

January 2015

Computational Investigations of Potential Energy Function Development for Metal-Organic Framework Simulations, Metal Carbenes, and Chemical Warfare Agents

Christian R. Cioce

University of South Florida, ccioce@mail.usf.edu

Follow this and additional works at: <http://scholarcommons.usf.edu/etd>



Part of the [Atomic, Molecular and Optical Physics Commons](#), and the [Physical Chemistry Commons](#)

Scholar Commons Citation

Cioce, Christian R., "Computational Investigations of Potential Energy Function Development for Metal-Organic Framework Simulations, Metal Carbenes, and Chemical Warfare Agents" (2015). *Graduate Theses and Dissertations*.
<http://scholarcommons.usf.edu/etd/5666>

This Dissertation is brought to you for free and open access by the Graduate School at Scholar Commons. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Scholar Commons. For more information, please contact scholarcommons@usf.edu.

Computational Investigations of Potential Energy Function Development for Metal–Organic
Framework Simulations, Metal Carbenes, and Chemical Warfare Agents

by

Christian R. Cioce

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Chemistry
College of Arts and Sciences
University of South Florida

Major Professor: Brian Space, Ph.D.
H. Lee Woodcock, Ph.D.
Randy Larsen, Ph.D.
Arjan van der Vaart, Ph.D.
Jonathan L. Belof, Ph.D.

Date of Approval:
July 6, 2015

Keywords: air purification, gas sorption, materials engineering, molecular simulation, monte carlo, nanomaterials

Copyright © 2015, Christian R. Cioce

Dedication

To my loving family and friends, for their endless support.

Acknowledgments

I owe everything to my family: Victor, Joy, and Victoria Cioce, who have continually supported my decisions and fueled my motivation. I have journeyed this far only because of your sacrifices and encouragement. You have taught me so much, which I aspire to pass on someday. Love is unbound by distance, and I have received it unconditionally from my family. To my caring grandmother, Mary, who has taught me to love and appreciate all that I have, thank you. And in honor of family remembered, I complete this chapter of my life with confidence that it would make you proud.

To my friends, both old and new, your continued support and sustained friendship encourages me to be a better person. As we have experienced all that we have together, I would like to acknowledge the fact that you all have had a significant impact on my past, my present, and most certainly my future. Jason Konopolsky, David Koch, Marc Epstein, Greg Parker, J.C. Flint, Jason Deck, Tarah Word, Aleksandra Karolak, and Justin White, you all have made a notable impact on my life and I cherish our friendship.

It cannot go unsaid that my fiancée, Arántzazu Gárate, has been my greatest supporter as of late. I have never admired someone so much in my entire life until I met you, Arántza. I respect your desire to help others, and I have learned so much from you. I will continue to uplift you and co-pilot you along the best paths, and will proudly walk beside you through this adventure of ours. Here's to many more "remember when"s, and a lifetime of happiness! Te amo, mi reina.

I am grateful to my Ph.D. committee of Brian Space, Lee Woodcock, Randy Larsen, Arjan van der Vaart, Jonathan Belof and Chair Babu Joseph for their guidance over the years. Jon, in addition to being a phenomenal mentor, you are truly a great friend. In everything from science to weight training, you dedicate 100% of your efforts to being the best that you possibly can be, and that is a respectable quality.

My present labmates Katherine Forrest, Tony Pham, Brant Tudor, Adam Hogan and Doug Franz – your friendship and collaboration will be missed.

A special thanks to an exceptional friend: Dr. Anthony Green. Tony, you and Lindsay have done for me what nobody else in this world has, and I am forever grateful for that. I have truly learned to appreciate all that I have thanks to the support that you both have given me. Thank you.

Lastly, I am infinitely grateful to Brian Space, my research advisor, my mentor and my friend. I sincerely appreciate all that you have done for me. I undoubtedly would not be where I am today, with all of my experiences and knowledge, without your support. You have opened the doors to a world of possibilities, and I am fully prepared for an outstanding career due to your mentorship. I cannot thank you enough for all that you have done, both professionally and especially personally, over the years. Thank you!

Table of Contents

| | |
|--|-----|
| List of Tables | iii |
| List of Figures | iv |
| Abstract | vii |
| Chapter 1: Introduction | 1 |
| Chapter 2: A Polarizable and Transferable PHAST N ₂ Potential for use in Materials Simulation | 7 |
| 2.1 Note to Reader | 7 |
| 2.2 Abstract | 7 |
| 2.3 Introduction | 8 |
| 2.4 Methods | 10 |
| 2.4.1 Born-Oppenheimer Potential Surface | 10 |
| 2.4.2 Many-Body Polarization | 12 |
| 2.4.3 Potential Energy Function | 14 |
| 2.5 Model Validation | 19 |
| 2.5.1 Bulk Pressure-Density Isotherms | 20 |
| 2.5.2 Solid Phase | 21 |
| 2.5.3 Trimer Energetics | 26 |
| 2.6 Conclusions | 27 |
| 2.7 Acknowledgments | 27 |
| Chapter 3: A PHAST Potential for Heterogeneous Simulation of Methane with Many-Body Interactions | 28 |
| 3.1 Introduction | 28 |
| 3.2 Methods | 29 |
| 3.2.1 Born-Oppenheimer Surface | 29 |
| 3.2.2 Many-Body Polarization | 30 |
| 3.2.3 Potential Energy Function | 31 |
| 3.3 Model Validation | 34 |
| 3.4 Limitations and Alternative Potentials | 35 |
| 3.5 Conclusions and Future Directions | 36 |

| | |
|---|----------|
| Chapter 4: Tuning Potentials for Heterogeneous Environments | 40 |
| 4.1 Introduction | 40 |
| 4.2 Methods | 41 |
| 4.2.1 Potential Energy Function | 41 |
| 4.2.2 Mixing Rules | 45 |
| 4.3 Results and Discussion | 45 |
| 4.4 Conclusions and Future Directions | 49 |
| Chapter 5: Characterization of Tunable Radical Metal-Carbenes: Key Interme- | |
| diates in Catalytic Cyclopropanation | 50 |
| 5.1 Note to Reader | 50 |
| 5.2 Abstract | 50 |
| 5.3 Introduction | 50 |
| 5.4 Methods | 56 |
| 5.5 Results and Discussion | 57 |
| 5.6 Conclusions | 65 |
| 5.7 Acknowledgments | 68 |
| 5.8 Supporting Information | 68 |
| Chapter 6: Virtually Engineering Dual Detection & Decontamination Materials | |
| for Chemical Warfare Agents: Computational Insights | 69 |
| 6.1 Introduction | 69 |
| 6.2 Methods | 72 |
| 6.2.1 Minimization and Complex Formation | 72 |
| 6.2.2 Sensor Engineering | 73 |
| 6.3 Conclusions and Future Directions | 76 |
| 6.4 Acknowledgments | 76 |
| Chapter 7: Conclusions | 77 |
| References | 80 |
| Appendix A: solidN2.py | 103 |
| Appendix B: genSorptionTraj.py | 130 |
| Appendix C: Copyright Permissions | 141 |
| About The Author | End Page |

List of Tables

| | | |
|-----------|---|----|
| Table 2.1 | Potential parameters (site position (R) and Lennard-Jones epsilon (ε) & sigma (σ)) of the polar (N ₂ -PHAST*) and non-polar (N ₂ -PHAST) models | 16 |
| Table 2.2 | Calculated cohesive energies for the α -N ₂ and γ -N ₂ crystals via the N ₂ -PHAST* potential | 24 |
| Table 3.1 | CH ₄ potential parameters | 34 |
| Table 4.1 | Potential parameters for the four helium clusters | 45 |
| Table 4.2 | The sigma values which map best to the CCSD(T) surface for the four helium clusters | 49 |
| Table 5.1 | Electrostatic Potential (ESP) and Natural Bond Orbital (NBO) results for the [Co(Por)]-carbene complexes | 61 |
| Table 5.2 | Charge decomposition analysis (CDA) for the [Co(Por)]-CHCO ₂ Et, Tungsten (W) Fischer carbene and Titanium (Ti) Schrock carbene complexes | 64 |
| Table 6.1 | Calculated binding energies of the chemical warfare agent sensor/target complexes | 76 |
| Table 6.2 | Calculated binding energies of sarin and DIMP with the modified ZnCS1 sensor | 79 |

List of Figures

| | | |
|------------|--|----|
| Figure 2.1 | The reference Born-Oppenheimer potential surface of the N ₂ dimer, calculated at the CCSD(T) level of theory, along with the N ₂ -PHAST* potential | 11 |
| Figure 2.2 | A diagrammatic five-site N ₂ monomer, indicating the positions (values in Å) of the three different sites: N2C, N2A and N2F | 14 |
| Figure 2.3 | Bulk N ₂ pressure-density isotherms at 77 K, 150 K and 298.15 K | 19 |
| Figure 2.4 | Calculated cohesive energies for the α -N ₂ and γ -N ₂ crystals via the N ₂ -PHAST* potential | 23 |
| Figure 2.5 | Validation of the N ₂ PHAST potential against an ensemble of 100 trimers | 26 |
| Figure 3.1 | The eight representative CH ₄ dimer orientations | 30 |
| Figure 3.2 | The reference Born-Oppenheimer potential surface of the CH ₄ dimer, calculated at the CCSD(T) level of theory, along with the Me-PHAST and Me-PHAST* potentials | 31 |
| Figure 3.3 | A site map of the nine-site methane monomer, indicating the positions of the three different site types: MeC, MeH and MeF | 33 |
| Figure 3.4 | Bulk methane pressure-density isotherm at 150 K | 36 |
| Figure 3.5 | Bulk methane pressure-density isotherm at 298 K | 37 |
| Figure 3.6 | The distribution of the Lennard-Jones ϵ parameter for the MeC and MeF sites across 200 fitting attempts | 38 |
| Figure 3.7 | The distribution of C ₆ for the MeC and MeF sites across 200 fitting attempts | 39 |
| Figure 4.1 | The Born-Oppenheimer N ₂ dimer surface along with the existing PHAST* potential and the newly developed potential of the Tang-Toennies type | 43 |
| Figure 4.2 | A schematic of the proposed “interfacial” fitting with helium atoms arranged in a linear fashion | 44 |

| | | |
|------------|---|----|
| Figure 4.3 | The C or “Cage” geometry of He atoms along with the resulting adsorption energy profile with a He atom | 47 |
| Figure 4.4 | The P or “Pocket” geometry of He atoms along with the resulting adsorption energy profile with a He atom | 48 |
| Figure 4.5 | The R or “Ring” geometry of He atoms along with the resulting adsorption energy profile with a He atom | 48 |
| Figure 4.6 | The W or “Wall” geometry of He atoms along with the resulting adsorption energy profile with a He atom | 49 |
| Figure 5.1 | Cyclopropanation reaction performed by a cobalt(II)-porphyrin catalyst. The diazoacetate reagent reacts with the porphyrin to form the carbene intermediate studied in this work | 53 |
| Figure 5.2 | Illustration of Fischer, Schrock, and [Co(Por)] orbitals | 55 |
| Figure 5.3 | Perspective depiction of [Co(Por)]-carbene complexes 2 & 3 | 59 |
| Figure 5.4 | Depiction of [Co(Por)]-carbene complexes 8 & 9 | 62 |
| Figure 5.5 | Plot of the positive spin density of the [Co(Por)]-CHCO ₂ Eт complex. The excess spin density is almost entirely localized on the carbene carbon in a nearly pure <i>p</i> atomic orbital in nature | 63 |
| Figure 5.6 | Molecular orbital diagram of the cobalt-porphyrin in complex with the carbene ligand -CHCO ₂ Eт (complex 1) | 66 |
| Figure 5.7 | Depiction of two low energy conformers of the [Co(Por)]-carbene complex 3 | 68 |
| Figure 6.1 | Structures of the nerve agent sarin, alongside its simulant, diisopropyl methylphosphonate (DIMP) and two additional chemical warfare agents, bis(2-chloroethyl) sulfide (mustard gas) and VX | 73 |
| Figure 6.2 | Rendered structures of the [Zn(Por)]-based sensors investigated: ZnTPP, ZnMBDPP, ZnDBDPP, and ZnCS1 | 74 |
| Figure 6.3 | The <i>in silico</i> chemical modifications made to both types of functionalized phenyl substituents on the ZnCS1 complex | 77 |
| Figure 6.4 | A functionalized porphyrin that has been found to tightly bind the nerve agent sarin selectively over the simulant DIMP, thus demonstrating the validity of a computational approach enabling specific binding of chemical agents | 78 |

| | | |
|------------|---|-----|
| Figure 6.5 | The newly crowded space at the distal position of the Zn center of the modified ZnCS1 sensor does not allow for interaction between the phosphoryl oxygen of DIMP to interact with the Zn center, thus resulting in a sensor which exhibits specificity for sarin over DIMP | 78 |
| Figure C.1 | Permission from the American Chemical Society to reuse the full article previously published in <i>J. Chem. Theory Comput.</i> 2013 , <i>9</i> (12), 5550–5557. in author’s dissertation | 147 |
| Figure C.1 | Permission from the American Chemical Society to reuse the full article previously published in <i>Organometallics</i> 2011 , <i>30</i> (10), 2739–2746. in author’s dissertation | 148 |

Abstract

Metal–Organic Frameworks (MOFs) are three-dimensional porous nanomaterials with a variety of applications, including catalysis, gas storage and separation, and sustainable energy. Their potential as air filtration systems is of interest for designer carbon capture materials. The chemical constituents (i.e. organic ligands) can be functionalized to create rationally designed CO₂ sequestration platforms, for example. Hardware and software alike at the bleeding edge of supercomputing are utilized for designing first principles-based molecular models for the simulation of gas sorption in these frameworks. The classical potentials developed herein are named PHAST – Potentials with High Accuracy, Speed, and Transferability, and thus are designed via a “bottom-up” approach. Specifically, models for N₂ and CH₄ are constructed and presented.

Extensive verification and validation leads to insights and range of applicability. Through this experience, the PHAST models are improved upon further to be more applicable in heterogeneous environments. Given this, the models are applied to reproducing high level ab initio energies for gas sorption trajectories of helium atoms in a variety of rare-gas clusters, the geometries of which being representative of sorption-like environments commonly encountered in a porous nanomaterial. This work seeks to push forward the state of classical and first principles materials modeling.

Additionally, the characterization of a new type of tunable radical metal–carbene is presented. Here, a cobalt(II)–porphyrin complex, [Co(Por)], was investigated to understand its role as an effective catalyst in stereoselective cyclopropanation of a diazoacetate reagent. Density functional theory along with natural bond order analysis and charge decomposition analysis gave insight into the electronics of the catalytic intermediate. The bonding pattern unveiled a new class of radical metal–carbene complex, with a doublet cobalt into which a triplet carbene σ donates, and subsequent back-bonding occurs into a π^* antibonding orbital. This is a different type of

interaction not seen in the three existing classes of metal–carbene complexes, namely Fischer, Schrock, and Grubbs.

Finally, the virtual engineering of enhanced chemical warfare agent (CWA) detection systems is discussed. As part of a U.S. Department of Defense supported research project, *in silico* chemical modifications to a previously synthesized zinc–porphyrin, ZnCS1, were made to attempt to achieve preferential binding of the nerve agent sarin versus its simulant, DIMP (diisopropyl methylphosphonate). Upon modification, a combination of steric effects and induced hydrogen bonding allowed for the selective binding of sarin. The success of this work demonstrates the role that high performance computing can play in national security research, without the associated costs and high security required for experimentation.

Chapter 1

Introduction

Nanomaterials have received an increasing amount of attention over the past several years, as they serve a variety of purposes. Metal-Organic Frameworks (MOFs) are one such type of porous material exhibiting tunability that is useful for gas separation and storage. The focus of the first three chapters of this dissertation is on designing potential energy functions for the accurate simulation of environmentally relevant gas sorption in MOFs. Such small molecules sorbates of interest are nitrogen, methane, hydrogen, carbon dioxide, among others. These potentials include electrostatic, repulsion and dispersion, and many-body polarization contributions to the total energy, and are designed to be transferable to exogenous systems through parameter mixing.

