume5]"

puts $DISTRO " "
puts $DISTRO "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
puts $DISTRO " "

puts ""
puts "METHANE counts: (total, pore, edge, side gas, bottom gas, adsorption region)"
puts "$countMET     $countMET1     $countMET2     $countMET3     $countMET4       $countMET5"
puts ""
puts "METHANE counts (per frame): "
puts "[expr $countMET/($framestop-$frameSkip)]   [expr $countMET1/($framestop-$frameSkip)]
        [expr $countMET2/($framestop-$frameSkip)]         [expr $countMET3/($framestop-
$frameSkip)]    [expr $countMET4/($framestop-$frameSkip)]        [expr $countMET5/($framestop-
$frameSkip)]"
puts " "
puts "METHANE average densities (nm^-3) (note: neglecting skin depths, need to adjust by vdW skin
depth for calculation of excess adsorption"
puts "(total, pore, edge, gas, upper edge)"

puts "[expr 1000*$countMET/($nFrames-$frameSkip)/$volume][expr 1000*$countMET1/($nFrames-
$frameSkip)/$volume1]   [expr 1000*$countMET2/($nFrames-$frameSkip)/$volume2]    [expr
1000*$countMET3/($nFrames-$frameSkip)/$volume3]  [expr 1000*$countMET4/($nFrames-
$frameSkip)/$volume4]   [expr 1000*$countMET5/($nFrames-$frameSkip)/$volume5]"

puts " "
puts "XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
```

```
puts " "

close $DISTRO
close $datmet

exit
#################################################################################################
#################################################################################################
#################################################################################################
```

Program C4 – *picture-STRUCTURE+GAS.tcl*

The purpose of this program is to automatically take a picture of the end of MD simulations GOF structures found in **Chapter 2**.

```
mol new STRUCTURE+GAS_IN.psf
mol addfile STRUCTURE+GAS-OUT.dcd waitfor all
####
pbc box -center origin
####
display projection orthographic

rotate z by -90
rotate x by -90
scale by 0.8

#######
## Select Graphenes
mol modselect 0 0 segname LAY0 LAY1 LAY2 LAY3 LAY4 LAY5 LAY6 LAY7 LAY8 LAY9 MET
mol representation
mol modstyle 0 0 CPK
mol modcolor 0 0 Name

#speed for animation
display resize 3600 1080
axes location off
scale to 0.055
animate goto end
sleep 5
render snapshot end.tga

exit
```

Program C5 – *graphene_spacing.py*

This program calculates the GOF pore spacing for the MD studies in **Chapter 2**.

```
bottom_count = 0 #counts how many instances an atom is on bottom for averaging at the
end
top_count = 0 #counts how many instances an atom is on top, if they don't match it
likely means the pore is lopsided
bottom_h_total = 0
top_h_total = 0
import os.path

OUT=open("average_spacing.txt","w")
OUT.write("config spacing\n")

for i in range(22,122):
        print "{}".format(i)

        filename =("{}/T300-N000/GO-PZ.dat".format(i))
        if os.path.exists(filename):
                IN=open("{}/T300-N000/GO-PZ.dat".format(i),"r")

                bottom_count = 0.0
                top_count = 0.0
                bottom_h_total = 0.0
                top_h_total = 0.0
                average_space = 0.0

                for line in IN:
                        zVal = float(line.strip())
                        if zVal > -2 and zVal < 3:
                                bottom_count = bottom_count + 1
                                bottom_h_total = bottom_h_total + zVal

                        elif zVal > 3:
                                top_count = top_count + 1
                                top_h_total = top_h_total + zVal

                if top_count != 0 and bottom_count != 0:
                        average_top=top_h_total/top_count
                        average_bottom=bottom_h_total/bottom_count
                        average_space=average_top-average_bottom

                if average_space > 7 and average_space < 11:
                        OUT.write("{} {}\n".format(i, average_space))

                IN.close()
```

Program C6 – *RunTests.sh*

This program is a sample of what was used to run the hundreds of test GOF configurations from **Chapter 2**.

```
for i in {22..122}
do
        mkdir $i
        cp -r T300-N000 $i
        cd $i/T300-N000
        python Cov-DBA-Placer6.py
        psfgen < STRUCTURE+GAS.IN
        vmd -dispdev none -e STRUCTURE-FixBETA.tcl
        namd2 +p11 STRUCTURE+GAS.namd > STRUCTURE+GAS.log
        cd ../..
done
```

Program C7 – *RunAll.sh*

This program is sample of what was used to get the data for the adsorption isotherms in the MD studies from **Chapter 2** and **Chapter 3**.

```
for i in 1 2 3 4 5 6 7 8 9 10
do
        cd $i
        for j in 010 025 050 075 100 150 200 300 400
        do
                cd T300-N$j
                ./Gas_Generator-CW.exe <<--EOF
                $j
                21.3
                37.22
                -10 40
                -EOF

                psfgen < STRUCTURE+GAS.IN
                vmd -dispdev none -e STRUCTURE-FixBETA.tcl
                namd2 +p10 STRUCTURE+GAS.namd > STRUCTURE+GAS.log
                cd ..
        done

        cd ..
done
```

Program C8 - *GasMolecules-201910.f*

This program random places gas molecules used in MD studies from **Chapter 2** and **Chapter 3**. It places gas molecules such that they are no closer than 2Å from the adsorbent or any other gas molecule.

```
      program Gas_MoleculesCW

!!! compilation command:
!!! gfortran -O -fbounds-check -ffree-form Gas_Molecules-201910.f -o Gas_Molecules-
201910.exe
!!! running the code:  ./Gas_Molecules-201910.exe
!!!
!!! generates initial configuration for natural gas mixture in the system
!!!
!!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!! basic parameters

      implicit none

      real, parameter :: RMIN=2.00        ! minimum possible distance between atoms
                                         ! (eventually could be substituted by Rmin vdW)
      real xij,yij,zij,rij2               ! to calculate distance between atoms

      integer, parameter :: ngasmax=20000    ! maximum number of GAS atoms to be
generated
      integer, parameter :: namax=200        ! maximum number of atoms to be read from
GAS
      integer, parameter :: nstrucmax = 10000 ! maximum number of atoms to be read from
STRUCTURE

1000  FORMAT (A4,2X,I5,1X,A4,1X,A3,1X,1X,I4,1X,3X,F8.3,F8.3,F8.3,F6.2,F6.2,6X,A4,A2)
1001  format (A6,   I5,1X,A4,1X,A3,1X,1X,I4,1X,3X,F8.3,F8.3,F8.3,F6.2,F6.2,6X,A4,A2)


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! DECLARATIONS RELATED TO MISC STUFF
!!!
      integer iclock,iseed                          ! seeds for rand()

      integer i,j,k,l,m,n,iatom,jprev,inew,icollision
      real pi

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! DECLARATIONS RELATED TO READING THE GAS & STRUCTURE IN
!!!
      integer imolinfile,iresmolec(namax),iatomsmolec,imolec  ! for gas molecule in
      real xmolec(namax),ymolec(namax),zmolec(namax)          ! for gas molecule in
      character*6 ATOMmolec(namax)                            ! for gas molecule in
      character*4 ATOMNAMEmolec(namax)                        ! for gas molecule in
      character*3 RESNmolec(namax)                            ! for gas molecule in
      character*4 SEGMmolec(namax)                            ! for gas molecule in
      character*2 ELEMmolec(namax)                            ! for gas molecule in
      character*20 molinfile                                  ! for gas molecule in
```

```
      integer istructinfile,iresstruct(ngasmax),iatomsstruct,istruct  ! for structure
in
      real xstruct(nstrucmax),ystruct(nstrucmax),zstruct(nstrucmax)     ! for structure
in
      character*6 ATOMstruct(nstrucmax)                                 ! for structure
in
      character*4 ATOMNAMEstruct(nstrucmax)                             ! for structure
in
      character*3 RESNstruct(nstrucmax)                                 ! for structure
in
      character*4 SEGMstruct(nstrucmax)                                ! for structure
in
      character*2 ELEMstruct(nstrucmax)                                 ! for structure
in
      character*20 structureinfile                                      ! for structure
in

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! DECLARATIONS RELATED TO CREATING THE GAS (OUT)
!!!
      integer igasfile,iresgas(ngasmax),iatomsgas,igas        ! for gas OUT
      real xgas(ngasmax),ygas(ngasmax),zgas(ngasmax)          ! for gas OUT
      character*6 ATOMgas(ngasmax)                            ! for gas OUT
      character*4 ATOMNAMEgas(ngasmax)                        ! for gas OUT
      character*3 RESNgas(ngasmax)                            ! for gas OUT
      character*4 SEGMgas(ngasmax)                            ! for gas OUT
      character*2 ELEMgas(ngasmax)                            ! for gas OUT
      character*20 gasOUTfile                                 ! for gas OUT
      integer Ngas                                            ! for gas OUT
      real LzMIN,LzMAX,Ly,Lx,deltax,deltay,deltaz             ! for gas OUT
      real eulerphi,eulerpsi,eulertheta                       ! for gas OUT
      real xtemp,ytemp,ztemp                                  ! for gas OUT


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!! MISC DECLARATIONS
      character*80 readline
      real occ,beta
      pi=4.*atan(1.)

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! SEEDING THE RAND() generator
      iseed = 128841          ! iseed value, any integer will work
      xtemp = rand(iseed)     ! initialize random generator
                             ! (different iseeds produce different sequences)
      call system_clock(COUNT=iclock)
     ! xtemp = rand(iclock)  ! this will seed every run differently (from the system
clock)


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Input GAS molecule file
!!!
!!! reads file molinfile (GAS_MOLECULE_IN.pdb)
!!! stores:
!!!   ATOMNAMEmolec(i) - atom names (C1,H1X,H1Y,H1Z,H1W)
!!!   RESNmolec(i)     - RESIDUE (NC1)
!!!   iresmolec(i)        - molecule number (should be just 1)
!!!   xmolec(i), ymolec(i), zmolec(i) - coordinates
!!!   occ              - occupation, should be 1.00
!!!   beta             - beta, should be 0.00
```

```
!!!   SEGMmolec(i)     - segment (MET)
!!!   ELEMmolec(i)     - atom type (C, H)
!!! at the end of the read, should also provide
!!!   iatomsmolec = number of atoms
!!!   imolec = number of molecules
!!!
!!!
      molinfile = 'GAS_MOLECULE_IN.pdb'  ! file where molecule is read from
      imolinfile = 11                    ! unit where molecule is read from
      open(unit=imolinfile,file=molinfile,status='OLD',form='formatted', &
            access='sequential',action='read')
!!! now read input molecule
      i = 1
      do j=1,namax
        read (imolinfile,'(A)',end=10) readline
        if ((readline(1:4) .eq. 'ATOM') .or. (readline(1:6) .eq. 'HETATM')) then
            read (readline(7:11),*)  iatom
            read (readline(13:16),'(A4)') ATOMNAMEmolec(i)
            read (readline(18:20),'(A3)') RESNmolec(i)
            read (readline(23:26),*) iresmolec(i)
            read (readline(31:38),*) xmolec(i)
            read (readline(39:46),*) ymolec(i)
            read (readline(47:54),*) zmolec(i)
            read (readline(55:60),*) occ
            read (readline(61:66),*) beta
            read (readline(73:76),'(A4)') SEGMmolec(i)
            read (readline(77:78),'(A2)') ELEMmolec(i)
            i = i + 1
        ENDiF
      enddo
10    close(imolinfile)
      iatomsmolec = i - 1               ! contains number of atoms in molecule read
      imolec = iresmolec(iatomsmolec)   ! number of molecules read
      print *, '-----------------------------------'
      print *, 'finished reading file ', molinfile
      print *, 'read molecule(s) = ', imolec
      print *, 'read     atom(s) = ', iatomsmolec
      print *, 'atoms/molecule   = ',iatomsmolec/imolec
      print *, '-----------------------------------'
!
!
!!!!!! JUST PRINTING THE READ GAS MOLECULE TO CHECK
!1011 format (A6,   I5,1X,A4,1X,A3,1X,1X,I4,1X,3X,F8.3,F8.3,F8.3,F6.2,F6.2,6X,A4,A2)
!     do i = 1, iatomsmolec
!         write (*,1011) 'ATOM  ',i,ATOMNAMEmolec(i),RESNmolec(i),iresmolec(i),xmolec(i)
&
!          ,ymolec(i),zmolec(i),occ,beta,SEGMmolec(i),ELEMmolec(i)
!     enddo
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!



!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Input STRUCTURE file
!!!
!!! reads file structureinfile (STRUCTURE_IN.pdb)
!!! stores:
!!!   ATOMNAMEstruct(i) - atom names (CA00,CA01,...)
!!!   RESNstruct(i)     - RESIDUE (GFX)
!!!   iresstruct(i)     - molecule number (should be just 1)
```

```
!!!   xstruct(i), ystruct(i), zstruct(i) - coordinates
!!!   occ              - occupation, should be 1.00
!!!   beta             - beta, should be 0.00
!!!   SEGMstruct(i)    - segment (GFX)
!!!   ELEMstruct(i)    - atom type (C)
!!! at the end of the read, should also provide
!!!   iatomsstruct = number of atoms in structure
!!!   istruct = number of molecules in structure
!!!
!!!
      structureinfile = 'STRUCTURE_IN.pdb'    ! file where structure is read from
      istructinfile = 21               ! unit where structure is read from
      open(unit=istructinfile,file=structureinfile,status='OLD',form='formatted', &
           access='sequential',action='read')
!!! now read input structure
      i = 1
      do j=1,nstrucmax
        read (istructinfile,'(A)',end=20) readline
        if ((readline(1:4) .eq. 'ATOM') .or. (readline(1:6) .eq. 'HETATM')) then
            read (readline(7:11),*)  iatom
            read (readline(13:16),'(A4)') ATOMNAMEstruct(i)
            read (readline(18:20),'(A3)') RESNstruct(i)
            read (readline(23:26),*) iresstruct(i)
            read (readline(31:38),*) xstruct(i)
            read (readline(39:46),*) ystruct(i)
            read (readline(47:54),*) zstruct(i)
            read (readline(55:60),*) occ
            read (readline(61:66),*) beta
            read (readline(73:76),'(A4)') SEGMstruct(i)
            read (readline(77:78),'(A2)') ELEMstruct(i)
            i = i + 1
        ENDiF
      enddo
20    close(istructinfile)
      iatomsstruct = i - 1               ! contains number of atoms in molecule read
      istruct = iresstruct(iatomsstruct)   ! number of molecules read
      print *, '-----------------------------------'
      print *, 'finished reading file ', structureinfile
      print *, 'read molecule(s) = ', istruct
      print *, 'read     atom(s) = ', iatomsstruct
      print *, '-----------------------------------'
!
!
!!!!!! JUST PRINTING THE READ STRUCTURE TO CHECK
!1021 format (A6,   I5,1X,A4,1X,A3,1X,1X,I4,1X,3X,F8.3,F8.3,F8.3,F6.2,F6.2,6X,A4,A2)
!     do i = 1, iatomsstruct
!         write (*,1021) 'ATOM
',i,ATOMNAMEstruct(i),RESNstruct(i),iresstruct(i),xstruct(i) &
!           ,ystruct(i),zstruct(i),occ,beta,SEGMstruct(i),ELEMstruct(i)
!     enddo
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!! Now generate Nmolec gas molecules at random locations (and orientations)
!!! while avoiding the structure and previous atoms (rij < RMIN ==> reject)
!!!
!!! Ngas = number of molecules to create

      print *, '****************************************'
      print *, 'enter the number of desired gas molecules:'
```