The form of the potential energy function is as follows:

$$U = U_{pol} + U_{es} + U_{rd} \quad (1.1)$$

where U_{pol} is the many-body polarization energy, U_{es} is the electrostatic energy, and U_{rd} is the repulsion/dispersion energy, modeled via the Lennard-Jones 12-6 function,

$$U_{ij}(r) = 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r} \right)^{12} - \left(\frac{\sigma_{ij}}{r} \right)^6 \right]. \quad (1.2)$$

The family of potentials developed are collectively referred to as PHAST (Potentials with High Accuracy, Speed and Transferability) and are parameterized from first principles calculations and

thus do not rely on experimental data for construction. To this end, they have been demonstrated to predict sorption isotherms for mixed gas systems of, e.g., CO₂ and N₂.

The implementation of a few different sets of mixing rules is performed to investigate an optimal set, and to verify those that are widely used throughout the literature. Specifically, the well-known and widely used mixing rule scheme of Lorentz-Berthelot (LB),

$$\begin{aligned}\varepsilon_{ij} &= (\varepsilon_{ii}\varepsilon_{jj})^{1/2} \\ \sigma_{ij} &= \frac{1}{2}(\sigma_{ii} + \sigma_{jj})\end{aligned}\quad (1.3)$$

is used throughout the potential development process. This mixing scheme estimates the well depth ε_{ij} for a mixed pair of unlike sites (ij , where $i \neq j$) as a geometric mean, and the characteristic length σ_{ij} as an arithmetic mean. A slightly more complicated mixing rule is that of Waldman-Hagler (WH),

$$\begin{aligned}\varepsilon_{ij} &= (\varepsilon_{ii}\varepsilon_{jj})^{1/2} \left(\frac{\sigma_{ii}^3\sigma_{jj}^3}{\sigma_{ij}^6} \right) \\ \sigma_{ij} &= \left(\frac{\sigma_{ii}^6 + \sigma_{jj}^6}{2} \right)^{1/6}\end{aligned}\quad (1.4)$$

Lastly, the set of similar, yet more sophisticated, mixing rules of Schmidt are investigated:

$$\begin{aligned}\varepsilon_{ij} &= \frac{(\varepsilon_{ii} + \varepsilon_{jj})\varepsilon_{ii}\varepsilon_{jj}}{\varepsilon_{ii}^2 + \varepsilon_{jj}^2} \\ \sigma_{ij} &= \frac{1}{2}(\sigma_{ii} + \sigma_{jj})\end{aligned}\quad (1.5)$$

Concerning the chapter specifics, Chapter 2 details the procedure for designing a PHAST potential energy function for molecular nitrogen, N₂. In total, four potential parameter sets are developed (polar and nonpolar, with LB and WH mixing rules) and vigorously tested to repro-

duce bulk thermodynamic properties, mainly pressure-density isotherms, but also for systems of thermal trimers. The potentials were tested across a wide pressure and temperature range, across the vapor, supercritical, liquid and solid phases. A Python script was written to simulate crystalline nitrogen, which constructs a supercell and maps the energy profile as a function of the lattice parameters (i.e. the crystal under compression and tension) of the alpha and gamma phases. For reference, the source code of this script is included as an Appendix. The resulting work from this project was published in the American Chemical Society journal *Journal of Chemical Theory and Computation*.

Chapter 3 contains much of the same procedural potential energy function development as in Chapter 2, but instead applied to methane, CH₄. Methane is a fossil fuel that has received a large amount of attention in the past several years due to the global climatic changes occurring. It is essential to efficiently separate methane from other environmentally relevant gases (N₂, CO₂, H₂O, H₂S) as well as toxic industrial chemicals, TICs (which themselves must be contained, though this is discussed in further detail in Chapter 6). The Me-PHAST model developed contains nine sites to represent a CH₄ monomer, and a total of eight representative dimer orientations were used to construct the Born-Oppenheimer surface. The potential provided in this chapter allows for immediate usage in heterogeneous simulations. In developing the model, it was discovered that several possible parameter sets exist which reproduce experimental bulk thermodynamic data. Viable solutions to circumvent this, e.g. the evolution of the PHAST potential, are presented.

Greater details of said alternative potentials are provided in Chapter 4. Here, a look into both the existing (PHAST, i.e. Lennard-Jones) and newly adopted (Tang-Toennies) potentials are given. Four characteristic rare-gas clusters composed exclusively of helium atoms are presented, their geometries being representative of sorption-like environments commonly encountered within a MOF. The adsorption energy profile along a trajectory is simulated for an incident helium atom for each of the clusters, and the models exhaustively explored to reproduce the minimum resulting surface. With the new form of the potential,

$$U = U_{rep} + U_{disp} + U_{es} + U_{pol} \quad (1.6)$$

where U_{rep} is the energy of electronic repulsion and is given by

$$U_{rep} = 315.77 * \exp(-\varepsilon(r - \sigma)) \quad (1.7)$$

U_{disp} is the dispersion energy,

$$U_{disp} = -f_6(r, \varepsilon) * \frac{C_6}{r^6} - f_8(r, \varepsilon) * \frac{C_8}{r^8} - f_{10}(r, \varepsilon) * \frac{C_{10}}{r^{10}} \quad (1.8)$$

U_{es} is the electrostatic energy given by Coulomb's law, and U_{pol} is the many-body polarization energy given by

$$U_{pol} = -\frac{1}{2} \sum_i^N \vec{\mu}_i \cdot \vec{E}_i^{stat} \quad (1.9)$$

as more thoroughly defined in Chapters 2 and 3, only a single parameter (σ) requires optimization. Correlation between the resulting optimal σ (that is, the σ whose resulting surface returns the minimum sum of squared errors, SSE, as compared to the high-level ab initio surface) and cluster type is sought and found to be unique for each cluster. It is shown that the new potential is requisite for sufficiently accurate quantitative agreement of high-level ab initio energies at both short and long pair separation distances. Thus the more physically grounded “softer” potential containing exponential repulsion and dispersion is demonstrated to be essential for accurately reproducing high-level calculated energies in both heterogeneous and homogeneous rare-gas environments.

Chapter 5 breaks the trend of potential energy function development, and consists of computational investigations of a catalytic cobalt(II)–porphyrin complex, [Co(Por)]. Here, high performance computing aided in characterization of a new class of radical metal–carbene complex. The [Co(Por)] complex serves as an effective catalyst for stereoselective cyclopropanation due to its radical character. Density functional theory (DFT) along with natural bond order analysis (NBO) and charge decomposition analysis (CDA) has given insight into the electronics of this catalytic intermediate. Subsequent to geometry optimization of the [Co(Por)] complex, NBO analysis yielded chemically insightful molecular orbitals for the structure, in addition to the spin density at the nucleus, partial charges and bond orders. CDA was performed in order to gain insight into the nature of the bonding taking place between the metallo-porphyrin and carbene ligand.

There are three well-known classes of metal–carbene complexes: Fischer, Schrock, and Grubbs. Fischer carbenes typically include a low oxidation state metal, whereas Schrock carbenes are characterized by higher oxidation states. The bonding pattern in Fischer carbenes can be described as singlet carbenes bonding with singlet metals. This is typically thought of as σ donation to the metal center and π back-bonding from the metal to the carbene. In contrast, Schrock complexes can be depicted as an interaction between triplet carbenes and triplet metals. In this work, the elucidated bonding pattern of the doublet Co(II) interacting with a triplet carbene gives rise to a new class of metal–carbene complex. Several varying functional groups adjacent to the carbene α -carbon were studied, and it was found that the presence of a neighboring functional group containing a π^* antibonding orbital makes possible the stabilization of the radical character. In the absence of this antibonding orbital, radical migration into the porphyrin macrocycle was observed, rendering the [Co(Por)] complex ineffective as a catalyst in the stereoselective cyclopropanation reaction.

Lastly, Chapter 6 presents a United States Department of Defense sponsored research project (via the Defense Threat Reduction Agency and the Charles Stark Draper Laboratory) in which chemical warfare agent (CWA) sensors were virtually designed for the preferential binding of selected targets. *In silico* chemical modifications were made to synthetically available sensor

platforms which resulted in preferential binding of the nerve agent sarin versus its laboratory approved simulant, diisopropyl methylphosphonate (DIMP). The original unmodified sensor, ZnCS1, demonstrated a preference for binding with DIMP as compared to sarin, where the binding energy is calculated as

$$E_{Bind} = E_{Complex} - (E_{Sensor}^{r=\infty} + E_{Target}^{r=\infty}) \quad (1.10)$$

That is, the energy of binding is calculated as the difference between the final energy of the optimized complex and the sum of the individual monomers, geometry optimized at infinite separation. Upon strategic modification of the ZnCS1 complex, reversal of preferential binding (now in favor of sarin) was achieved. The success of this work demonstrates the role that high performance computing (HPC) can play in national security research, without the associated costs and high security required for experimentation.

That is, the energy of binding is calculated as the difference between the final energy of the optimized complex and the sum of the individual monomers, geometry optimized at infinite separation. Upon strategic modification of the ZnCS1 complex, reversal of preferential binding (now in favor of sarin) was achieved. The success of this work demonstrates the role that high performance computing (HPC) can play in national security research, without the associated costs and high security required for experimentation.

Chapter 2

A Polarizable and Transferable PHAST N₂ Potential for use in Materials Simulation

2.1 Note to Reader

This chapter contains content previously published in *J. Chem. Theory Comput.*, **2013**, *9* (12), 5550–5557, and has been reproduced within the guidelines provided by the American Chemical Society.

2.2 Abstract

A polarizable and transferable intermolecular potential energy function, PHAST (Potentials with High Accuracy, Speed and Transferability), has been developed from first principles for molecular nitrogen to be used in the modeling of heterogeneous processes such as materials sorption and separations. A five-site (van der Waals and point charge) anisotropic model, that includes many-body polarization, is proposed. It is parameterized to reproduce high level electronic structure calculations (CCSD(T) using Dunning-type basis sets extrapolated to the CBS limit) for a representative set of dimer potential energy curves. Thus it provides a relatively simple yet robust and broadly applicable representation of nitrogen. Two versions are developed, differing by the type of mixing rules applied to unlike Lennard-Jones potential sites. It is shown that the Waldman-Hagler mixing rules are more accurate than Lorentz-Berthelot. The resulting potentials are demonstrated to be effective in modeling neat nitrogen, but are designed to also be useful in modeling N₂ interactions in a large array of environments such as metal-organic frame-

works, zeolites and at interfaces. In such settings, capturing anisotropic forces and interactions with (open and coordinated) metals and charged/polar environments is essential. In developing the potential, it was found that adding a seemingly redundant dimer orientation, slip-parallel (**S**), improved the transferability of the potential energy surface (PES). Notably, one of the solid phases of nitrogen was not as accurately represented energetically without including **S** in the representative set. Liquid simulations, however, were unaffected and worked equally well for both potentials. This suggests that accounting for a wide variety of configurations is critical in designing a potential that is intended for use in heterogeneous environments where many orientations, including those not commonly explored in the bulk, are possible. Testing and validation of the potential is achieved via simulations of a thermal distribution of trimer geometries compared to analogous high level electronic structure calculations, and molecular simulations of bulk pressure-density isotherms across the vapor, supercritical and liquid phases. Crystal lattice parameters and energetics of the α -N₂ and γ -N₂ solid phases are also evaluated, and determined to be in good agreement with experiment. Thus the proposed potential is shown to be efficacious for gas, liquid and solid use, representing both disordered and ordered configurations.

2.3 Introduction

Dinitrogen, N₂, is a major component of the Jovian planets and their moons, such as Saturn and Titan, as well as Neptune and Triton. Here on Earth, N₂ is the most abundant constituent of air, comprising roughly 78% of it along with oxygen, water vapor and other trace gases. Thus, nitrogen plays a critical role in many energy, environmental and life processes. Given its ubiquity and relative inertness, separating it efficiently and completely from other molecular species, such as methane and CO₂, is a useful goal.^{1,2}

Metal-Organic Frameworks (MOFs) are promising platforms to achieve efficient and cost-effective separations of a variety of gases. There is particular urgency in applying MOFs to energy-related processes, including associated separations, given the need for increased energy production in an environmentally sustainable fashion. MOFs consist of metal ions coordinated to organic linkers;³ these materials can present large (chemically and physically) tunable surface

areas and topologies that result in specific and selective molecular interactions.^{4–7} As a result, MOFs have shown utility for a multitude of purposes, including molecular capture, storage, catalysis and separation.^{1,2,7,8}

Given their ability to preferentially sorb CO₂ over N₂^{7,9–11} or CH₄ over N₂, for example, even in humid environments, MOFs are realistic solutions to many of the green challenges faced today. Molecular simulations now play an essential role in helping to both predict and retrodict material properties and processes that are determined empirically.⁷ Simulation is capable of both designing and improving material/MOF functionality by elucidating essential sorbent–sorbate interactions.¹² In order for simulation to play a central role, accurate and transferable potential energy functions must be available – high fidelity descriptions of small molecule potential energy surfaces (PESs), when available, are usually expressed in complex forms unsuitable for simulations where unlike species interact.^{13,14} Building on our previous work describing MOF and sorbate (e.g., H₂, CO₂ and CH₄) force fields,^{7,15–21} a transferable and polarizable potential energy function for dinitrogen, N₂-PHAST* (**P**otentials with **H**igh **A**ccuracy, **S**peed and **T**ransferability), has been developed. It is parameterized to high level ab initio data (CCSD(T) using Dunning-type basis sets extrapolated to the CBS limit) and constructed for use in heterogeneous environments such as MOFs and planar interfaces, or more generally where polarization and specific anisotropic interactions cannot be ignored. Typical liquid state potential surfaces are fit to bulk properties, and can be highly effective in like environments or mixtures where the neat bulk like orientations are still prevalent.^{22,23} As a result, the region of applicability of such force fields reflects the typical orientations dominant in that phase. In contrast, molecular sorption at interfaces, both planar and sorbate/solid, tends to explore a wider range of configuration space and thus requires a potential that is effective for the entire range of molecular orientations and interactions associated with highly heterogeneous environments. Thus, the form of our potential is chosen to be compact and reasonably simple to implement yet sufficiently accurate, containing electrostatic, repulsion/dispersion and explicit induction contributions to the total energy. Note, constructing transferable potentials for such applications is an active area of investigation.²⁴ Murthy et al. developed a five charge site model for solid state simulation in 1983, namely 5q,

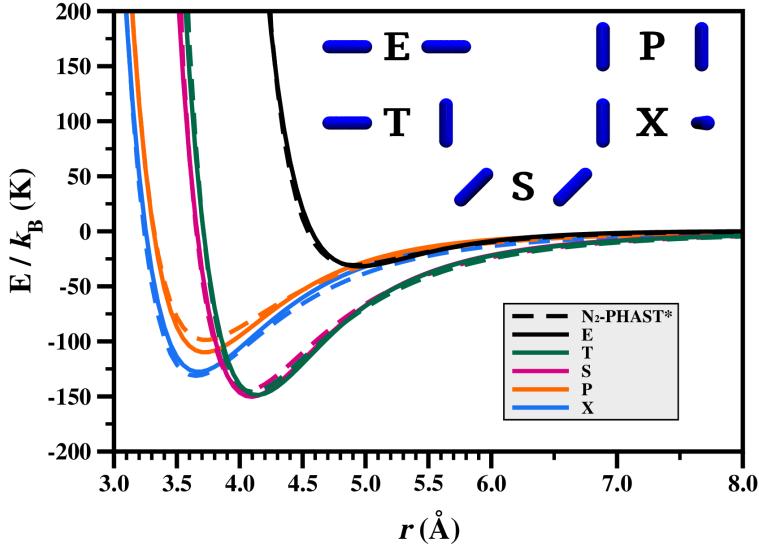
which contains electrostatic and repulsion/dispersion contributions to the total energy.²⁵ This potential is similar to the nonpolar variant of our model, N₂-PHAST, though it will be shown that our models outperform it.