```
      read (*,*) Ngas
      print *, 'NOW WE ENTER THE PBC CHARACTERISTICS (Lx, Ly, zMIN, zMAX)...'
      print *, 'enter Lx (note: gas will go from -Lx/2 to +Lx/2):'
      read (*,*) Lx
      print *, 'enter Ly (note: gas will go from -Ly/2 to +Ly/2):'
      read (*,*) Ly
      print *, 'enter zMIN and zMAX separated by space (note: gas will go from zMIN to
zMAX:)'
      read (*,*) LzMIN,LzMAX
      print *, '****************************************'
      print *, 'GENERATING GAS...'
      print *, 'number of molecules = ', Ngas
      print *, 'number of atoms = ', Ngas*iatomsmolec/imolec
      print *, '****************************************'

      iatom = 0
      do j = 1, Ngas          ! number of molecules to generate
        !
!666     print *, 'I am on molecule ', j
666     deltax = (rand()-0.5)*Lx  ! random displacement for each molecule
        deltay = (rand()-0.5)*Ly
        deltaz = LzMIN + rand()*(LzMAX-LzMIN)
        !
        eulerpsi = 2*pi*rand()          ! random euler rotation angles
        eulertheta = acos(2*rand()-1)
        eulerphi = 2*pi*rand()
!!! euler rotation matrix (not used)
!       rotmatrix(1,1)=cos(eulerphi)*cos(eulerpsi)-
cos(eulertheta)*sin(eulerphi)*sin(eulerpsi)
!       rotmatrix(1,2)=-cos(eulertheta)*sin(eulerphi)*cos(eulerpsi)-
sin(eulerphi)*cos(eulerpsi)
!       rotmatrix(1,3)=sin(eulertheta)*sin(eulerphi)
!
rotmatrix(2,1)=sin(eulerphi)*cos(eulerpsi)+cos(eulertheta)*cos(eulerphi)*sin(eulerpsi)
!       rotmatrix(2,2)=cos(eulertheta)*cos(eulerphi)*cos(eulerpsi)-
sin(eulerphi)*sin(eulerpsi)
!       rotmatrix(2,3)=-sin(eulertheta)*cos(eulerphi)
!       rotmatrix(3,1)=sin(eulertheta)*sin(eulerpsi)
!       rotmatrix(3,2)=sin(eulertheta)*cos(eulerpsi)
!       rotmatrix(3,3)=cos(eulertheta)
!
        do i = 1, iatomsmolec/imolec      ! number of atoms/molecule
          iatom = iatom + 1       ! index, atom number
          ATOMNAMEgas(iatom) = ATOMNAMEmolec(i)
          RESNgas(iatom) = RESNmolec(i)
          SEGMgas(iatom) = SEGMmolec(i)
          ELEMgas(iatom) = ELEMmolec(i)
          iresstruct(iatom) = j + istruct
          !! rotating the molecule (all atoms around the origin)
          xtemp =  &
              xmolec(i)*(Cos(eulerphi)*Cos(eulerpsi)-
Cos(eulertheta)*Sin(eulerphi)*Sin(eulerpsi)) &
            + ymolec(i)*(-(Cos(eulerpsi)*Cos(eulertheta)*Sin(eulerphi))-
Cos(eulerphi)*Sin(eulerpsi)) &
                        + zmolec(i)*Sin(eulerphi)*Sin(eulertheta)
          ytemp =  &

xmolec(i)*(Cos(eulerpsi)*Sin(eulerphi)+Cos(eulerphi)*Cos(eulertheta)*Sin(eulerpsi)) &
             + ymolec(i)*(Cos(eulerphi)*Cos(eulerpsi)*Cos(eulertheta)-
Sin(eulerphi)*Sin(eulerpsi)) &
             - zmolec(i)*Cos(eulerphi)*Sin(eulertheta)
```

```
          ztemp =  &
              xmolec(i)*Sin(eulerpsi)*Sin(eulertheta) &
            + ymolec(i)*Cos(eulerpsi)*Sin(eulertheta) &
            + zmolec(i)*Cos(eulertheta)
          xgas(iatom) = xtemp+deltax
          ygas(iatom) = ytemp+deltay
          zgas(iatom) = ztemp+deltaz
        enddo
!! COLLISION DETECTION...
        icollision = 0    ! collision flag reset
        if (icollision .eq. 0) then
!! COLLISION DETECTION... WITH THE STRUCTURE
          iatom = iatom - iatomsmolec/imolec
          do inew = 1, iatomsmolec/imolec      ! number of atoms/molecule
              iatom = iatom + 1
              do jprev = 1, iatomsstruct
                !print *, iatom, jprev
                if (icollision .eq. 0) then
                  xij = abs(xgas(iatom)-xstruct(jprev)) - Lx*NINT(abs(xgas(iatom)-
xstruct(jprev))/Lx)
                  yij = abs(ygas(iatom)-ystruct(jprev)) - Ly*NINT(abs(ygas(iatom)-
ystruct(jprev))/Ly)
                  zij = abs(zgas(iatom)-zstruct(jprev)) -  &
                        (LzMAX-LzMIN)*NINT(abs(zgas(iatom)-zstruct(jprev))/(LzMAX-
LzMIN))
                  rij2 = xij**2 + yij**2 + zij**2
                  if (rij2 .le. 1*RMIN**2) then
                    icollision = 1                     ! collision has occurred
                    write (*,'(A32,I4,3X,A6,2I4)') 'collision of molecule/structure ',
&
                         j, 'atoms (g/s) ', iatom, jprev
                  ENDiF
                ENDiF
              enddo
          enddo
        ENDiF
        !
!! COLLISION DETECTION... WITH EARLIER MOLECULES/ATOMS
        if (icollision .eq. 0) then
          iatom = iatom - iatomsmolec/imolec
          do inew = 1, iatomsmolec/imolec      ! number of atoms/molecule
              iatom = iatom + 1
              do jprev = 1, iatom - inew
                if (icollision .eq. 0) then
                  xij = abs(xgas(iatom)-xgas(jprev)) - Lx*NINT(abs(xgas(iatom)-
xgas(jprev))/Lx)
                  yij = abs(ygas(iatom)-ygas(jprev)) - Ly*NINT(abs(ygas(iatom)-
ygas(jprev))/Ly)
                  zij = abs(zgas(iatom)-zgas(jprev)) -  &
                        (LzMAX-LzMIN)*NINT(abs(zgas(iatom)-zgas(jprev))/(LzMAX-LzMIN))
                  rij2 = xij**2 + yij**2 + zij**2
                  if (rij2 .le. 1*RMIN**2) then
                    icollision = 1                     ! collision has occurred
                    write (*,'(A23,2I4,3X,A6,2I4)') 'collision b/ molecules ',  &
                      j, jprev/(iatomsmolec/imolec)+1, 'atoms ', iatom, jprev
                  ENDiF
                endif
              enddo
          enddo
        ENDiF
        !
```

```
        !!! REJECT MOLECULE AND REDO IF COLLISION FOUND
        if (icollision .eq. 1) then   ! collision has occurred, recalculate the
molecule
          iatom = iatom - iatomsmolec/imolec
          goto 666
        endif
      enddo
!
!
!
      print *, 'DONE...'
      print *, 'created gas atoms = ', iatom
      print *, '****************************************'
!
!
      gasOUTfile = 'GAS.pdb'         ! file where GAS goes to
      igasfile = 31               ! unit
      open(unit=igasfile,file=gasOUTfile,status='unknown',form='formatted', &
            access='sequential',action='write')
!!!!!! WRITING THE GAS MOLECULES TO gasOUTfile
1031  format (A6,   I5,1X,A4,1X,A3,1X,1X,I4,1X,3X,F8.3,F8.3,F8.3,F6.2,F6.2,6X,A4,A2)
      do i = 1, iatom
         write (igasfile,1031) 'ATOM
',i+iatomsstruct,ATOMNAMEgas(i),RESNgas(i),iresstruct(i),xgas(i) &
          ,ygas(i),zgas(i),occ,beta,SEGMgas(i),ELEMgas(i)
      enddo
      write (igasfile,'(A3)') 'END'
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
      end
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

Bibliography

(1) Schaeperkoetter, J. C.; Connolly, M. J.; Buck, Z. N.; Taub, H.; Kaiser, H.; Wexler, C. Adsorption-Induced Expansion of Graphene Oxide Frameworks: Observation by in Situ Neutron Diffraction. *ACS Omega* **2019**, *4* (20), 18668–18676. https://doi.org/10.1021/acsomega.9b02589.

(2) Golebiowska, M.; Roth, M.; Firlej, L.; Kuchta, B.; Wexler, C. The Reversibility of the Adsorption of Methane–Methyl Mercaptan Mixtures in Nanoporous Carbon. *Carbon* **2012**, *50* (1), 225–234. https://doi.org/10.1016/j.carbon.2011.08.039.