The remainder of this article is outlined as follows: we next present the form of our potential in greater detail, offering a short review of the Thole-Applequist type polarization models, then discuss the parameterization process, and lastly transition to the validation of the potential.

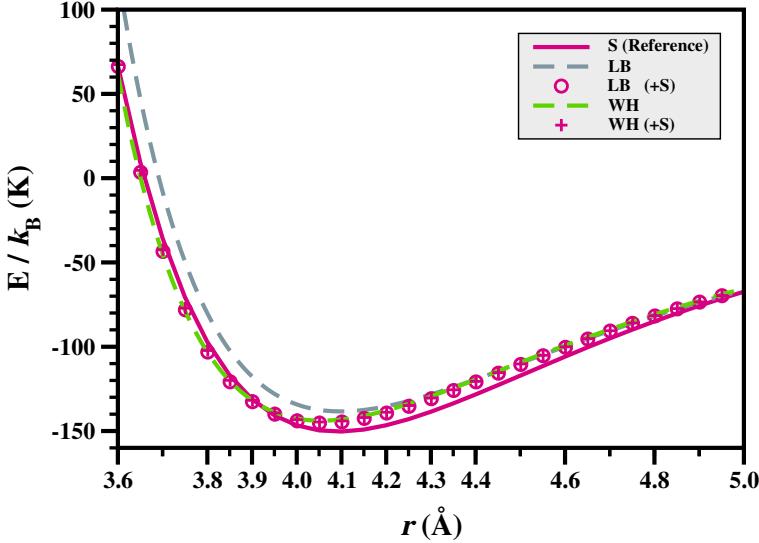
2.4 Methods

2.4.1 Born-Oppenheimer Potential Surface

The nitrogen molecule was approximated as rigid with a bond length of 1.098 Å, as determined by experiment.^{26,27} Note, it is simple to relax this constraint and include full flexibility,^{28,29} but this was not done here. To construct the Born-Oppenheimer potential surface for N₂-N₂, five dimer orientations were selected as representative of the geometries and energetics explored by nitrogen in the condensed phase, namely, **E** (end-to-end), **T** (T-configuration), **S** (slip-parallel), **P** (parallel) and **X** (X-configuration) (see Figure 2.1(a)).^{18,30,31} For each orientation, the pair energies were calculated for center-of-mass distances r apart along the z -axis. In the domain spanning $r = 3.00$ to 8.00 Å, energies were calculated every $\Delta 0.05$ Å. All ab initio calculations were performed using Molpro³² at the CCSD(T)³³ level of theory, unless otherwise stated. The employed basis functions were the augmented correlation-consistent sets of Dunning et al. (aug-cc-pVTZ/QZ).^{34,35} Basis set superposition errors were counterpoise-corrected via the method of Boys and Bernardi,³⁶ and the energy eigenvalues extrapolated to the complete basis set (CBS) limit in a standard fashion.³⁷ The PESs including (Figure 2.1(a)) and neglecting (not shown) the slip-parallel (**S**) orientation both nearly exactly reproduce the high accuracy CCSD(T) reference calculation. However, even though **S** is only a seemingly small orientational change from the already-included dimer configurations, it is not predicted as accurately unless explicitly included in the reference set – Figure 2.1(b) demonstrates this effect. It will be shown below that this does not change simulated liquid state data, but has a large effect on the solid state energetics where



(a)



(b)

Figure 2.1: (a) The reference Born-Oppenheimer potential surface of the N_2 dimer (solid lines), calculated at the CCSD(T) level of theory, along with the $\text{N}_2\text{-PHAST}^*$ potential (dashed lines). The illustrated polar potential makes use of the Waldman-Hagler (WH) mixing rules, however the Lorentz-Berthelot (LB) results are visually indistinguishable. Additionally, the five corresponding representative dimer orientations, labeled **E** (end-to-end), **T** (T-configuration), **S** (slip-parallel), **P** (parallel) and **X** (X-configuration), are shown. (b) The ab initio reference **S** potential curve from (a) (solid pink line) along with the calculated curves via the $\text{N}_2\text{-PHAST}^*$ model, both before (dashed lines) and after (pink symbols) the inclusion of this orientation in the dimer set. It can be seen that including this configuration results in a closer mapping to the reference, an effect independent of the mixing rule applied. It will be shown below that only the potentials that included the **S** configuration in their construction are highly transferable.

the **S** configuration is relevant. Thus including the **S** configuration appears to be essential in constructing a highly transferable potential. This suggests that it is critical for the representative set of orientations to be comprehensive in order to develop potentials for use in heterogeneous media, where parts of configuration space that are not important for neat liquid simulations may be explored, such as simulating sorption in porous media. Note, it is also possible to include a large set of random dimers in a representative PES set.³⁸

2.4.2 Many-Body Polarization

A highly transferable potential energy function for use in charged/polar heterogeneous environments requires inclusion of induction effects which typically demand a many-body treatment. Thus a point atomic polarizability model of the Thole-Applequist type was adopted. A brief description of such polarization models, as applied here, is presented first. A more thorough description can be found elsewhere.^{17,39–42} In such models, subjecting a collection of bodies with non-zero static point polarizabilities α (that after interaction become polarizability tensors on each site) to a static electric field E induces a dipole moment,

$$u_i^m = \alpha_i(E_i^m + E'_i) \quad (2.1)$$

$$u_i^m = \alpha_i(E_i^m - \sum_{j,n} T_{ij}^{mn} u_j^n) \quad (2.2)$$

where lower indices label the sites, upper indices label the vector/tensor components, T is the exponentially damped dipole field tensor, and E' is the induced electric field. With some manipulation, one finds the $3N \times 3N$ matrix equation,

$$\sum_{j,n} \left(\frac{1}{\alpha_i} \delta_{ij} \delta^{mn} + T_{ij}^{mn} \right) u_j^n = E_i^m \quad (2.3)$$

from which one may construct block-matrices,

$$\hat{\mathbf{A}} \equiv \begin{pmatrix} [\frac{I}{\alpha_0}] & [T_{0,1}] & \dots & [T_{0,N-1}] \\ [T_{1,0}] & [\frac{I}{\alpha_1}] & \dots & [T_{1,N-1}] \\ \dots & \dots & \dots & \dots \\ [T_{N-1,0}] & [T_{N-1,1}] & \dots & [\frac{I}{\alpha_{N-1}}] \end{pmatrix} \quad (2.4)$$

$$\mathbf{u} = \begin{pmatrix} [u_0] \\ [u_1] \\ \dots \\ [u_{N-1}] \end{pmatrix} \quad \mathbf{E} = \begin{pmatrix} [E_0] \\ [E_1] \\ \dots \\ [E_{N-1}] \end{pmatrix} \quad (2.5)$$

leading to

$$\hat{\mathbf{A}}\mathbf{u} = \mathbf{E} \quad (2.6)$$

which has solutions by either matrix inversion or iterative methods – the latter being used herein.

Finally, the polarization energy is given by

$$U_{pol} = -\frac{1}{2} \sum_{i,m} u_i^m E_i^m \quad (2.7)$$

Long range corrections are handled using an implementation of the Wolf field method, which we have recently shown to be efficient and effective.^{17,43} The polarizability model has the advantage that it can incorporate polarizable sites from any other species (e.g., material or other sorbent) that is similarly parameterized to gas phase data, without further assumptions. The ostensible disadvantage is the cost of evaluating the many-body function. However, this can be largely offset by iterating efficiently and porting these operations to graphics processing units, or GPUs.^{15,44}

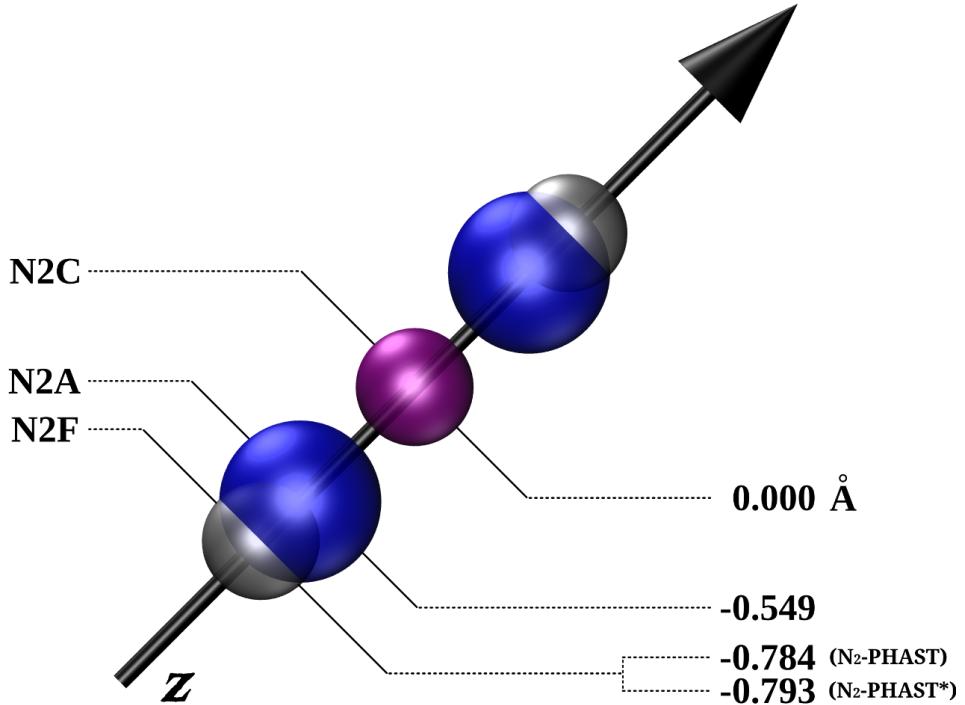


Figure 2.2: A diagrammatic five-site N_2 monomer, indicating the positions (values in \AA) of the three different sites: N2C (purple), N2A (blue) and N2F (gray). Note the only positional difference between the nonpolar (N₂-PHAST) and polar (N₂-PHAST*) model is observed in the N2F site, where the nonpolar site resides slightly closer to the molecule's center relative to the polar N2F site. The site locations illustrated above are those obtained when implementing the Waldman-Hagler (WH) mixing rules, though the analogous Lorentz-Berthelot (LB) positions differ minutely. All potential parameters are listed in Table 2.1.

2.4.3 Potential Energy Function

The nitrogen potential model is referred to as N₂-PHAST*, where N₂ signifies nitrogen, * denotes the inclusion of explicit polarization, and PHAST is a newly defined acronym which refers to the form of our growing group of broadly applicable potential energy functions.^{17,18,20,21} We additionally define a distinguishable model which neglects polarization effects, N₂-PHAST, for comparative purposes. Suitable for both molecular dynamics and Monte Carlo simulations, the PHAST potential has been designed with transferability in mind by explicitly including all essential potential energy interactions encountered in describing complex systems from the gas to solid phases. The form of our potential includes contributions to the many-body polarization, electrostatic and the electronic repulsion/dispersion energy, respectively, as

$$U = U_{pol} + U_{es} + U_{rd} \quad (2.8)$$

The form of U_{pol} is given by Equations 2.3 and 2.7. To evaluate the polarization energy, atomic point polarizabilities are required and were determined in an iterative manner. Given the ab initio polarizability tensor, calculated in NWChem⁴⁵ at the CCSD(T)/aug-cc-pVQZ level of theory,

$$\hat{\alpha}_{N_2} = \begin{pmatrix} 1.4982 & 0 & 0 \\ 0 & 1.4982 & 0 \\ 0 & 0 & 2.1640 \end{pmatrix} \text{\AA}^3 \quad (2.9)$$

one constructs a model tensor via the Thole polarization model in attempt to match the ab initio tensor. Initial values for point polarizabilities were chosen from a carefully constructed training set, developed by van Duijnen et al.⁴² Through refinement, the ab initio tensor, that itself is in agreement with experimental measurements of the trace,²⁷ was exactly reproduced. The atomic point polarizabilities which gave the best fit to $\hat{\alpha}_{N_2}$ are $\alpha_{N2A} = 0.4551 \text{ \AA}^3$ and $\alpha_{N2C} = 1.4965 \text{ \AA}^3$ (respectfully corresponding to the atom-centered and center-of-mass sites, as illustrated in Figure 3.3). Note that modeling without induction amounts to setting $U_{pol} = 0$, and when this is the case our model is referred to as N₂-PHAST (nonpolar), whereas N₂-PHAST* signifies the inclusion of polarization, as stated. Constructing a model neglecting explicit polarization (the N₂-PHAST model has some measure of two-body polarization implicit in the potential) is useful for widely-employed simulation codes and appropriate when strong charge interactions are less relevant. N₂-PHAST is also useful as a negative control in modeling phenomena such as sorption in polar media, in which case it can provide a metric for the degree of substrate polarization.

Table 2.1: Potential parameters (site position (\mathbf{R}) and Lennard-Jones epsilon (ε) & sigma (σ)) of the polar ($\text{N}_2\text{-PHAST}^*$) and nonpolar ($\text{N}_2\text{-PHAST}$) models, for both the Lorentz-Berthelot (LB) and Waldman-Hagler (WH) mixing rules. Common parameters (charge (\mathbf{Q}) and point polarizability (α°)) are globally shared, and therefore nonexclusive to either mixing rule schemes.

| | Model | Site | \mathbf{Q} (e) | α° (\AA^3) | \mathbf{R} (\AA) | ε/k_B (K) | σ (\AA) | \mathbf{R} (\AA) | ε/k_B (K) | σ (\AA) |
|-----------------------------|-------|--------|------------------|-----------------------------------|-------------------------------|-----------------------|---------------------------|-------------------------------|-----------------------|---------------------------|
| N₂-PHAST* | N2C | | 1.0474 | 1.4965 | 0.000 | 27.0222 | 3.4357 | 0.000 | 31.0329 | 3.4047 |
| | N2F | | 0.0000 | 0.0000 | ± 0.791 | 14.5517 | 3.0841 | ± 0.793 | 13.4780 | 3.0899 |
| | N2A | | -0.5237 | 0.4551 | ± 0.549 | 0.0000 | 0.0000 | ± 0.549 | 0.0000 | 0.0000 |
| N₂-PHAST | N2C | | 1.0474 | 0.0000 | 0.000 | 25.6443 | 3.4442 | 0.000 | 27.2601 | 3.4203 |
| | N2F | | 0.0000 | 0.0000 | ± 0.788 | 15.5320 | 3.0729 | ± 0.784 | 15.3094 | 3.0777 |
| | N2A | | -0.5237 | 0.0000 | ± 0.549 | 0.0000 | 0.0000 | ± 0.549 | 0.0000 | 0.0000 |
| | | Common | | | Lorentz-Berthelot | | | | | Waldman-Hagler |

Figure 3.3 provides a visual description of the five-site nitrogen model. Two sites, labeled N2A, are atom-centered. A third site, denoted N2C, is positioned at the molecule’s center. Both the N2A and N2C sites have associated charge and polarization parameters. Additionally, the N2C site contains a Lennard-Jones ε and σ value. The remaining two sites, named N2F, reside at the extremes of the molecule and are Lennard-Jones sites. Note, the only mass-containing sites are those of the N2A type.