(3) Collins, S. P.; Perim, E.; Daff, T. D.; Skaf, M. S.; Galvão, D. S.; Woo, T. K. Idealized Carbon-Based Materials Exhibiting Record Deliverable Capacities for Vehicular Methane Storage. *J. Phys. Chem. C* **2019**, *123* (2), 1050–1058. https://doi.org/10.1021/acs.jpcc.8b09447.

(4) Szczęśniak, B.; Choma, J.; Jaroniec, M. Gas Adsorption Properties of Graphene-Based Materials. *Adv. Colloid Interface Sci.* **2017**, *243*, 46–59. https://doi.org/10.1016/j.cis.2017.03.007.

(5) Dailly, A.; Beckner, M. Methane Storage on Metal-Organic Frameworks. In *Nanoporous Materials for Gas Storage*; Springer: Singapore, 2019; pp 227–253.

(6) Pazoki, H.; Anbia, M. Synthesis of a Microporous Copper Carboxylate Metal Organic Framework as a New High Capacity Methane Adsorbent. *Polyhedron* **2019**, *171*, 108–111. https://doi.org/10.1016/j.poly.2019.07.013.

(7) Feng, X.; Ding, X.; Jiang, D. Covalent Organic Frameworks. *Chem. Soc. Rev.* **2012**, *41* (18), 6010–6022. https://doi.org/10.1039/C2CS35157A.

(8) Geng, K.; He, T.; Liu, R.; Dalapati, S.; Tan, K. T.; Li, Z.; Tao, S.; Gong, Y.; Jiang, Q.; Jiang, D. Covalent Organic Frameworks: Design, Synthesis, and Functions. *Chem. Rev.* **2020**, *120* (16), 8814–8933. https://doi.org/10.1021/acs.chemrev.9b00550.

(9) Burress, J. W.; Gadipelli, S.; Ford, J.; Simmons, J. M.; Zhou, W.; Yildirim, T. Graphene Oxide Framework Materials: Theoretical Predictions and Experimental Results. *Angew. Chem. Int. Ed.* **2010**, *49* (47), 8902–8904. https://doi.org/10.1002/anie.201003328.

(10) Haque, E.; Yamauchi, Y.; Malgras, V.; Reddy, K. R.; Yi, J. W.; Hossain, Md. S. A.; Kim, J. Nanoarchitectured Graphene-Organic Frameworks (GOFs): Synthetic Strategies, Properties, and Applications. *Chem. – Asian J.* **2018**, *13* (23), 3561–3574. https://doi.org/10.1002/asia.201800984.

(11) Prosniewski, M.; Gillespie, A.; Knight, E.; Rash, T.; Stalla, D.; Romanos, J.; Smith, A. Evaluating Methane Adsorbed Film Densities on Activated Carbon in Dynamic Systems. *J. Energy Storage* **2018**, *20*, 357–363. https://doi.org/10.1016/j.est.2018.10.017.

(12) Knight, E. W.; Gillespie, A. K.; Prosniewski, M. J.; Stalla, D.; Dohnke, E.; Rash, T. A.; Pfeifer, P.; Wexler, C. Determination of the Enthalpy of Adsorption of Hydrogen in Activated Carbon at Room Temperature. *Int. J. Hydrog. Energy* **2020**, *45* (31), 15541–15552. https://doi.org/10.1016/j.ijhydene.2020.04.037.

(13) Bhatia, S. K.; Myers, A. L. Optimum Conditions for Adsorptive Storage. *Langmuir* **2006**, *22* (4), 1688–1700. https://doi.org/10.1021/la0523816.

(14) Kuchta, B.; Firlej, L.; Roszak, Sz.; P.Pfeifer; Wexler, C. Influence of Structural Heterogeneity of Nanoporous Sorbent Walls on Hydrogen Storage. *Seventh Int. Symp. Eff. Surf. Heterog. Adsorpt. Catal. Solids - ISSHAC-7* **2010**, *256* (17), 5270–5274. https://doi.org/10.1016/j.apsusc.2009.12.116.

(15) Gómez-Gualdrón, D. A.; Simon, C. M.; Lassman, W.; Chen, D.; Martin, R. L.; Haranczyk, M.; Farha, O. K.; Smit, B.; Snurr, R. Q. Impact of the Strength and Spatial Distribution of Adsorption Sites on Methane Deliverable Capacity in Nanoporous Materials. *ICAMD – Integrating Comput.-Aided Mol. Des. Prod. Process Des.* **2017**, *159*, 18–30. https://doi.org/10.1016/j.ces.2016.02.030.

(16) Rash, T. A.; Gillespie, A.; Holbrook, B. P.; Hiltzik, L. H.; Romanos, J.; Soo, Y. C.; Sweany, S.; Pfeifer, P. Microporous Carbon Monolith Synthesis and Production for Methane Storage. *Fuel* **2017**, *200*, 371–379. https://doi.org/10.1016/j.fuel.2017.03.037.

(17) Romanos, J.; Rash, T.; Abou Dargham, S.; Prosniewski, M.; Barakat, F.; Pfeifer, P. Cycling and Regeneration of Adsorbed Natural Gas in Microporous Materials. *Energy Fuels* **2017**, *31* (12), 14332–14337. https://doi.org/10.1021/acs.energyfuels.7b03119.

(18) Prosniewski, M. J.; Rash, T. A.; Knight, E. W.; Gillespie, A. K.; Stalla, D.; Schulz, C. J.; Pfeifer, P. Controlled Charge and Discharge of a 40-L Monolithic Adsorbed Natural Gas Tank. *Adsorption* **2018**, *24* (6), 541–550. https://doi.org/10.1007/s10450-018-9961-2.

(19) Romanos, J.; Abou Dargham, S.; Roukos, R.; Pfeifer, P. Properties of Adsorbed Supercritical Methane Film in Nanopores. *AIP Adv.* **2018**, *8* (12), 125011. https://doi.org/10.1063/1.5074086.

(20) Prosniewski, M.; Rash, T.; Romanos, J.; Gillespie, A.; Stalla, D.; Knight, E.; Smith, A.; Pfeifer, P. Effect of Cycling and Thermal Control on the Storage and Dynamics of a 40-L Monolithic Adsorbed Natural Gas Tank. *Fuel* **2019**, *244*, 447–453. https://doi.org/10.1016/j.fuel.2019.02.022.

(21) Rodriguez-Reinoso, F.; Silvestre-Albero, J. Nanoporous Materials for Gas Storage. In *Nanoporous Materials for Gas Storage*; Methane Storage on Nanoporous Carbons; Springer: Singapore, 2019; pp 209–226.

(22) Pfeifer, P.; Burress, J. W.; Wood, M. B.; Lapilli, C. M.; Barker, S. A.; Pobst, J. S.; Cepel, R. J.; Wexler, C.; Shah, P. S.; Gordon, M. J.; Suppes, G. J.; Buckley, S. P.; Radke, D. J.; Ilavsky, J.; Dillon, A. C.; Parilla, P. A.; Benham, M.; Roth, M. W. High-Surface-Area Biocarbons for Reversible On-Board Storage of Natural Gas and Hydrogen. *MRS Proc.* **2007**, *1041*, 1041-R02-02. https://doi.org/10.1557/PROC-1041-R02-02.

(23) *Adsorbed Natural Gas*; Ingevity.

(24) Ingevity Unveils Low-Pressure Adsorbed Natural Gas Fueling Station in Support of Its ANG Bi-Fuel Vehicle Technology. *Green Car Congress*. January 25, 2019.

(25) *National Overview: Facts and Figures on Materials, Wastes and Recycling*; 2020.

(26) *National Program 206: Manure and Byproduct Utilization Action Plan*; USDA, 2004; p 1.

(27) Cook, J.; Nuccitelli, D.; Green, S. A.; Richardson, M.; Winkler, B.; Painting, R.; Way, R.; Jacobs, P.; Skuce, A. Quantifying the Consensus on Anthropogenic Global Warming in the Scientific Literature. *Environ. Res. Lett.* **2013**, *8* (2), 024024. https://doi.org/10.1088/1748-9326/8/2/024024.

(28) Vijayagopal, R.; Rousseau, A. Electric Truck Economic Feasibility Analysis. *World Electr. Veh. J.* **2021**, *12* (2). https://doi.org/10.3390/wevj12020075.

(29) *Natural Gas Explained*; U.S. Energy Information Administration, https://www.eia.gov/energyexplained/natural-gas/.

(30) Lozano-Castelló, D.; Cazorla-Amorós, D.; Linares-Solano, A. Powdered Activated Carbons and Activated Carbon Fibers for Methane Storage: A Comparative Study. *Energy Fuels* **2002**, *16* (5), 1321–1328. https://doi.org/10.1021/ef020084s.

(31) Ortiz, L.; Kuchta, B.; Firlej, L.; Roth, M.; Wexler, C. Methane Adsorption in Nanoporous Carbon: The Numerical Estimation of Optimal Storage Conditions. *Mater. Res. Express* **2016**, *3*.

(32) Abdelaal, M. M.; Hegab, A. H. Combustion and Emission Characteristics of a Natural Gas-Fueled Diesel Engine with EGR. *IREC 2011 Int. Renew. Energy Congr.* **2012**, *64*, 301–312. https://doi.org/10.1016/j.enconman.2012.05.021.

(33) *Sulfur Oxides*; University Corporation for Atmospheric Research, https://scied.ucar.edu/learning-zone/air-quality/sulfur-oxides.

(34) *Basic Information about NO2*; Environmental Protection Agency, https://www.epa.gov/no2-pollution/basic-information-about-no2.

(35) Menon, V. C.; Komarneni, S. Porous Adsorbents for Vehicular Natural Gas Storage: A Review. *J. Porous Mater.* **1998**, *5* (1), 43–58. https://doi.org/10.1023/A:1009673830619.

(36) Golebiowska, M.; Roth, M.; Firlej, L.; Kuchta, B.; Wexler, C. The Reversibility of the Adsorption of Methane−methyl Mercaptan Mixtures in Nanoporous Carbon. *Carbon* **2012**, *50*, 225–334.

(37) Allen, M. R.; Frame, D. J.; Huntingford, C.; Jones, C. D.; Lowe, J. A.; Meinshausen, M.; Meinshausen, N. Warming Caused by Cumulative Carbon Emissions towards the Trillionth Tonne. *Nature* **2009**, *458* (7242), 1163–1166. https://doi.org/10.1038/nature08019.

(38) *Electricity*; U.S. Energy Information Administration, 2020; https://www.eia.gov/outlooks/aeo/pdf/AEO2020%20Electricity.pdf.

(39) *Petroleum and Other Liquids*; U.S. Energy Information Administration, 2020; https://www.eia.gov/outlooks/aeo/pdf/AEO2020%20Petroleum%20and%20Other%20Liquids.pdf.

(40) Archer, D. Methane Hydrate Stability and Anthropogenic Climate Change. *Biogeosciences* **2007**, *4* (4), 521–544. https://doi.org/10.5194/bg-4-521-2007.

(41) *Methane: The Other Important Greenhouse Gas*; Environmental Defense Fund.

(42) Butler, J. H.; Montzka, S. A. THE NOAA ANNUAL GREENHOUSE GAS INDEX, 2020.

(43) Archer, D. Greenhouse Gases. In *Global Warming: Understanding the Forecast*; Wiley, 2011; pp 35–39.

(44) Nunez; Pavley. *AB-32 Air Pollution: Greenhouse Gases: California Global Warming Solutions Act of 2006*; 2006.

(45) *Oregon House Bill 2186*; 2009.

(46) Fitzgibbon. *Washington HB 1091*; 2021.

(47) *California Low Carbon Fuel Standard Credit Price*; Neste, https://www.neste.com/investors/market-data/lcfs-credit-price.

(48) *Renewable Natural Gas*; EPA Landfill Methane Outreach Program, https://www.epa.gov/lmop/renewable-natural-gas.

(49) *Fleets Run Cleaner on Natural Gas*; Natural Gas Vehicles for America, 2018.

(50) Natural Gas Explained Natural Gas Pipelines.

(51) *Biogas Market Size, Share & Trends Analysis Report By Source (Municipal, Agricultural), By Application (Vehicle Fuel, Electricity), By Region (North America, Europe, APAC, CSA), And Segment Forecasts, 2021 - 2028*; Grand View Research; p https://www.grandviewresearch.com/industry-analysis/biogas-market.

(52) Mercier, G.; Klechikov, A.; Hedenström, M.; Johnels, D.; Baburin, I. A.; Seifert, G.; Mysyk, R.; Talyzin, A. V. Porous Graphene Oxide/Diboronic Acid Materials: Structure and Hydrogen Sorption. *J. Phys. Chem. C* **2015**, *119* (49), 27179–27191. https://doi.org/10.1021/acs.jpcc.5b06402.

(53) Hung, W.-S.; Tsou, C.-H.; De Guzman, M.; An, Q.-F.; Liu, Y.-L.; Zhang, Y.-M.; Hu, C.-C.; Lee, K.-R.; Lai, J.-Y. Cross-Linking with Diamine Monomers To Prepare Composite Graphene Oxide-Framework Membranes with Varying d-Spacing. *Chem. Mater.* **2014**, *26* (9), 2983–2990. https://doi.org/10.1021/cm5007873.

(54) Sun, J.; Morales-Lara, F.; Klechikov, A.; Talyzin, A. V.; Baburin, I. A.; Seifert, G.; Cardano, F.; Baldrighi, M.; Frasconi, M.; Giordani, S. Porous Graphite Oxide Pillared with Tetrapod-Shaped Molecules. *Carbon* **2017**, *120*, 145–156. https://doi.org/10.1016/j.carbon.2017.05.007.

(55) Srinivas, G.; Burress, J. W.; Ford, J.; Yildirim, T. Porous Graphene Oxide Frameworks: Synthesis and Gas Sorption Properties. *J. Mater. Chem.* **2011**, *21* (30), 11323–11329. https://doi.org/10.1039/C1JM11699A.

(56) Kumar, R.; Suresh, V. M.; Maji, T. K.; Rao, C. N. R. Porous Graphene Frameworks Pillared by Organic Linkers with Tunable Surface Area and Gas Storage Properties. *Chem. Commun.* **2014**, *50* (16), 2015–2017. https://doi.org/10.1039/C3CC46907G.

(57) Lombardi, T. N.; Schaeperkoetter, J. C.; Albesa, A.; Wexler, C. Adsorption-Induced Expansion of Graphene Oxide Frameworks with Covalently Bonded Benzene-1,4-Diboronic Acid: Numerical Studies. *ACS Omega* **2022**, *7* (14), 11980–11987. https://doi.org/10.1021/acsomega.2c00151.

(58) Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. CHARMM: A Program for Macromolecular Energy, Minimization, and Dynamics Calculations. *J. Comput. Chem.* **1983**, *4* (2), 187–217. https://doi.org/10.1002/jcc.540040211.

(59) Phillips, J. C.; Hardy, D. J.; Maia, J. D. C.; Stone, J. E.; Ribeiro, J. V.; Bernardi, R. C.; Buch, R.; Fiorin, G.; Hénin, J.; Jiang, W.; McGreevy, R.; Melo, M. C. R.; Radak, B. K.; Skeel, R. D.; Singharoy, A.; Wang, Y.; Roux, B.; Aksimentiev, A.; Luthey-Schulten, Z.; Kalé, L. V.; Schulten, K.; Chipot, C.; Tajkhorshid, E. Scalable

Molecular Dynamics on CPU and GPU Architectures with NAMD. *J. Chem. Phys.* **2020**, *153* (4), 044130. https://doi.org/10.1063/5.0014475.

(60) Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual Molecular Dynamics. *J. Mol. Graph.* **1996**, *14* (1), 33–38. https://doi.org/10.1016/0263-7855(96)00018-5.

(61) Lemmon, E. W.; McLinden, M. O.; Friend, D. G. Thermophysical Properties of Fluid Systems. In *NIST Chemistry WebBook, NIST Standard Reference Database Number 69*; National Institute of Standards and Technology: Gaithersburg MD, 20899.