Further deconstructing the potential energy function piece-wise, the electrostatic energy term U_{es} is parameterized to reproduce the quadrupole-quadrupole interactions between nitrogen molecules. The ab initio molecular electric quadrupole tensor was calculated at the CCSD(T)/aug-cc-pV5Z level of theory via the finite-field method⁴⁶ and found to be $\Theta = -1.514 \text{ D}\AA$, which is in very good agreement with experiment.^{27,47–50} A homogeneous field strength of 0.005 a.u. was found to be effective after careful refinement, and is consistent with values used by others.⁵¹ The expression for the quadrupole moment of an axially symmetric molecule aligned along the z -axis is:⁵²

$$\Theta \equiv \Theta_{zz} = \sum_i e_i z_i^2 \quad (2.10)$$

Establishment of appropriate partial charges for the N2A and N2C sites requires a consideration of the N_2 electrostatics. Given the calculated permanent quadrupole moment and the positions of the nitrogen nuclei ($z = \pm 0.549\text{\AA}$), rearrangement of Equation 2.10 gives atomic charges that reproduce the first non-vanishing multipole moment, by construction. To parameterize the potential form, the partial charges were allowed to float, through a simulated annealing process in parameter space, along with the other potential parameters in seeking an overall best mapping to the high level reference surface.¹⁸ Initial charge values reproduced the permanent quadrupole at the Hartree–Fock level of theory (which is also negative, with about 20% smaller magnitude), and the resulting charges reproduced the CCSD(T) quadrupole moment to within 0.2%. The charges on the nitrogen atoms are then $q_{\text{N2A, float}} = -0.5237 \text{ } e$, and the site at the

molecule's center is assigned a charge equal to $q_{N2C,\text{float}} = -2q_{N2A,\text{float}} = +1.0474 e$. Throughout our studies these charges have been used, therefore the "float" subscript notation shall be implied and omitted.

The electronic repulsion/dispersion contribution to the total energy, U_{rd} , is described via a Lennard-Jones 12-6 function. Pair mixing of non-identical sites is treated via two sets of rules: Lorentz-Berthelot (LB) and Waldman-Hagler (WH).⁵³ The well-known LB rules estimate the well depth ε_{ij} for a mixed pair of unlike sites (ij , where $i \neq j$) as a geometric mean, and the characteristic length σ_{ij} as an arithmetic mean:

$$\begin{aligned}\varepsilon_{ij} &= (\varepsilon_{ii}\varepsilon_{jj})^{1/2} \\ \sigma_{ij} &= \frac{1}{2}(\sigma_{ii} + \sigma_{jj})\end{aligned}\tag{2.11}$$

WH proposed mixing rules that are slightly more complicated:

$$\begin{aligned}\varepsilon_{ij} &= (\varepsilon_{ii}\varepsilon_{jj})^{1/2} \left(\frac{\sigma_{ii}^3\sigma_{jj}^3}{\sigma_{ij}^6} \right) \\ \sigma_{ij} &= \left(\frac{\sigma_{ii}^6 + \sigma_{jj}^6}{2} \right)^{1/6}\end{aligned}\tag{2.12}$$

LB mixing rules have the advantage that they are extensively tested and widely implemented. However, WH showed that their prescription was substantially more effective in reproducing mixed rare gas interactions.^{53,54} The WH rules, in contrast to other more advanced schemes,⁵⁵ retain simplicity and ease of implementation. WH was also chosen because it can be shown to be consistent with a perturbative expansion of the polarizability model (Equation 2.3) that leads to dispersion interactions in the zero field limit.¹⁷ Note, it is required to use mixing rules even in constructing a N₂ pair potential as there are unlike sites within the proposed model (see Figure 3.3).

Along with the partial charges and the N2F site's position, the Lennard-Jones potential parameters ε and σ were included in the multi-parameter stochastic mapping to the Born-

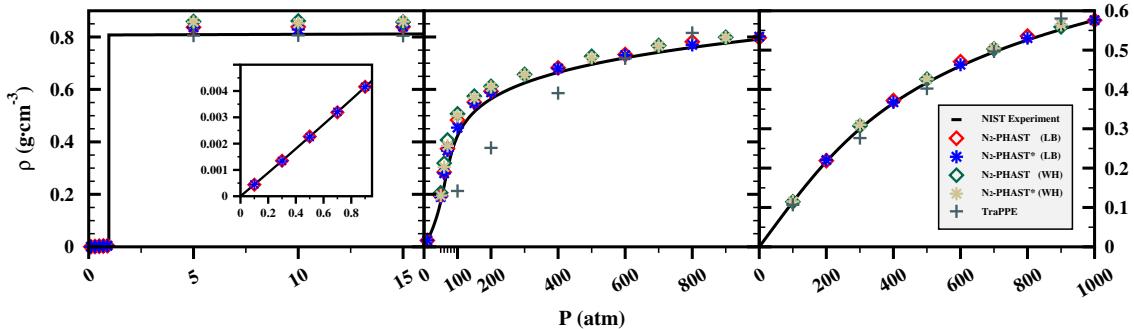


Figure 2.3: Pressure-density isotherms at 77 K (left), 150 K (center) and 298.15 K (right). The inset of the 77 K isotherm depicts the simulated low pressure points of the gas phase.

Oppenheimer surface (Figure 2.1(a)). Note, during this process the parameters associated with U_{pol} are held fixed at the values given above. Also note that the N2F site positions were constrained to the z -axis such that the COM displacements remain equivalent, i.e., their z -components are equal in magnitude but opposite in sign.

The process was performed a total of four times, thereby resulting in four parameter sets: each model (polar and nonpolar) was treated using both sets of mixing rules (LB and WH). The potential surface illustrated in Figure 2.1(a) was generated using the WH mixing rules. The analogous LB surface is visually indistinguishable, and therefore not depicted. Table 2.1 provides a complete listing of the potential parameters.

2.5 Model Validation

To validate our potential, several comparisons were performed. The following experimental measurements were chosen as benchmarks: pressure-density isotherms in the vapor, supercritical and liquid phases, and the energetics and lattice constants of crystalline α -N₂ and γ -N₂. Further, thermal trimer configurations were generated and the model energetics compared to nearly-exact high level electronic structure calculations (at the same level of theory used to generate the potential curves). Classical simulations were performed using an in-house code, MPMC (Massively Parallel Monte Carlo),⁵⁶ which is distributed under the GNU General Public License (v3) via Google Code.

2.5.1 Bulk Pressure-Density Isotherms

In developing a potential effective under a variety of conditions, from gas phase to highly anisotropic sorption environments, a first step is validating its neat, bulk behavior. Thermodynamic averages were collected, and pressure-density isotherms constructed, for bulk systems over a large region of phase space and compared to experiment. Simulations were performed using the isothermal-isobaric (NPT) ensemble, and all experimental data was obtained from the Thermophysical Properties of Fluid Systems database of the NIST Webbook.⁵⁷

At 77 K, simulations were performed at pressures up to 15 atm. The resulting isotherm, plotted against experiment, is shown in Figure 2.3. Note that a phase transition (vapor → liquid) is experimentally observed at 0.96 atm, and that our potential captures the densities well on either side. It is important to test our potential in this neighborhood of phase space not only to ensure that the potential accurately captures the transition and existence in the liquid phase, but also because this is representative of nitrogen loading in MOFs; liquid nitrogen is used to conveniently and inexpensively characterize pore volume/size at 77 K. Exploring deeper into the liquid phase, one additional simulation was performed at 100 K/100 atm (not shown). All models calculated densities to within 5% of experiment at this state point.

Isothermal pressure-density plots were also constructed at 150 K & 298.15 K, and are illustrated in Figure 2.3. For both temperatures, the potentials are validated up to 1000 atm and maintain excellent agreement with experiment throughout. Note the sigmoidal shape of the 150 K isotherm, indicative of a dense supercritical fluid ($T_C=126.192\text{ K}$, $P_C=33.514\text{ atm}$, $\rho_C=0.3133\text{ g}\cdot\text{cm}^{-3}$).⁵⁷

Included in the plots within Figure 2.3 are pressure-density data for the three-site nonpolar TraPPE²² model. TraPPE was designed for use in liquid simulations and thus it is no surprise that it performs very well at 77 K. Similarly, its prediction of densities at 298.15 K is relatively good. At 150 K, however, it is not as accurate. As exemplified in each plot within Figure 2.3, all models (LB/WH in combination with N₂-PHAST/PHAST*) performed equally well at all simulated temperatures and pressures.

2.5.2 Solid Phase

To test the transferability of our compact potential form, crystal lattice parameters and associated energetics were calculated for nitrogen in the alpha ($\alpha\text{-N}_2$) and gamma ($\gamma\text{-N}_2$) solid phases. Both phases have been examined theoretically^{25,58–61} and experimentally.^{62–68} Figure 2.4 depicts the cohesive energy^{59,62,63} as a function of volume, for the various models. Further, it demonstrates the importance of including the **S** configuration in the representative set of dimers.

$\alpha\text{-N}_2$

$\alpha\text{-N}_2$ is the low temperature and pressure phase, and thus the lowest energy configuration. Experimentally, it is characterized as having a primitive cubic crystal structure with space group $Pa\bar{3}$, lattice constant $a = 5.644 \text{ \AA}$ (at 4.2 K) and bond length equal to 1.055 \AA .^{62,64,65} In simulation, a crystal composed of 125 unit cells was used. The chosen N_2 bond length was 1.098 \AA (representative of the gas phase), and therefore slightly longer than the experimental value. The change in bond length associated with different environments can be captured by augmenting our model with flexibility, as was done in a similar polarizable potential.^{28,29}

The energy profile as a function of the lattice constant a was calculated for values ranging from $5.300 \rightarrow 5.650 \text{ \AA}$, in $\Delta 0.001 \text{ \AA}$ increments. When using the WH parameters, the N_2 -PHAST* and N_2 -PHAST models captured the minimum energy at $a = 5.488 \text{ \AA}$ and $a = 5.485 \text{ \AA}$, respectively; this compares favorably to within 2.8% of the experimental value. Alternatively, when the nonpolar LB parameters are used, the calculated lattice constant is slightly further from the experimental result. Comparing to the existing $5q$ potential of Murthy,²⁵ it is found that $5q$ more accurately predicts the lattice parameters for $\alpha\text{-N}_2$, being 2% different from experiment as opposed to our best 2.8% via the N_2 -PHAST* model. The equilibrium cohesive energy, however, is 1 kJ/mol greater than the minimum of our best model, as is shown in Figure 2.4(b). Table 2.2 provides a complete listing of the calculated and experimental lattice constants and volumes.

Figure 2.4(a) compares the model energetics for the two different mixing rules. The alpha phase is represented in green. Including the **S** orientation in the representative dimer set (solid lines with symbols) results in a more stable crystal versus its exclusion (symbols only). As

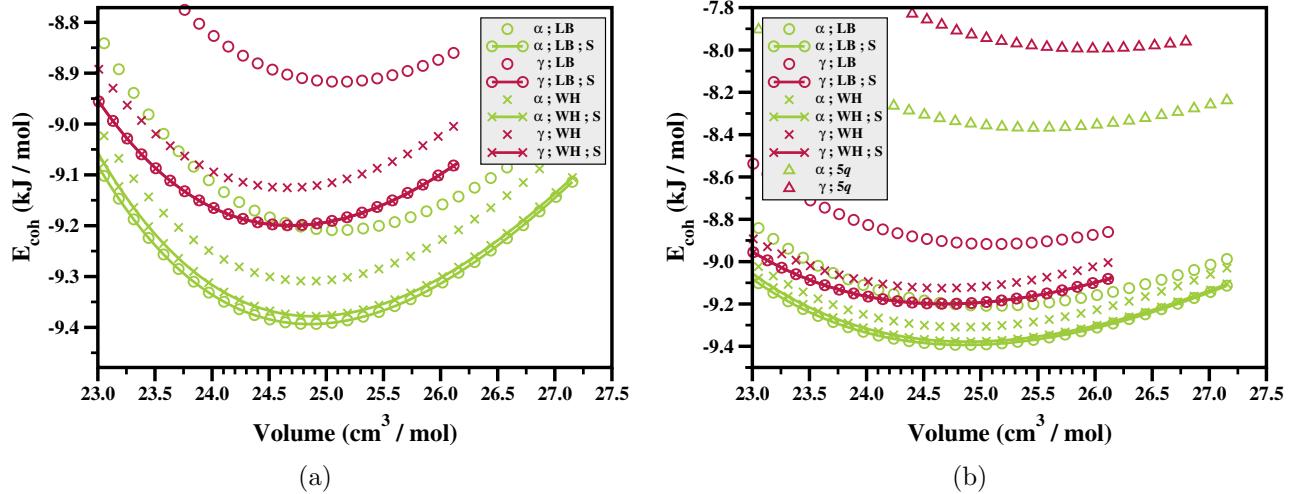


Figure 2.4: Calculated cohesive energies for the α -N₂ (green) and γ -N₂ (red) crystals via the N₂-PHAST* potential. **(a)** Comparing the two sets of mixing rules, LB and WH, a notable energy difference exists when an incomplete representative dimer set is used (symbols only). In contrast, both mixing rules trend towards the same energetics when the dimer set is more comprehensive (symbols with lines), i.e., by including the **S** configuration. **(b)** A wider-scaled version of (a), including the calculated energetics via the 5q potential of Murthy.²⁵

seen experimentally, the alpha phase is the most stable at zero temperature, and the magnitude of the cohesive energy is consistent with both theoretical (MP2)⁵⁹ and experimentally derived estimates.^{62,63} This is especially interesting given that the WH models, both including and neglecting the **S** configuration, give very similar dimer potential energy surfaces (even for **S**, see Figure 2.1(b)), nearly identical liquid phase results, but different solid energies. Further, the convergence of the two models, i.e. LB and WH, when including a larger configuration space, is encouraging and suggests a more robust representation of the true PES.

Table 2.2: Calculated α -N₂ and γ -N₂ lattice parameters for both the polar (N₂-PHAST*) and nonpolar (N₂-PHAST) models, as well as both the Lorentz-Berthelot (LB) and Waldman-Hagler (WH) mixing rules. Also included is the 5 q model of Murthy.²⁵ The experimental alpha phase lattice constant, a_{exp} , was reported by Wyckoff.⁶⁴ All lattice parameters are reported in units of Angstroms, and volumes in cm³/mol.

| | | α -N ₂ | γ -N ₂ |
|----|------------------------|--------------------------|--------------------------|
| LB | N ₂ -PHAST* | 5.486 | 5.644 |
| | N ₂ -PHAST | 5.483 | 5.644 |
| WH | N ₂ -PHAST* | 5.488 | 5.644 |
| | N ₂ -PHAST | 5.485 | 5.644 |
| | Murthy 5 q | 5.532 | 5.644 |
| | | 25.49 | 27.07 |
| | | | |
| | | a_{calc} | a_{exp} |
| | | Vol _{calc} | Vol _{exp} |
| | | | |
| | | a_{calc} | a_{exp} |
| | | Vol _{calc} | Vol _{exp} |

γ -N₂

The γ crystalline lattice is tetragonal with two molecules per unit cell, and is of the $P4_2/mnm$ space group. Determined by x-ray diffraction, the lattice constants are reported to be $a = 3.957 \text{ \AA}$ and $c = 5.109 \text{ \AA}$ for a crystal with an N₂ interatomic distance of 1.100 \AA .⁶⁷

By the same method described above for α -N₂, the energy profile of a γ -N₂ crystal as a function of volume was calculated, using all PHAST models, and the results tabulated in Table 2.2. Using the WH mixing rules, the calculated lattice parameters for the polar and nonpolar models, respectively, are $a = 3.990 \text{ \AA} / c = 5.152 \text{ \AA}$ and $a = 3.989 \text{ \AA} / c = 5.150 \text{ \AA}$. The analogous LB parameters are little changed, with N₂-PHAST* and N₂-PHAST predicting $a = 3.990 \text{ \AA} / c = 5.152 \text{ \AA}$, and $a = 3.987 \text{ \AA} / c = 5.148 \text{ \AA}$, respectively. In contrast to α -N₂, the 5q model of Murthy less accurately predicts the volume of the γ -N₂ crystal, being 7.47% different from experiment as opposed to our best 2.26% via the N₂-PHAST model with LB mixing. The minimum cohesive energy, again, is too repulsive and thus does not give an acceptable description of the energetics of either solid phase.

Similar to the trend observed in the α phase, but to a larger degree for γ , including the S orientation in mapping to the reference PES results in stabilization (red symbols and lines in Figure 2.4(a)). Note that in both crystalline phases, excluding the S configuration and modeling via the N₂-PHAST* potential using the WH mixing rules (Xs only, no lines) results in more accurate energetics compared to the LB rules. As above, it is shown that both mixing rules trend towards the same energetics as the representative dimer set becomes more comprehensive.

The present minimum, zero temperature, energy values for α and γ nitrogen are comparable within expected uncertainties from experiments and MP2 calculations. Exact comparison to experiment is difficult due to needed zero-point corrections and thermal effects that have significant errors compared to the small energy differences considered. The comparison with theoretical MP2 calculations is also limited by uncertainties associated with this level of perturbative analysis.^{59,63,69} For reference, an estimate of the zero-point corrected cohesive energy of α nitrogen was given as -8.3 kJ/mol;⁵⁹ this is reasonably close to our model calculation. Note, the calculations presented in Figure 2.4 are zero temperature and, in that sense, represent a

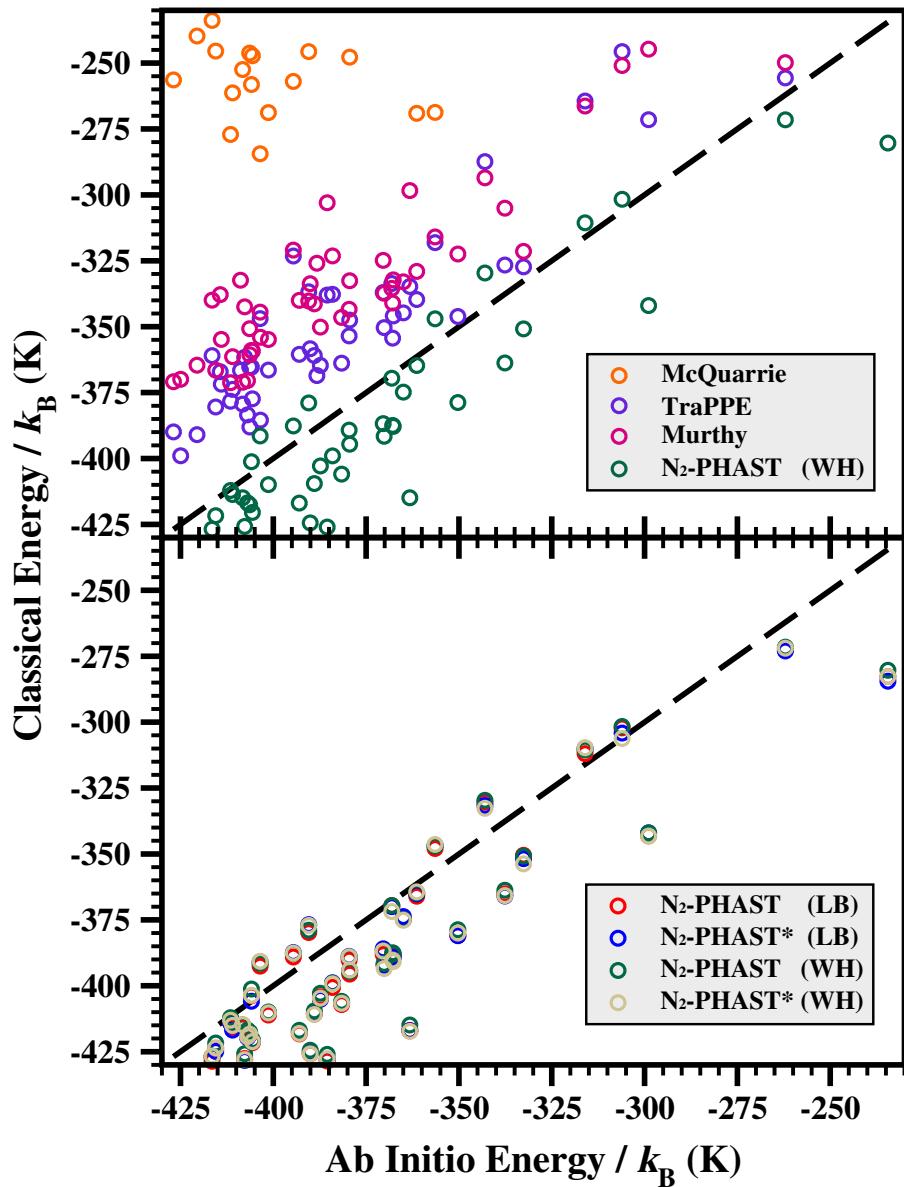


Figure 2.5: The classical potential energy function is compared to the ab initio values for an ensemble of 100 trimers (circles). Data points in perfect agreement would fall along the dashed line.

lower bound to the energy. Further, distinct calculations have found the energy difference between the α and γ phases to be comparably small.^{59,63} Again, note that the N₂-PHAST model bond length is taken as the gas phase value which differs from the observed crystalline value by nearly four percent. This can also have a small nonnegligible effect on the energetics.⁵⁹ The general agreement between our model solids and experimental data is also inspiring given the small magnitude of energy involved, which requires a high fidelity representation of the PES. The solid-solid transition line is not calculated in this work as the focus of this potential is for use in heterogeneous media, however might be of interest in future studies.

2.5.3 Trimer Energetics

A final test was performed, comparing thermal trimer configurations to nearly exact electronic structure calculations on the same configuration. This test involves (minimal) many-body contributions and explores a relatively wide region of configuration space, while still allowing rigorous detailed comparison. An ensemble of 100 three-molecule configurations were generated via Monte Carlo (using N₂-PHAST) at 15 K, effectively exploring the potential energy surface whilst avoiding dissociation. Single-point energy calculations were then performed for each configuration subject to each potential of interest: N₂-PHAST* and N₂-PHAST, with LB and WH mixing rules. Ab initio single-point calculations were then performed for each configuration at the CCSD(T) level of theory using aug-cc-pVTZ/QZ with CBS extrapolation.

Taking the CCSD(T) value as exact, the root-mean-square error in single-point energy over the ensemble was calculated. The results are plotted in Figure 2.5. For the two LB potentials, values of 6.84% and 6.81% were found for N₂-PHAST* and N₂-PHAST, respectively. For the WH potentials, values of 6.85% and 6.51% were found. In both cases, adding polarization to the potential was not found to significantly affect the accuracy of trimer energy calculations. All of the potentials from this work perform markedly better than comparable potentials from other works in the context of this test. A calculation using the 5 q potential of Murthy over the ensemble produced a rms error of 13.26%, the three-site nonpolar TraPPE²² potential produced an error of 9.41% and the single-site isotropic model by McQuarrie,⁷⁰ unsurprisingly, produced an

error of 56.59%. These results are represented by pink, purple and orange datasets, respectively, in Figure 2.5. It can be seen that the $5q$ potential gives a reasonably good result comparable to that of TraPPE, but with slightly more systematic error.

2.6 Conclusions

Using five high-level reference ab initio curves, a five-site anisotropic and transferable potential energy function for N₂ has been developed. Two potential models, one including (N₂-PHAST*) and one neglecting (N₂-PHAST) explicit induction effects, have been defined. Additionally, two differing sets of mixing rules (Lorentz-Berthelot and Waldman-Hagler) are used. The combination of each model with mixing rule results in four total parameter sets. All have been extensively tested and proven accurate for bulk systems across a wide region of phase space, encompassing the condensed, vapor and supercritical phases. Pressure-density isotherms were constructed at 77 K, 150 K and 298.15 K, and closely reproduces experiments in all explored regions of phase space. Thermal trimer and solid state energetics were calculated, and successfully tested the transferability of the defined potential models.

Including an ostensibly redundant dimer orientation, the **S** configuration, led to a more robust PES. Further, only with the improved surface did both sets of mixing rules give similar solid energetics. This suggests mapping the PES to carefully chosen stochastic sets of dimer configurations to exhaustively explore configuration space, and this approach³⁸ will be pursued in the future.

2.7 Acknowledgements

Financial support from the National Science Foundation (Grant No. CHE-1152362) is gratefully acknowledged. Computations were performed on the following XSEDE resources (TG-DMR090028): PSC Blacklight, TACC Ranger and SDSC Trestles. Local computations were performed at the USF Research Computing Center, where NSF-funded computational resources (under Grant No. CHE-0722887) were greatly appreciated. The authors would also like to thank A. Erba for our communications, and his willingness to share data.

Chapter 3

A PHAST Potential for Heterogeneous Simulation of Methane with Many-Body Interactions

3.1 Introduction

Methane is a fossil fuel that is commonly found in natural gas, hydrocarbon fields, air and in the form of methane ice (clathrates), among other sources. Atmospheric methane is one of the more potent greenhouse gases and current levels are at an all-time high. As our fossil fuel dependent society continues to consume energy at an aggravated rate, containing and separating methane efficiently and completely from other molecular species, such as nitrogen (N_2), carbon dioxide (CO_2), water vapor (H_2O), hydrogen sulfide (H_2S) and other impurities, is an important goal. Currently, purification of methane from natural gas and biogas is achieved via permeation through engineered membranes.⁷¹ Not limited to polymers and mixed matrix membranes (MMM), metal-organic materials (MOM) and associated frameworks (MOF) offer a competitive solution. With oftentimes high porosity and chemical tunability, MOFs can be rationally designed for maximal separability and/or retainment of gaseous CH_4 . Frameworks such as ZJU-60,⁷² dia-7i-1-Co,^{73,74} and PCN-14⁷⁵ have already been synthesized for the engineered selectivity of CH_4 versus CO_2 and C_2 -hydrocarbon mixtures.

The efforts herein are focused on designing an accurate and transferable classical model for the simulation of methane in material environments. A nine-site CH_4 model has been developed from first principles calculations. It is named Me-PHAST (Potentials with High Accuracy, Speed and Transferability) and is part of the PHAST family of potential energy functions developed in the Space Research Group, alongside N_2 (N_2 -PHAST, see Chapter 2) and CO_2 (CO_2 -PHAST)

and thus is constructed according to the same procedures. The potential parameters are fit to reproduce high level ab initio calculations and are designed to be transferable to exogenous systems through parameter mixing.

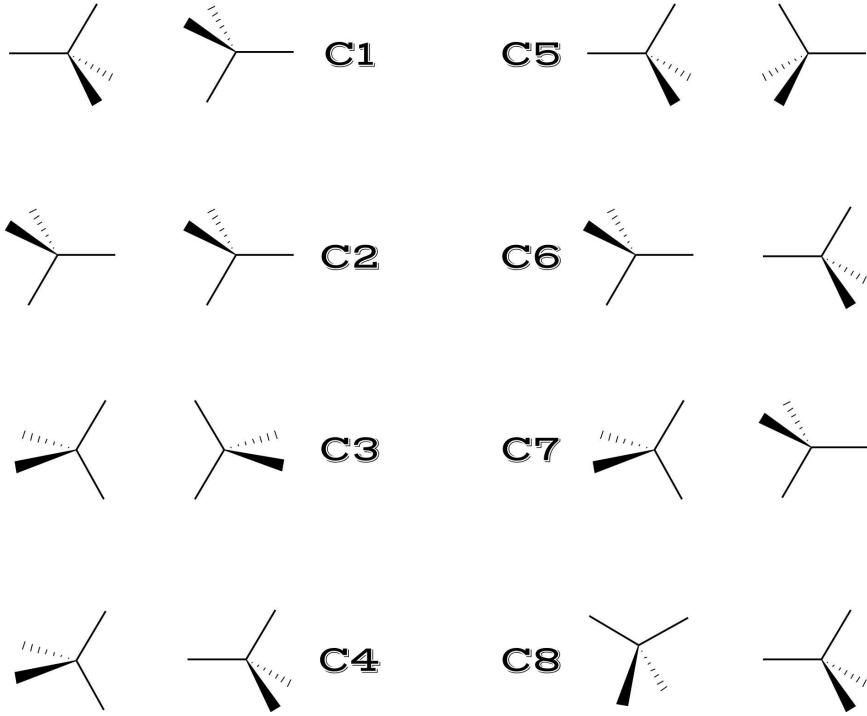


Figure 3.1: The eight representative CH_4 dimer orientations, labeled **C1-8**.

3.2 Methods

3.2.1 Born-Oppenheimer Surface

The methane molecule was approximated as rigid with a bond length of 1.099 Å, as determined by experiment.²⁷ To construct the Born-Oppenheimer potential surface for $\text{CH}_4\text{-CH}_4$ (depicted in Figure 3.2), eight unique dimer orientations were selected as representative of the geometries and energetics explored by methane in the condensed phase, namely, **C1-8** (see Figure 3.1).¹⁴ For each configuration, the pair energies were calculated for center-of-mass distances r apart along the z -axis. In the domain spanning $r = 3.00$ to 8.00 Å, energies were calculated every Δ0.05 Å. All ab initio calculations were performed using Molpro³² at the CCSD(T)³³ level of theory, unless otherwise stated. The basis functions used were the augmented correlation-consistent

sets of Dunning et al. (aug-cc-pVTZ/QZ).^{34,35} Basis set superposition errors were counterpoise-corrected via the method of Boys and Bernardi,³⁶ and the energy eigenvalues extrapolated to the complete basis set (CBS) limit in a standard fashion.³⁷

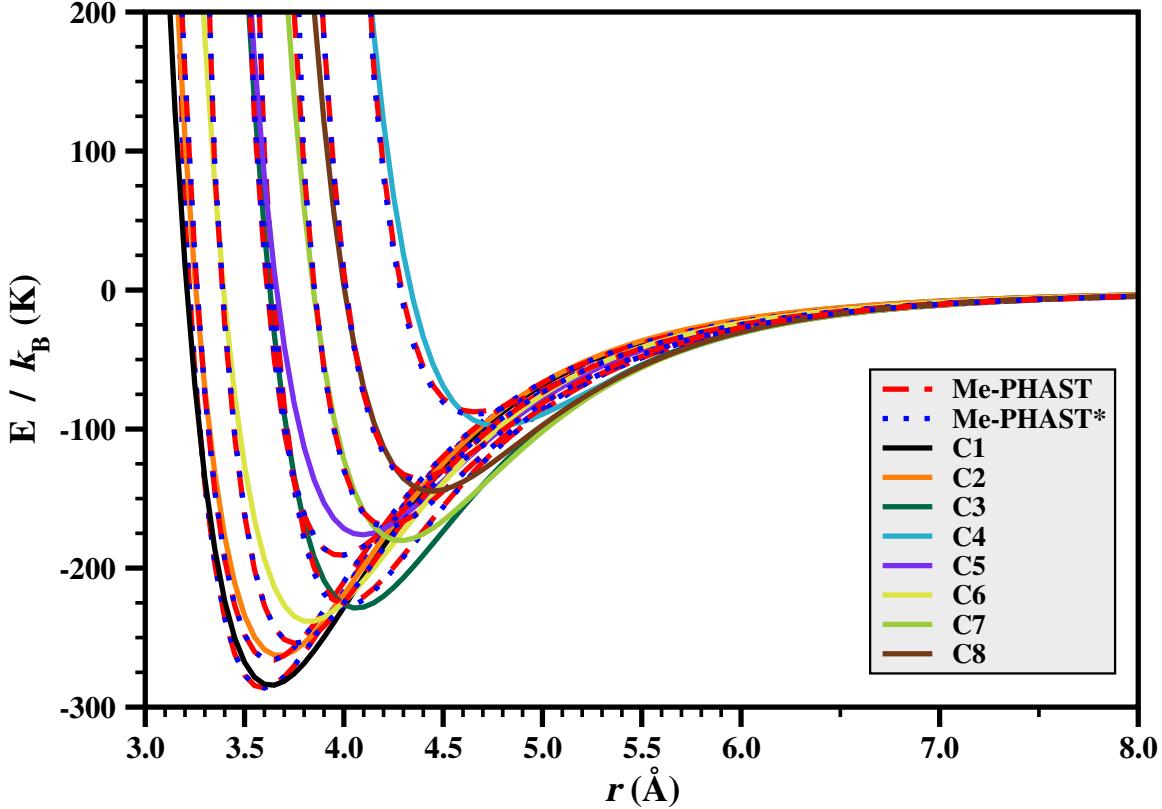


Figure 3.2: The reference Born-Oppenheimer potential surface of the CH_4 dimer (solid lines), calculated at the CCSD(T) level of theory, along with the Me-PHAST (dashed red lines) and Me-PHAST* (dotted blue lines) potentials. The illustrated potentials make use of the Lorentz-Berthelot (LB) mixing rules.