(62) Valleroy, Z.; dos Santos, G.; Lombardi, T.; Wexler, C. Adsorption of Natural Gas Mixtures of Methane, Ethane, and Propane in Nanoporous Carbon: Fully Atomistic Numerical Studies. *Langmuir* **2020**, *36* (14), 3690–3702. https://doi.org/10.1021/acs.langmuir.9b03962.

(63) Andersen, H. C. Rattle: A "Velocity" Version of the Shake Algorithm for Molecular Dynamics Calculations. *J. Comput. Phys.* **1983**, *52* (1), 24–34. https://doi.org/10.1016/0021-9991(83)90014-1.

(64) Amaro, R.; Dhaliwal, B.; Luthey-Schulten, Z. University of Illinois at Urbana-Champaign Computational Biophysics Workshop. **2007**.

(65) Firlej, L.; Kuchta, B.; Roth, M. W.; Wexler, C. Molecular Simulations of Intermediate and Long Alkanes Adsorbed on Graphite: Tuning of Non-Bond Interactions. *J. Mol. Model.* **2011**, *17*, 811–816.

(66) MacKerell, A. D.; Bashford, D.; Bellott, M.; Dunbrack, R. L.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; Joseph-McCarthy, D.; Kuchnir, L.; Kuczera, K.; Lau, F. T. K.; Mattos, C.; Michnick, S.; Ngo, T.; Nguyen, D. T.;

Prodhom, B.; Reiher, W. E.; Roux, B.; Schlenkrich, M.; Smith, J. C.; Stote, R.; Straub, J.; Watanabe, M.; Wiórkiewicz-Kuczera, J.; Yin, D.; Karplus, M. All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins. *J. Phys. Chem. B* **1998**, *102* (18), 3586–3616. https://doi.org/10.1021/jp973084f.

(67) Ewald, P. P. Die Berechnung Optischer Und Elektrostatischer Gitterpotentiale. *Ann. Phys.* **1921**, *369* (3), 253–287. https://doi.org/10.1002/andp.19213690304.

(68) Becke, A. D. Density-functional Thermochemistry. III. The Role of Exact Exchange. *J. Chem. Phys.* **1993**, *98* (7), 5648–5652. https://doi.org/10.1063/1.464913.

(69) Lee, C.; Yang, W.; Parr, R. G. Development of the Colle-Salvetti Correlation-Energy Formula into a Functional of the Electron Density. *Phys. Rev. B* **1988**, *37* (2), 785–789. https://doi.org/10.1103/PhysRevB.37.785.

(70) Vosko, S. H.; Wilk, L.; Nusair, M. Accurate Spin-Dependent Electron Liquid Correlation Energies for Local Spin Density Calculations: A Critical Analysis. *Can. J. Phys.* **1980**, *58* (8), 1200–1211. https://doi.org/10.1139/p80-159.

(71) Stephens, P. J.; Devlin, F. J.; Chabalowski, C. F.; Frisch, M. J. Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields. *J. Phys. Chem.* **1994**, *98* (45), 11623–11627. https://doi.org/10.1021/j100096a001.

(72) Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; Nakatsuji, H.; Caricato, M.; Li, X.; Hratchian, H. P.; Izmaylov, A. F.; Bloino, J.; Zheng, G.; Sonnenberg, J. L.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.;

Montgomery, J. A., Jr.; Peralta, J. E.; Ogliaro, F.; Bearpark, M.; Heyd, J. J.; Brothers, E.; Kudin, K. N.; Staroverov, V. N.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Rega, N.; Millam, J. M.; Klene, M.; Knox, J. E.; Cross, J. B.; Bakken, V.; Adamo, C.; Jaramillo, J.; Gomperts, R.; Stratmann, R. E.; Yazyev, O.; Austin, A. J.; Cammi, R.; Pomelli, C.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Zakrzewski, V. G.; Voth, G. A.; Salvador, P.; Dannenberg, J. J.; Dapprich, S.; Daniels, A. D.; Farkas, Ö.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. *Gaussian~09 Revision E.01*.

(73) Vanommeslaeghe, K.; MacKerell, A. D. Automation of the CHARMM General Force Field (CGenFF) I: Bond Perception and Atom Typing. *J. Chem. Inf. Model.* **2012**, *52* (12), 3144–3154. https://doi.org/10.1021/ci300363c.

(74) Vanommeslaeghe, K.; Raman, E. P.; MacKerell, A. D. Automation of the CHARMM General Force Field (CGenFF) II: Assignment of Bonded Parameters and Partial Atomic Charges. *J. Chem. Inf. Model.* **2012**, *52* (12), 3155–3168. https://doi.org/10.1021/ci3003649.

(75) Stephens, P. J.; Devlin, F. J.; Chabalowski, C. F.; Frisch, M. J. Ab Initio Calculation of Vibrational Absorption and Circular Dichroism Spectra Using Density Functional Force Fields. *J. Phys. Chem.* **1994**, *98* (45), 11623–11627. https://doi.org/10.1021/j100096a001.

(76) Ditchfield, R.; Hehre, W. J.; Pople, J. A. Self-Consistent Molecular-Orbital Methods. IX. An Extended Gaussian-Type Basis for Molecular-Orbital Studies of Organic

Molecules. *J. Chem. Phys.* **1971**, *54* (2), 724–728. https://doi.org/10.1063/1.1674902.

(77) Nicolaï, A.; Zhu, P.; Sumpter, B. G.; Meunier, V. Molecular Dynamics Simulations of Graphene Oxide Frameworks. *J. Chem. Theory Comput.* **2013**, *9* (11), 4890–4900. https://doi.org/10.1021/ct4006097.

(78) Hyun, S.; Lee, J. H.; Jung, G. Y.; Kim, Y. K.; Kim, T. K.; Jeoung, S.; Kwak, S. K.; Moon, D.; Moon, H. R. Exploration of Gate-Opening and Breathing Phenomena in a Tailored Flexible Metal–Organic Framework. *Inorg. Chem.* **2016**, *55* (4), 1920–1925. https://doi.org/10.1021/acs.inorgchem.5b02874.

(79) Coudert, F.-X.; Mellot-Draznieks, C.; Fuchs, A. H.; Boutin, A. Prediction of Breathing and Gate-Opening Transitions Upon Binary Mixture Adsorption in Metal−Organic Frameworks. *J. Am. Chem. Soc.* **2009**, *131* (32), 11329–11331. https://doi.org/10.1021/ja904123f.

(80) Russell, B.; Villaroel, J.; Sapag, K.; Migone, A. D. O2 Adsorption on ZIF-8: Temperature Dependence of the Gate-Opening Transition. *J. Phys. Chem. C* **2014**, *118* (49), 28603–28608. https://doi.org/10.1021/jp508382v.

(81) Neimark, A. V.; Coudert, F.-X.; Boutin, A.; Fuchs, A. H. Stress-Based Model for the Breathing of Metal−Organic Frameworks. *J. Phys. Chem. Lett.* **2010**, *1* (1), 445–449. https://doi.org/10.1021/jz9003087.

(82) Gor, G. Yu.; Neimark, A. V. Adsorption-Induced Deformation of Mesoporous Solids. *Langmuir* **2010**, *26* (16), 13021–13027. https://doi.org/10.1021/la1019247.

(83) Ancilotto, F.; Cole, M. W.; Grosman, A.; Hernández, E. S.; Toigo, F. Expansion or Contraction of Slit Pores Due to Gas Uptake. *J. Low Temp. Phys.* **2011**, *163* (5), 284–301. https://doi.org/10.1007/s10909-011-0350-4.

(84) Lerf, A.; He, H.; Riedl, T.; Forster, M.; Klinowski, J. 13C and 1H MAS NMR Studies of Graphite Oxide and Its Chemically Modified Derivatives. *Int. Symp. React. Solids* **1997**, *101–103*, 857–862. https://doi.org/10.1016/S0167-2738(97)00319-6.

(85) He, H.; Riedl, T.; Lerf, A.; Klinowski, J. Solid-State NMR Studies of the Structure of Graphite Oxide. *J. Phys. Chem.* **1996**, *100* (51), 19954–19958. https://doi.org/10.1021/jp961563t.

(86) He, H.; Klinowski, J.; Forster, M.; Lerf, A. A New Structural Model for Graphite Oxide. *Chem. Phys. Lett.* **1998**, *287* (1), 53–56. https://doi.org/10.1016/S0009-2614(98)00144-4.