3.2.2 Many-body Polarization

A highly transferable potential energy function for use in charged/polar heterogeneous environments requires inclusion of induction effects which typically demand a many-body treatment. Thus a point atomic polarizability model of the Thole-Applequist type was adopted. The details of this model are omitted here, as a thorough description is provided in Chapter 2, Section 2.4.2.

3.2.3 Potential Energy Function

The methane potential model developed here is named Me-PHAST*, where Me represents methane, * denotes the inclusion of explicit polarization, and PHAST (Potentials with High Accuracy, Speed and Transferability) is our previously defined acronym⁷⁶ which refers to the form of our growing group of broadly applicable potential energy functions.^{17,18,20,76} We additionally define a distinguishable model which neglects polarization effects, Me-PHAST, for comparative purposes. Suitable for both molecular dynamics and Monte Carlo simulations, the PHAST potential has been designed with transferability in mind by explicitly including all essential potential energy interactions encountered in describing complex systems from the gas and liquid phases. The form of our potential includes contributions to the many-body polarization, electrostatic and the electronic repulsion/dispersion energy, respectively, as

$$U = U_{pol} + U_{es} + U_{rd} \quad (3.1)$$

The form of U_{pol} is given by Equations 2.3 and 2.7. To evaluate the polarization energy, atomic point polarizabilities are required and were determined in an iterative manner. Given the ab initio polarizability tensor, calculated in NWChem⁴⁵ at the CCSD(T)/aug-cc-pVTZ level of theory,

$$\hat{\alpha}_{\text{CH}_4} = \begin{pmatrix} 2.488 & 0 & 0 \\ 0 & 2.488 & 0 \\ 0 & 0 & 2.488 \end{pmatrix} \text{\AA}^3 \quad (3.2)$$

one constructs a model tensor via the Thole polarization model in attempt to match the ab initio tensor. Initial values for point polarizabilities were chosen from a carefully constructed training set, developed by van Duijnen *et. al.*⁴² Through refinement, the ab initio tensor, that itself is in agreement with calculated measurements of the trace,²⁷ was exactly reproduced. The atomic point polarizabilities which gave the best fit to $\hat{\alpha}_{\text{CH}_4}$ are $\alpha_{\text{C}} = 1.1297 \text{ \AA}^3$ and $\alpha_{\text{H}} = 0.4469 \text{ \AA}^3$.

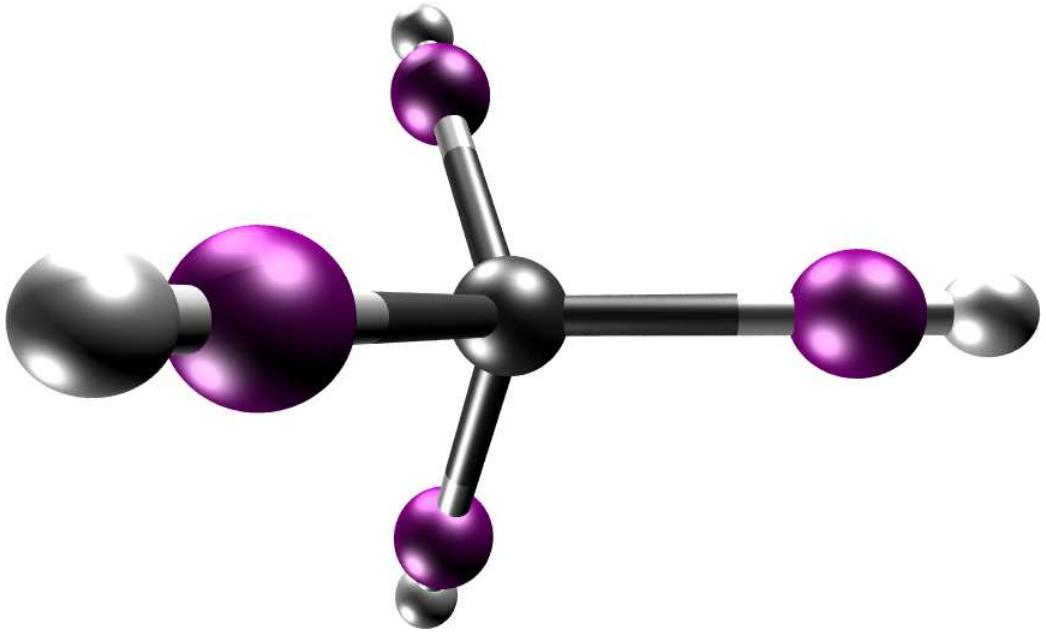


Figure 3.3: A perspective view of the nine-site methane monomer, indicating the positions of the three different site types: MeC (carbon, gray), MeH (hydrogen, white) and MeF (phantom site, purple). Note that the “F” in MeF denotes that it is a floatable site, and thus its position is determined through rigorous stochastic optimization.

Note that modeling without induction amounts to setting $U_{pol} = 0$, and when this is the case our model is referred to as Me-PHAST (nonpolar), whereas Me-PHAST* signifies the inclusion of polarization, as previously mentioned. Constructing a model neglecting explicit polarization (the Me-PHAST model has some measure of two-body polarization implicit in the potential) is useful for widely-employed simulation codes and appropriate when strong charge interactions are less relevant. Me-PHAST is also useful as a negative control in modeling phenomena such as sorption in polar media, in which case it can provide a metric for the degree of substrate polarization.

Figure 3.3 provides a visual depiction of the nine-site methane model. Five sites, four labeled MeH and one labeled MeC, are atom-centered on the hydrogens and carbon atoms, respectively. Both the MeH and MeC sites have associated charge and polarization parameters. Additionally, the MeC site contains a Lennard-Jones ε and σ value. The remaining four sites, named MeF, reside along the C–H bond vector between the MeC and MeH sites and are Lennard-Jones sites.

Table 3.1: Potential parameters (site position along the C–H bond vector (**R**) and Lennard-Jones epsilon (ε) & sigma (σ) of the polar (Me-PHAST*) and nonpolar (Me-PHAST) models.

| Model | Site | R (Å) | Q (e) | α° (Å ³) | ε (K) | σ (Å) |
|------------------------------|------|--------------|--------------|----------------------------------|-------------------|--------------|
| CH₄-PHAST* | MeC | 0.000 | -0.5868 | 1.1297 | 45.0973 | 2.1625 |
| | MeF | 0.815 | 0.0000 | 0.0000 | 18.5717 | 2.9479 |
| | MeH | 1.099 | 0.1467 | 0.4469 | 0.0000 | 0.0000 |
| CH₄-PHAST | MeC | 0.000 | -0.5868 | 0.0000 | 58.5387 | 2.2242 |
| | MeF | 0.817 | 0.0000 | 0.0000 | 16.8542 | 2.9629 |
| | MeH | 1.099 | 0.1467 | 0.0000 | 0.0000 | 0.0000 |

Note, the mass-containing sites are those of the MeC and MeH types; the MeF site is massless and is sometimes referred to as a “phantom” site.

Further deconstructing the potential energy function piece-wise, the electrostatic energy term U_{es} is parameterized to reproduce the octupole-octupole interactions between methane molecules. The expression for the octupole moment of a tetrahedral molecule is:

$$\Omega = \Omega_{xyz} = \frac{5}{2} \sum_i e_i x_i y_i z_i \quad (3.3)$$

where the x -, y -, and z -axes are the sides of a cube at whose center is the carbon atom and at four of whose corners are hydrogen atoms.⁵² The ab initio molecular electric octupole tensor is $\Omega = 2.7231$ a.u., which was adopted from Hellmann et al.¹⁴ Given this, and the positions of the nuclei, rearrangement of Equation 3.3 gives atomic charges that reproduce the first non-vanishing multipole moment, by construction. The resulting charges on the hydrogen atoms are each $q_{\text{MeH}} = +0.1467$ e, and on the carbon atom at the molecule’s center $q_{\text{MeC}} = -0.5868$ e. These values, along with the complete list of potential parameters, are tabulated in Table 3.1.

The electronic repulsion/dispersion contribution to the total energy, U_{rd} , is described via a Lennard-Jones 12-6 function. Pair mixing of non-identical sites (i.e. cross-interaction) is treated via the Lorentz-Berthelot (LB) combining rules. The well-known LB rules estimate the well

depth ε_{ij} for a mixed pair of unlike sites (ij , where $i \neq j$) as a geometric mean, and the core diameter σ_{ij} as an arithmetic mean:

$$\begin{aligned}\varepsilon_{ij} &= (\varepsilon_{ii}\varepsilon_{jj})^{1/2} \\ \sigma_{ij} &= \frac{1}{2}(\sigma_{ii} + \sigma_{jj})\end{aligned}\quad (3.4)$$

The LB mixing rules have the advantage that they are extensively tested and widely implemented.^{77–81} There does exist other mixing rules, namely that of Waldman and Hagler⁵³ (WH) which have shown superiority in reproducing mixed rare gas interactions, but have not been investigated in this work. Note that it is required to make use of mixing rules even in constructing a CH₄ pair potential as there are unlike sites within the proposed model (see Figure 3.3).

The MeF site positions along the C–H bond vectors (F signifies “floatable”) in addition to the Lennard-Jones potential parameters ε and σ were included in the multi-dimensional stochastic fitting to the Born-Oppenheimer surface (shown in Figure 3.2). Note, during this process the parameters associated with U_{pol} are held static at the values previously defined. Also note that the MeF site positions were constrained to scale in unison such that the COM displacements along each C–H bond vector remain equivalent.

3.3 Model Validation

At 150 K, simulations were performed at pressures up to 30 atm. The resulting isotherm, plotted against experiment, is shown in Figure 3.4. Note that a phase transition (vapor → liquid) occurs at 10.264 atm,⁵⁷ and that our potential captures the densities well. It is important to test our potential in this neighborhood of phase space to ensure that the potential accurately captures the transition and existence in the liquid phase.

Isothermal pressure-density plots were also constructed at 298.15 K, and are illustrated in Figure 3.5. At this temperature, the potentials are validated up to 600 atm and maintain excellent agreement with experiment throughout.

The developed models are then implemented in grand canonical Monte Carlo (GCMC) simulations, to model and measure methane uptake in MOFs. In particular, one structure with published results is dia-7i-1-Co where the Me-PHAST model reproduced experimental sorption isotherms to good agreement.⁷³

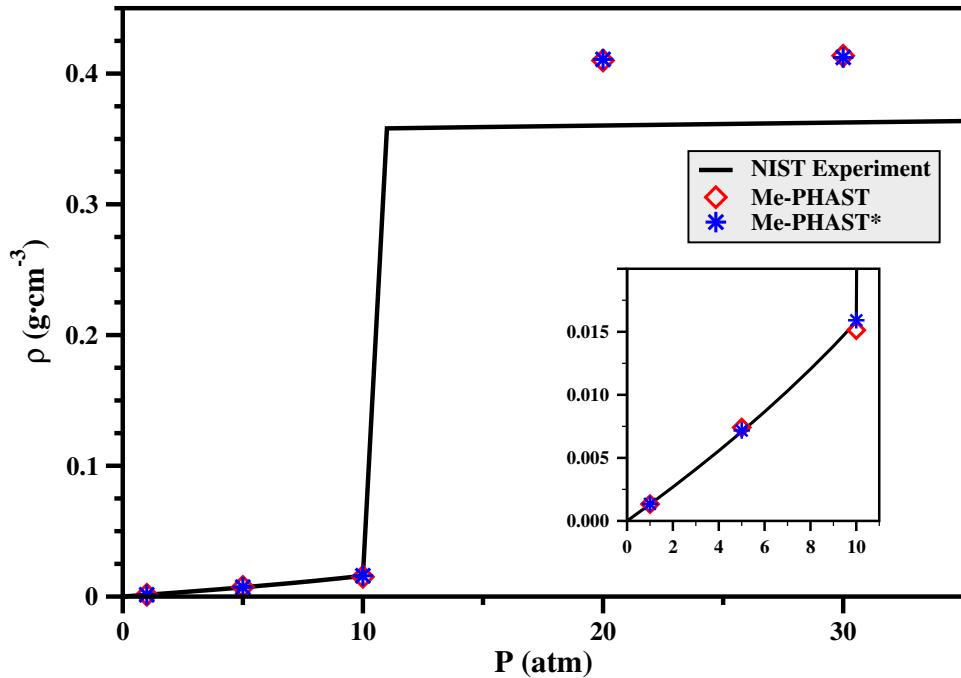


Figure 3.4: Pressure-density isotherm of bulk methane at 150 K. The inset plot depicts the simulated low pressure points of the gas phase region.

3.4 Limitations and Alternative Potentials

While it is certain that an adequate potential parameter set has been determined, it is imperative to discern that the reported parameters are by no means unique. In fact, there are several equally seemingly adequate parameter sets capable of reproducing bulk thermodynamic properties. To illustrate this point, a stochastic optimization of large proportion (a so-called “mass fit” of 200 attempts) was carried out. Figures 3.6 and 3.7 give a visual depiction of the resulting distribution in parameter space for both nonpolar and polar potential fit attempts. It can be seen that the distribution is uncomfortably great, indicating that there are several possible parameter combinations within a reasonable delta of sum of squared errors (SSE).

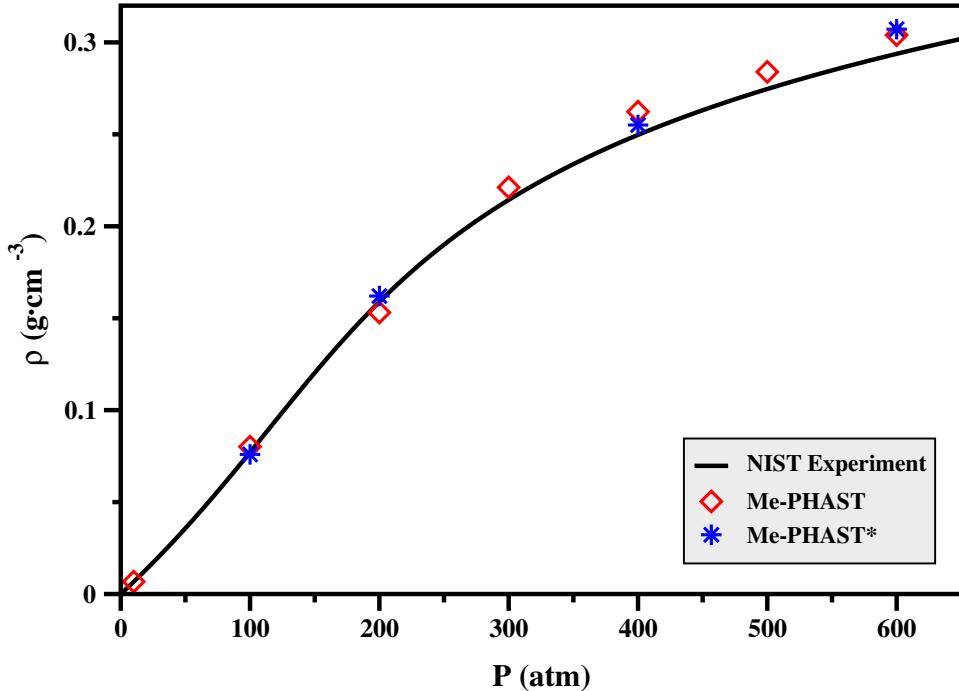


Figure 3.5: Pressure-density isotherm of bulk methane at 298 K.

The parameter set which yields the lowest squared error (i.e. the best fit to the ab initio dimer potential surface) does not necessarily correspond to the best selection, however. As an external example, even for the three-site model of TraPPE²² for CO₂,²⁰ the best overall fit contains an end-on-end (E) configuration that does not map well to the dimer surface, yet still excellent neat simulations result. Ultimately, we seek the parameter set which best reproduces sorption isotherms in a MOF. For this reason, the exploration of alternative potential forms is underway, the details of which are outlined in Chapter 4.

3.5 Conclusions and Future Directions

Using eight high level reference ab initio curves (CCSD(T) + aug-cc-pVQZ/TZ), a nine-site anisotropic and transferable potential energy function for CH₄ has been developed. Two potential models, one including (Me-PHAST*) and one neglecting (Me-PHAST) explicit induction effects, have been defined. All have been extensively tested and proven accurate for bulk systems across a wide region of phase space, encompassing the condensed, vapor and supercritical

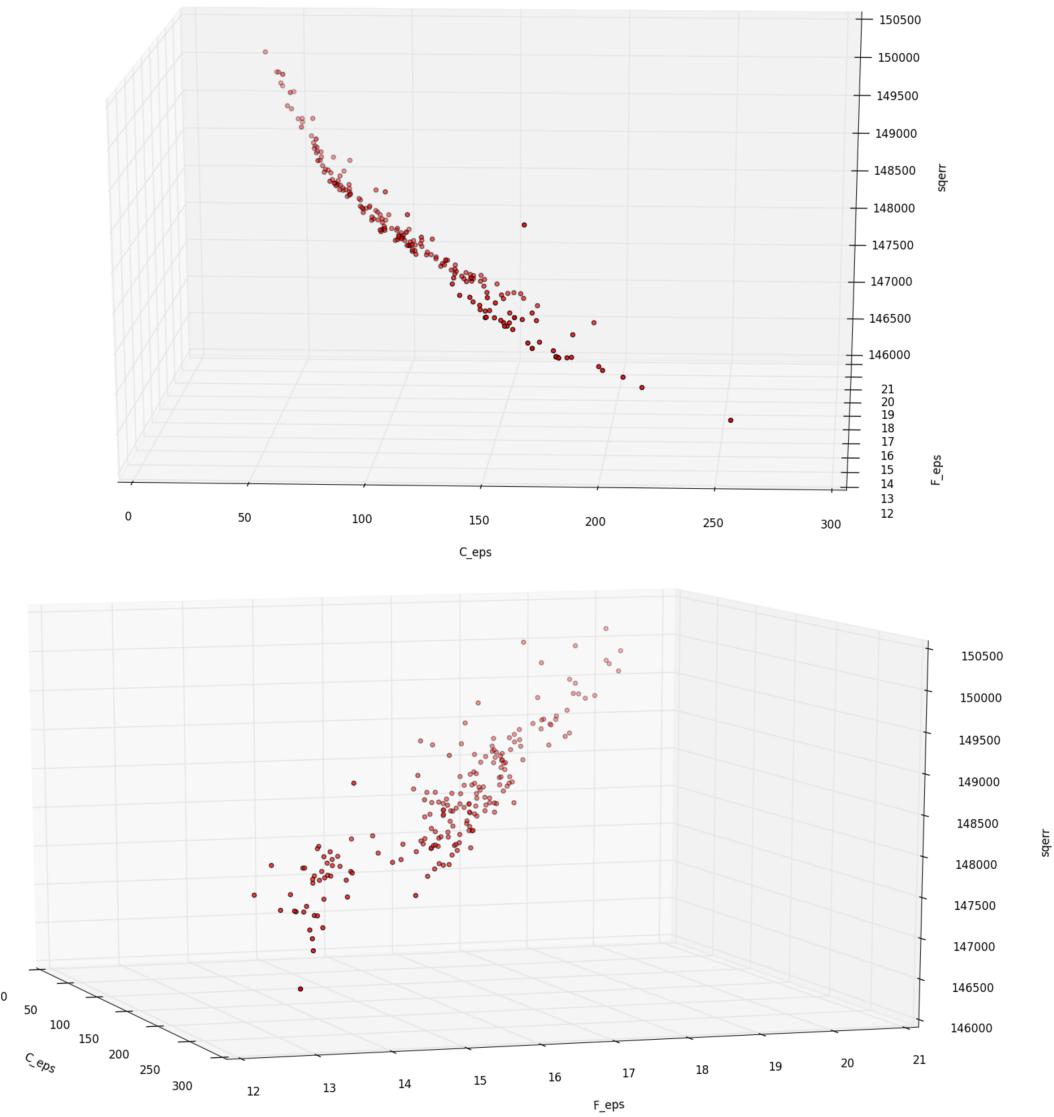


Figure 3.6: The distribution of the Lennard-Jones ε parameter for the MeC and MeF sites across 200 fitting attempts.

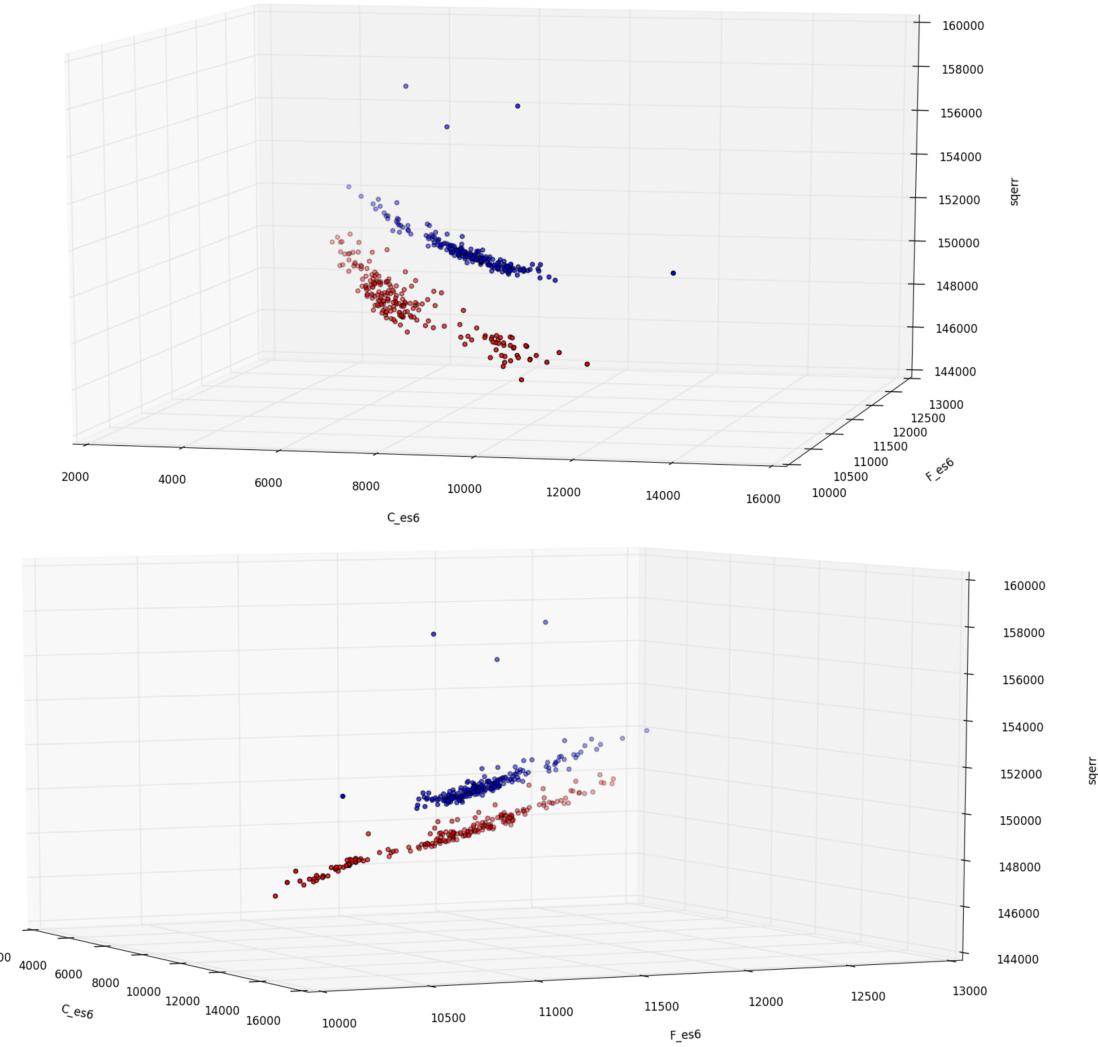


Figure 3.7: The distribution of C_6 for the MeC and MeF sites across 200 fitting attempts. The red circles indicate a nonpolar (i.e. $U_{pol} = 0$) potential, whereas the blue circles correspond to explicit inclusion of many body polarization effects.

phases. Pressure-density isotherms were constructed at 150 K and 298.15 K, and nearly exactly reproduce experiment in all explored regions of phase space. The mixing rule applied throughout this work was that of Lorentz-Berthelot (LB). It is recommended that any further progression of this project should begin with the development of a parameter set using the Waldman-Hagler (WH) mixing rules. Further, as alluded to in section 3.4, it is worth exploring the adoption of a different potential form for the repulsion/dispersion contribution to the total energy, in specific that of Tang-Toennies.⁸² The most current work on potential development, detailed in Chapter 4, suggests that in most cases the Lennard-Jones potential does not accurately reproduce high level ab initio energies.

Additionally, a study of crystalline methane, as has been explored with dinitrogen (in Chapter 2), might be of interest to further validate and stress test the model. There are at least seven known phases of methane ice,⁸³ of which only three have been solved. The crystal structure of methane phase III has most recently been reported.⁸⁴ It is orthorhombic with space group *Cmca*, and contains 16 molecules in the unit cell. One could make immediate use of the python script “solidN2.py”, the source code of which is presented in Appendix A, to begin simulating this low temperature, moderate pressure (≥ 200 bar) phase. Only minor modifications to the script need to be made, as the N₂ primitive and basis vectors are hard coded. Note that this script was modified and used in simulating solid carbon dioxide (phase I: *Pa3* space group, face-centered cubic), the results of which have been published by Mullen *et al.*²⁰

Chapter 4

Tuning Potentials for Heterogeneous Environments

4.1 Introduction

The existing developmental process and potential energy function including Lennard-Jones interactions (detailed in Chapters 2 & 3) has revealed limitations. To date, this extensive development and testing has produced effective simple empirical force fields for H_2 ^{17,18} and CO_2 ²⁰ interacting with MOFs. However, even now, after much application and inquiry it is not entirely clear why our potentials work so well. Investigations by our group and others^{18,20,24,28,29,76,85,86} have demonstrated that superior potential energy forms are necessary and that current fitting protocols are inadequate (as exemplified in Chapter 3, Section 3.4). Further, the extensive modeling efforts of the Space research group have examined sorption in hundreds of MOFs with many sorbates.^{6,15,19,73,87–98} These investigations suggest that extant models of, e.g., N_2 , CH_4 and even noble gases are unreliable. The work herein consists of constructing various configurations composed of clustered helium atoms which are geometrically akin to those encountered in MOF sorption environments. An additional helium atom serves as the incident particle to be adsorbed while the concurrent energetics along a trajectory are calculated.

The introduction of helium atoms was chosen because the dimer surfaces can still be calculated at the CCSD(T) level with complete basis set (CBS) extrapolation (aug-cc-pVQZ/TZ) and also because the potential energy surface of helium is very well known. The later is critical because of the issue of potential parameter mixing rules. Because it is desired to model heterogeneous media with multiple sorbates, mutually interacting and sorbing, the issue of cross terms in the energy function arises. Further, the sorbate model itself has distinct sites with, for example,

repulsion/dispersion at the center of the linear molecules. Many common force fields use Lennard-Jones potentials and Lorentz-Berthelot mixing rules with arithmetic means for the diameters and geometric means for the well depth.⁹⁹ Better choices have been developed and evaluated for this form such as those proposed by Waldman & Hagler.^{17,100} Further, more sophisticated potential forms have been shown to have reasonably reliable and predictable mixing behavior¹⁰¹ with approaches strongly guided by theoretical principles.¹⁰²

Nonetheless, because the form of the He potential is known to high precision, and given its chemically inert nature, interacting predominantly by van der Waals attraction over longer distances, the composite system makes an excellent testing ground for the mixing rules aspect of the empirical potential development. Four distinct helium atom clusters will be implemented and the adsorption energy profile of a single He atom will be constructed. In this developmental work with a newly introduced potential form, connections between geometries, mixing rules, parameter sets, and reproducibility of the CCSD(T) energies will be sought. This area of model development is often not emphasized but is critically important in developing a family of empirical potentials (especially for sorption and MOFs) for use in simulating a wide and emerging array of materials. Fortunately, related studies have already shown substantial promise.^{101,103–105}

4.2 Methods

4.2.1 Potential Energy Function

While our existing potential energy development methodology is sound, we intend to explore an alternative approach to design sorbate potentials for use in materials simulation. We have begun developing potentials for the Tang-Toennies type, which include C₆, C₈, and C₁₀ dispersion coefficients and exponential repulsion. This differs from our existing potential form, PHAST, where the repulsion and dispersion energies are described by the Lennard-Jones epsilon and sigma parameters. Representing the permanent electrostatics as point charges and modeling polarization through the use of induced dipoles is retained. Making use of the new potential form, it was discovered that the fits to the Born-Oppenheimer dimer surface are not as accurately

reproduced compared to the PHAST potential, for the N₂ dimer, as illustrated in Figure 4.1. If, however, a linear arrangement of three helium atoms are introduced into the dimer system (shown in Figure 4.2) and the energy as a function of dimer separation distance determined, it becomes clear that the PHAST models are incapable of reproducing not only the ab initio energies but the shape of the intermolecular potential as well. The Tang-Toennies type potentials, however, do in fact replicate the CCSD(T) results, which along with them being more physically grounded in intermolecular perturbation theory, indicates that they are more applicable to heterogeneous systems. The sorbate potentials are now taken with exponential repulsions and include reciprocal six, eight and ten terms for dispersion as assessed using high quality SAPT.^{103–108}

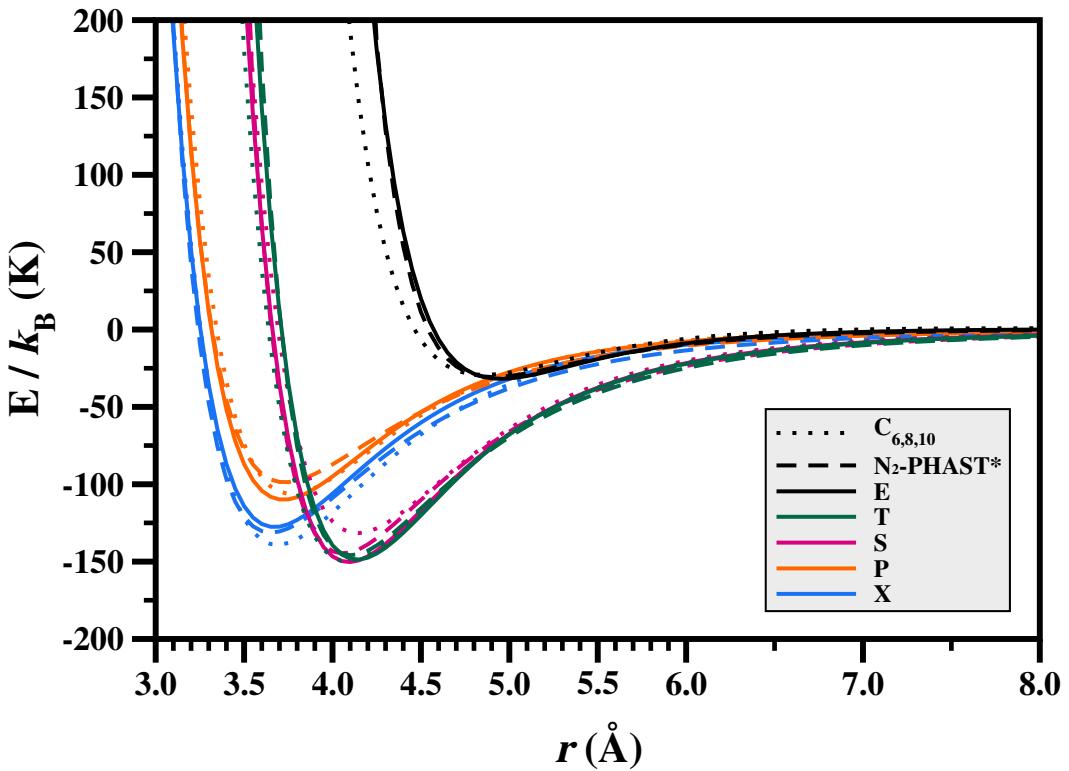


Figure 4.1: The ab initio Born-Oppenheimer surface for highly symmetric dinitrogen dimers (solid lines) alongside the resulting classical potential fits (dashed and dotted lines).