(87) Sircar, S.; Golden, T. C.; Rao, M. B. Activated Carbon for Gas Separation and Storage. *Carbon* **1996**, *34* (1), 1–12. https://doi.org/10.1016/0008-6223(95)00128-X.

(88) Hall, D. G. *Boronic Acids: Preparation, Applications in Organic Synthesis and Medicine*; John Wiley & Sons, 2006.

(89) Sana, M.; Leroy, G.; Wilante, C. Enthalpies of Formation and Bond Energies in Lithium, Beryllium, and Boron Derivatives. A Theoretical Attempt for Data Rationalization. *Organometallics* **1991**, *10* (1), 264–270. https://doi.org/10.1021/om00047a059.

(90) Malek, A.; Farooq, S. Kinetics of Hydrocarbon Adsorption on Activated Carbon and Silica Gel. *AIChE J.* **1997**, *43* (3), 761–776. https://doi.org/10.1002/aic.690430321.

(91) Reich R.; Ziegler W.T.; Rogers K.A. Adsorption of Methane, Ethane, and Ethlyene Gases and Their Binary and Ternary Mixture and Carbon Dioxide on Activated Carbon at 212-301 K and Pressure to 35 Atmospheres. **1980**.

(92) Walters, R. W.; Luthy, R. G. Equilibrium Adsorption of Polycyclic Aromatic Hydrocarbons from Water onto Activated Carbon. *Environ. Sci. Technol.* **1984**, *18* (6), 395–403.

(93) Agarwal, R. K.; Schwarz, J. A. Analysis of High Pressure Adsorption of Gases on Activated Carbon by Potential Theory. *Carbon* **1988**, *26* (6), 873–887. https://doi.org/10.1016/0008-6223(88)90111-X.

(94) He, Y.; Seaton, N. A. Monte Carlo Simulation and Pore-Size Distribution Analysis of the Isosteric Heat of Adsorption of Methane in Activated Carbon. *Langmuir* **2005**, *21* (18), 8297–8301. https://doi.org/10.1021/la050694v.

(95) Lawson, S.; Al-Naddaf, Q.; Newport, K.; Rownaghi, A.; Rezaei, F. Assessment of CO2/CH4 Separation Performance of 3D-Printed Carbon Monoliths in Pressure Swing Adsorption. *Ind. Eng. Chem. Res.* **2021**, *60* (45), 16445–16456. https://doi.org/10.1021/acs.iecr.1c01741.

(96) Kurig, H.; Russina, M.; Tallo, I.; Siebenbürger, M.; Romann, T.; Lust, E. The Suitability of Infinite Slit-Shaped Pore Model to Describe the Pores in Highly Porous Carbon Materials. *Carbon* **2016**, *100*, 617−624.

(97) Pfeifer, P.; Burress, J. W.; Wood, M. B.; Lapilli, C. M.; Barker, S. A.; Pobst, J. S.; Cepel, R. J.; Wexler, C.; Shah, P. S.; Gordon, M. J.; Suppes, G. J.; Buckley, S. P.; Radke, D. J.; Ilavsky, J.; Dillon, A. C.; Parilla, P. A.; Benham, M.; Roth, M. W. High-Surface-Area Biocarbon for Reversible on-Board Storage of Natural Gas and Hydrogen. *Carbon* **2016**, *100*, 617–624.

(98) Rubinstein, R. Y.; Kroese, D. P. *Simulation and the Monte Carlo Method*, 3rd ed.; Wiley, 2016.

(99) Martin, M. G.; Siepman, J. I. Transferable Potentials for Phase Equilibria. 1. United-Atom Description of n-Alkanes. *J Phys Chem B* **1998**, *102*, 2569–2577.

(100) Potoff, J. J.; Siepman, J. I. Vapor-Liquid Equilibria of Mixtures Containing Alkanes, Carbon Dioxide and Nitrogen. *AIChE J* **2001**, *47*, 1676–1682.

(101) Bretz, M.; Dash, J. G. Quasiclassical and Quantum Degenerate Helium Monolayers. *Phys. Rev. Lett.* **1971**, *26* (16), 963–965. https://doi.org/10.1103/PhysRevLett.26.963.

(102) Bretz, M.; Dash, J. G.; Hickernell, D. C.; McLean, E. O.; Vilches, O. E. Phases of He3 and He4 Monolayer Films Adsorbed on Basal-Plane Oriented Graphite. *Phys. Rev. A* **1973**, *8* (3), 1589–1615. https://doi.org/10.1103/PhysRevA.8.1589.

(103) Dash, J.; Ruvalds, J. *Phase Transitions in Surface Films*; Nato Science Series B; (Plenum Press: New York, NY, 1979.

(104) Greywall, D. S. Heat Capacity and the Commensurate-Incommensurate Transition of 4He Adsorbed on Graphite. *Phys. Rev. B* **1993**, *47* (1), 309–318. https://doi.org/10.1103/PhysRevB.47.309.

(105) Dash, J. G.; Schick, M.; Vilches, O. E. Phases of Helium Monolayers: Search and Discovery. *Surf. Sci.* **1994**, *299–300*, 405–414. https://doi.org/10.1016/0039-6028(94)90670-X.

(106) Gasparini, F. M.; Kimball, M. O.; Mooney, K. P.; Diaz-Avila, M. Finite-Size Scaling of 4He at the Superfluid Transition. *Rev. Mod. Phys.* **2008**, *80* (3), 1009–1059. https://doi.org/10.1103/RevModPhys.80.1009.

(107) Reatto, L.; Galli, D. E.; Nava, M.; Cole, M. W. Novel Behavior of Monolayer Quantum Gases on Graphene, Graphane and Fluorographene. *J. Phys. Condens. Matter* **2013**, *25* (44), 443001. https://doi.org/10.1088/0953-8984/25/44/443001.

(108) Makiuchi, T.; Tagai, M.; Nago, Y.; Takahashi, D.; Shirahama, K. Elastic Anomaly of Helium Films at a Quantum Phase Transition. *Phys. Rev. B* **2018**, *98* (23), 235104. https://doi.org/10.1103/PhysRevB.98.235104.

(109) Saunders, J.; Cowan, B.; Nyéki, J. Atomically Layered Helium Films at Ultralow Temperatures: Model Systems for Realizing Quantum Materials. *J. Low Temp. Phys.* **2020**, *201* (5), 615–633. https://doi.org/10.1007/s10909-020-02448-9.

(110) Saunders, J. Realizing Quantum Materials with Helium: Helium Films at Ultralow Temperatures, from Strongly Correlated Atomically Layered Films to Topological Superfluidity. In *Topological Phase Transitions and New Developments*; World Scientific: Hackensack, New Jersey, 2018; p 165.

(111) Thomy, A.; Duval, X. Adsorption de Molécules Simples Sur Graphite. *J Chim Phys* **1969**, *66*, 1966–1973. https://doi.org/10.1051/jcp/196966s21966.

(112) Crowell, P. A.; Reppy, J. D. Superfluidity and Film Structure in 4He Adsorbed on Graphite. *Phys. Rev. B* **1996**, *53* (5), 2701–2718. https://doi.org/10.1103/PhysRevB.53.2701.

(113) Nyéki, J.; Ray, R.; Sheshin, G.; Maidanov, V.; Mikheev, V.; Cowan, B.; Saunders, J. Structure and Superfluidity of 4He Films on Plated Graphite. *Low Temp. Phys.* **1997**, *23* (5), 379–388. https://doi.org/10.1063/1.593382.

(114) Whitlock, P. A.; Chester, G. V.; Krishnamachari, B. Monte Carlo Simulation of a Helium Film on Graphite. *Phys. Rev. B* **1998**, *58* (13), 8704–8715. https://doi.org/10.1103/PhysRevB.58.8704.

(115) Pierce, M.; Manousakis, E. Path-Integral Monte Carlo Simulation of the Second Layer of 4He Adsorbed on Graphite. *Phys. Rev. B* **1999**, *59* (5), 3802–3814. https://doi.org/10.1103/PhysRevB.59.3802.

(116) Corboz, P.; Boninsegni, M.; Pollet, L.; Troyer, M. Phase Diagram of 4He Adsorbed on Graphite. *Phys. Rev. B* **2008**, *78* (24), 245414. https://doi.org/10.1103/PhysRevB.78.245414.

(117) Castro Neto, A. H.; Guinea, F.; Peres, N. M. R.; Novoselov, K. S.; Geim, A. K. The Electronic Properties of Graphene. *Rev. Mod. Phys.* **2009**, *81* (1), 109–162. https://doi.org/10.1103/RevModPhys.81.109.

(118) Nichols, N. S.; Del Maestro, A.; Wexler, C.; Kotov, V. N. Adsorption by Design: Tuning Atom-Graphene van Der Waals Interactions via Mechanical Strain. *Phys. Rev. B* **2016**, *93* (20), 205412. https://doi.org/10.1103/PhysRevB.93.205412.

(119) Gordillo, M. C.; Boronat, J. 4He on a Single Graphene Sheet. *Phys. Rev. Lett.* **2009**, *102* (8), 085303. https://doi.org/10.1103/PhysRevLett.102.085303.

(120) Happacher, J.; Corboz, P.; Boninsegni, M.; Pollet, L. Phase Diagram of 4He on Graphene. *Phys. Rev. B* **2013**, *87* (9), 094514. https://doi.org/10.1103/PhysRevB.87.094514.

(121) Novoselov K. S.; Mishchenko A.; Carvalho A.; Castro Neto A. H. 2D Materials and van Der Waals Heterostructures. *Science* **2016**, *353* (6298), aac9439. https://doi.org/10.1126/science.aac9439.

(122) Gordillo, M. C.; Cazorla, C.; Boronat, J. Supersolidity in Quantum Films Adsorbed on Graphene and Graphite. *Phys. Rev. B* **2011**, *83* (12), 121406. https://doi.org/10.1103/PhysRevB.83.121406.

(123) Bruch, L. W.; Cole, M. W.; Kim, H.-Y. Transitions of Gases Physisorbed on Graphene. *J. Phys. Condens. Matter* **2010**, *22* (30), 304001. https://doi.org/10.1088/0953-8984/22/30/304001.

(124) Vranješ Markić, L.; Stipanović, P.; Bešlić, I.; Zillich, R. E. 4He Clusters Adsorbed on Graphene. *Phys. Rev. B* **2013**, *88* (12), 125416. https://doi.org/10.1103/PhysRevB.88.125416.

(125) Yu, J.; Lauricella, E.; Elsayed, M.; Shepherd, K.; Nichols, N. S.; Lombardi, T.; Kim, S. W.; Wexler, C.; Vanegas, J. M.; Lakoba, T.; Kotov, V. N.; Del Maestro, A. Two-Dimensional Bose-Hubbard Model for Helium on Graphene. *Phys. Rev. B* **2021**, *103* (23), 235414. https://doi.org/10.1103/PhysRevB.103.235414.

(126) Fradkin, E. Chapter 2. In *Field Theories of Condensed Matte Physics*; Cambrdige University Press, 2013.

(127) Chen, B. Y.; Mahanti, S. D.; Yussouff, M. Helium Atoms in Zeolite Cages: Novel Mott-Hubbard and Bose-Hubbard Systems. *Phys. Rev. Lett.* **1995**, *75* (3), 473–476. https://doi.org/10.1103/PhysRevLett.75.473.

(128) Fisher, M. P. A.; Weichman, P. B.; Grinstein, G.; Fisher, D. S. Boson Localization and the Superfluid-Insulator Transition. *Phys. Rev. B* **1989**, *40* (1), 546–570. https://doi.org/10.1103/PhysRevB.40.546.

(129) Cramer, C. *Essentials of Computational Chemistry: Theories and Models*, 2nd ed.; John Wiley & Sons: New York, NY, 2004.

(130) Aziz, R. A.; Nain, V. P. S.; Carley, J. S.; Taylor, W. L.; McConville, G. T. An Accurate Intermolecular Potential for Helium. *J. Chem. Phys.* **1979**, *70* (9), 4330–4342. https://doi.org/10.1063/1.438007.

(131) Anderson, J. B. An Exact Quantum Monte Carlo Calculation of the Helium–Helium Intermolecular Potential. II. *J. Chem. Phys.* **2001**, *115* (10), 4546–4548. https://doi.org/10.1063/1.1390512.

(132) Burganova, R.; Lysogorskiy, Y.; Nedopekin, O.; Tayurskii, D. Adsorption of Helium Atoms on Two-Dimensional Substrates. *J. Low Temp. Phys.* **2016**, *185*, 392–398. https://doi.org/10.1007/s10909-016-1473-4.

(133) Ambrosetti, A.; Silvestrelli, P. L. Adsorption of Rare-Gas Atoms and Water on Graphite and Graphene by van Der Waals-Corrected Density Functional Theory. *J. Phys. Chem. C* **2011**, *115* (9), 3695–3702. https://doi.org/10.1021/jp110669p.

(134) Hehre, W. J.; Stewart, R. F.; Pople, J. A. Self-Consistent Molecular-Orbital Methods. I. Use of Gaussian Expansions of Slater-Type Atomic Orbitals. *J. Chem. Phys.* **1969**, *51* (6), 2657–2664. https://doi.org/10.1063/1.1672392.

(135) Pritchard, B. P.; Altarawy, D.; Didier, B.; Gibson, T. D.; Windus, T. L. New Basis Set Exchange: An Open, Up-to-Date Resource for the Molecular Sciences Community. *J. Chem. Inf. Model.* **2019**, *59* (11), 4814–4820. https://doi.org/10.1021/acs.jcim.9b00725.

(136) Gheeraert, N.; Chester, S.; May, M.; Eggert, S.; Pelster, A. Mean-Field Theory for Extended Bose-Hubbard Model with Hard-Core Bosons. In *Selforganization in Complex Systems: The Past, Present, and Future of Synergetics*; Springer: Switzerland, 2016.

(137) Murthy, G.; Arovas, D.; Auerbach, A. Superfluids and Supersolids on Frustrated Two-Dimensional Lattices. *Phys. Rev. B* **1997**, *55* (5), 3104–3121. https://doi.org/10.1103/PhysRevB.55.3104.

(138) Wessel, S.; Troyer, M. Supersolid Hard-Core Bosons on the Triangular Lattice. *Phys. Rev. Lett.* **2005**, *95* (12), 127205. https://doi.org/10.1103/PhysRevLett.95.127205.

(139) Gan, J.-Y.; Wen, Y.-C.; Yu, Y. Supersolidity and Phase Diagram of Soft-Core Bosons on a Triangular Lattice. *Phys. Rev. B* **2007**, *75* (9), 094501. https://doi.org/10.1103/PhysRevB.75.094501.

(140) Zhang, X.-F.; Dillenschneider, R.; Yu, Y.; Eggert, S. Supersolid Phase Transitions for Hard-Core Bosons on a Triangular Lattice. *Phys. Rev. B* **2011**, *84* (17), 174515. https://doi.org/10.1103/PhysRevB.84.174515.

(141) Steele, W. A. The Physical Interaction of Gases with Crystalline Solids: I. Gas-Solid Energies and Properties of Isolated Adsorbed Atoms. *Surf. Sci.* **1973**, *36* (1), 317–352. https://doi.org/10.1016/0039-6028(73)90264-1.

(142) Steele, W. A. The Interaction of Rare Gas Atoms with Graphitized Carbon Black. *J. Phys. Chem.* **1978**, *82* (7), 817–821. https://doi.org/10.1021/j100496a011.

(143) Holstein, B. R. Semiclassical Treatment of the Periodic Potential. *Am. J. Phys.* **1988**, *56* (10), 894–898. https://doi.org/10.1119/1.15405.

(144) Lifshitz, E.; Pitaevskii, L. P. In *Statistcal Physics*; Pergamon Press: Oxford, 1980; p Part 2.

VITA

Todd Lombardi was born on June 17th, 1994, in St. Louis, Missouri. He received a BS in Applied Physics with a Computer Science focus from St. Mary's University in San Antonio, Texas in 2017. He received a PhD in Physics from the University of Missouri in 2022 under the advisement of Carlos Wexler.