The form of the new potential is the following:

$$U = U_{rep} + U_{disp} + U_{es} + U_{pol} \quad (4.1)$$

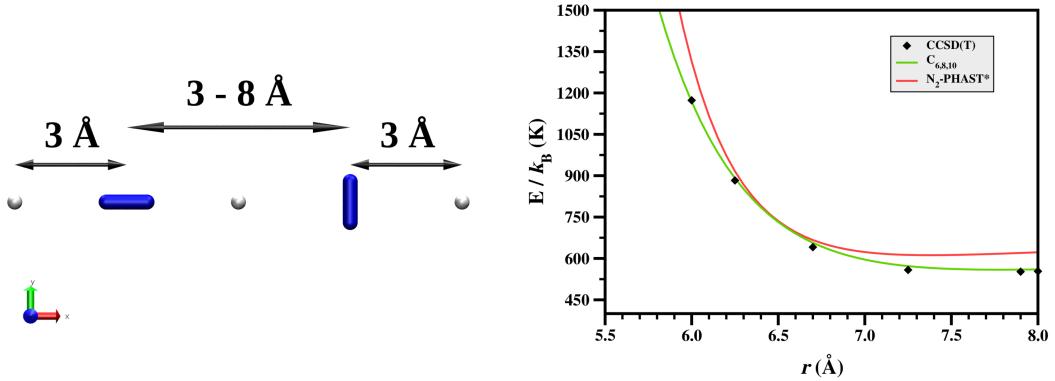


Figure 4.2: (**Left**) A schematic of the proposed “interfacial” fitting with He atoms. Depicted is one such possibility of a linear array with outer preserved distances from the N_2 monomers, but an increasingly distant fixed central He atom. (**Right**) The resulting curves of the He- N_2 -He- N_2 -He system. The extrapolated CBS CCSD(T) result is shown as black diamonds, while the classical potentials are the solid lines (red is the existing PHAST potential and green is the newly proposed potential form).

where U_{rep} is the energy of electronic repulsion and is given by:

$$U_{rep} = 315.77 * \exp(-\varepsilon(r - \sigma)) \quad (4.2)$$

U_{disp} is the dispersion energy,

$$U_{disp} = -f_6(r, \varepsilon) * \frac{C_6}{r^6} - f_8(r, \varepsilon) * \frac{C_8}{r^8} - f_{10}(r, \varepsilon) * \frac{C_{10}}{r^{10}} \quad (4.3)$$

U_{es} is the electrostatic energy given by Coulomb’s law, and U_{pol} is the many-body polarization energy given by:

$$U_{pol} = -\frac{1}{2} \sum_i^N \vec{\mu}_i \cdot \vec{E}_i^{stat} \quad (4.4)$$

Table 4.1: Potential parameters for the four He clusters. Note the only varying parameter between systems is sigma.

| System | ε (\AA^{-1}) | σ (\AA) | C_6 | C_8 | C_{10} | α° (\AA^3) |
|--------|-------------------------------------|---------------------------|----------|-----------|----------|-----------------------------------|
| Cage | 4.49880 | 2.18205 | 1.407164 | 11.136350 | 107.964 | 0.20494 |
| Pocket | 4.49880 | 2.17705 | 1.407164 | 11.136350 | 107.964 | 0.20494 |
| Ring | 4.49880 | 2.14605 | 1.407164 | 11.136350 | 107.964 | 0.20494 |
| Wall | 4.49880 | 2.16605 | 1.407164 | 11.136350 | 107.964 | 0.20494 |

as previously defined in Chapters 2 and 3. As stated above, modeling polarization through the use of induced dipoles and representing the permanent electrostatics as point charges is retained.

The epsilon (ε) and sigma (σ) parameters in the U_{rep} term are determined through either fitting to the Born-Oppenheimer potential surface (i.e. CCSD(T)/aug-cc-pVQZ/TZ + CBS extrapolation) or symmetry adapted perturbation theory (SAPT). The first two dispersion coefficients (C_6 and C_8) are calculated using the atomic polarizabilities at imaginary frequencies with the help of the Casimir–Polder identity¹⁰⁹ and Gauss–Legendre quadrature.¹¹⁰ Higher order dispersion coefficients (in specific C_{10}) are not calculated in the same way due to the need for extremely large basis sets with high angular frequencies. Rather, this dispersion coefficient is calculated recursively using the formula presented by Thakkar:¹¹¹

$$C_{10} = \left(\frac{49}{40} \right) \frac{C_8^2}{C_6}. \quad (4.5)$$

Note that $f_n(r, \varepsilon)$ is a damping function which ensures the dispersion goes to zero at $r = 0$ and therefore does not infinitely increase, or “explode.”

The full list of parameters for the newly defined model is presented in Table 4.1. For optimization purposes, all parameters are held fixed except for sigma. To find the best sigma for each cluster configuration, we sweep a range within $\pm 10\%$ of the best guess value of 2.18405 \AA . Using a step size of $\Delta = 0.001 \text{\AA}$, a total of 438 different sigma values are tested.

4.2.2 Mixing Rules

In addition to introducing a new potential form, the scheme to “mix” parameters of dissimilar sites is also altered slightly. The well-known Lorentz-Berthelot (LB) and Waldman-Hagler (WH) mixing rules have previously been defined in detail in Chapter 2, Section 2.4.3 and thus shall not be reiterated here. To point, the mixing of sigma remains the same as with LB:

$$\sigma_{ij} = \frac{1}{2}(\sigma_{ii} + \sigma_{jj}) \quad (4.6)$$

while the epsilon mixing is newly defined as:

$$\varepsilon_{ij} = \frac{(\varepsilon_{ii} + \varepsilon_{jj})\varepsilon_{ii}\varepsilon_{jj}}{\varepsilon_{ii}^2 + \varepsilon_{jj}^2} \quad (4.7)$$

and is adopted from Schmidt.¹⁰⁴

4.3 Results and Discussion

Four geometrically characteristic clusters of helium atoms were chosen to represent environments commonly encountered at sorption sites within a MOF. They are named **C**, **P**, **R**, & **W**, referring to the **Cage**, **Pocket**, **Ring** and **Wall** geometries they resemble, respectively (see Figures 4.3–4.6). In all cases except **R**, the adsorbing particle directly encounters a He atom; with **R** the particle passes through the ring, mimicking a corridor within a porous material.

Figures 4.3–4.6 depict the energetic adsorption profile of a He atom in four characteristic sorption-like environments. In each of the cases, the potential parameters listed in Table 4.1 are used. The sigma parameter is exhaustively enumerated within a ±10% delta range of the best guess value ($\sigma = 2.18405 \text{ \AA}$) in order to determine the optimal sigma value for each sorption system. This is plotted against the potential curve which makes use of the sigma value that best fits to the isolated He-He Born-Oppenheimer dimer surface ($\sigma_{He-He} = 2.18205 \text{ \AA}$),

for comparative purposes. Recall that in the original style of parameter fitting (that which is employed throughout Chapters 2 and 3), the potential was constructed based on the best fits to the Born-Oppenheimer dimer surface. However this may not necessarily be the best practice, and thus it is intended to explore different methods of parameter fitting.

The optimal sigma values for the **C**, **P**, **R**, and **W** systems with a He atom as the adsorbing particle are presented in Table 4.2. That is, this value of sigma returns the minimum sum of squared errors (SSE) for the surface as compared to high-level CCSD(T) calculations. Note that in all cases except **R** (Figure 4.5), the Lennard-Jones potential is unable to accurately reproduce the shape of the CCSD(T) surface, thus demonstrating the necessity of the newly proposed potential form. The He-He Lennard-Jones parameters were adopted from Hirschfelder et al.,¹¹² and are $\varepsilon = 10.9$ K and $\sigma = 2.64$ Å. It is interesting to note that the σ_{min} determined by the **C** configuration exactly matches that of σ_{He-He} , while the largest difference observed occurs in the **R** configuration (1.66% difference from σ_{He-He}).

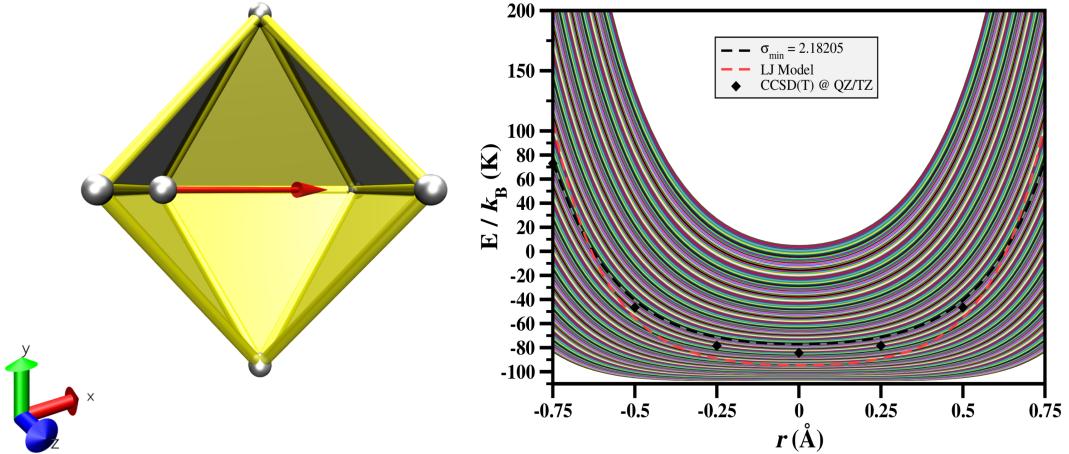


Figure 4.3: A helium atom translating through an enclosed cage of helium atoms. Note the 1D trajectory along the x -axis as depicted by the elongated red arrow. This caged shape of heliums is referred to as the **C** configuration. The resulting energy profile for this trajectory is displayed in the right panel.

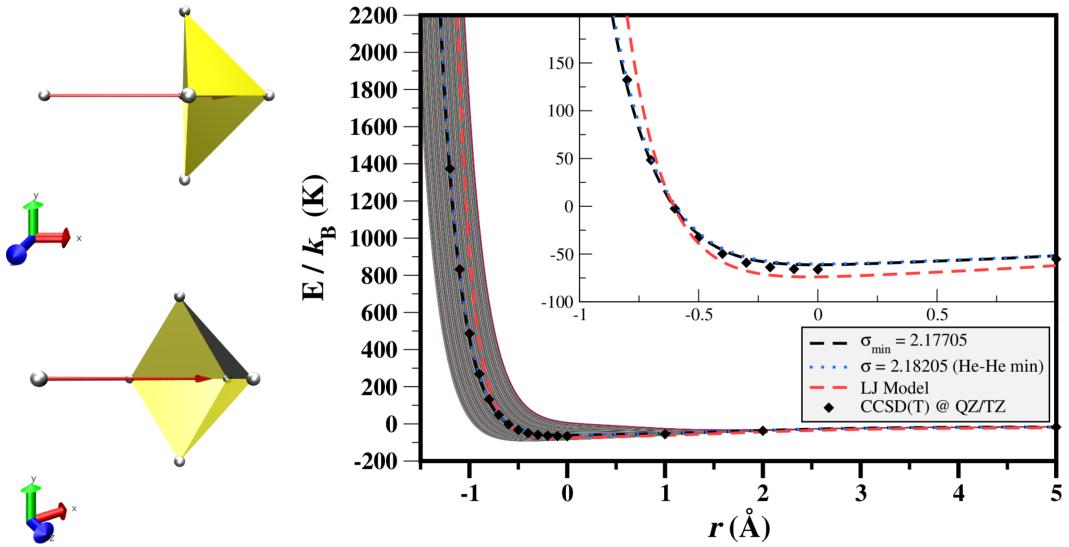


Figure 4.4: A helium atom approaching a pocket of helium atoms. Note 1D the trajectory along the x -axis as depicted by the elongated red arrow. This pocket shape of heliums is referred to as the **P** configuration. The resulting energy profile for this trajectory is displayed in the right panel.

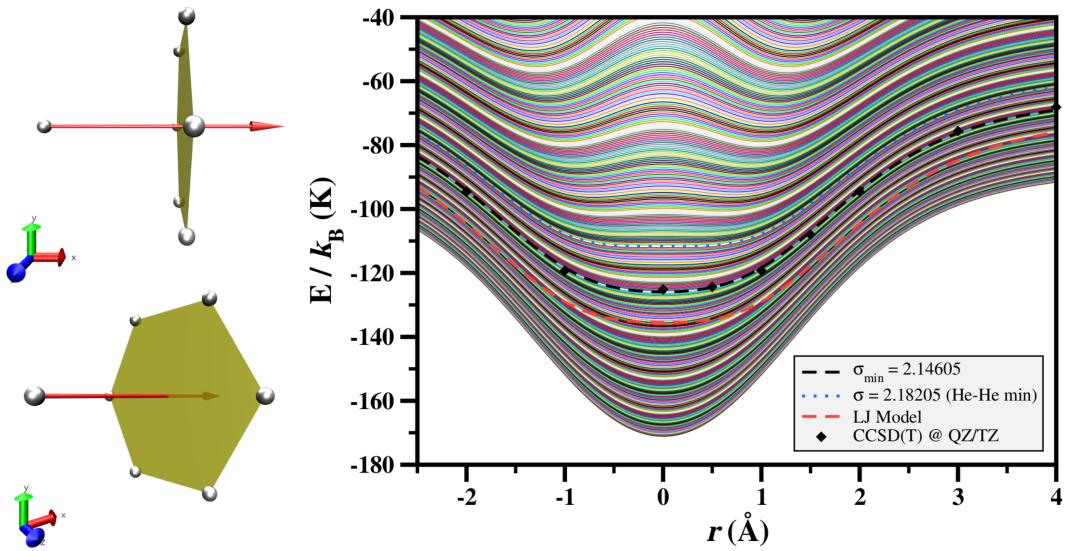


Figure 4.5: A helium atom passing through a channel, or ‘ring’ of helium atoms. Note the 1D trajectory along the x -axis as depicted by the elongated red arrow. This ring shape of heliums is referred to as the **R** configuration. The resulting energy profile for this trajectory is displayed in the right panel.

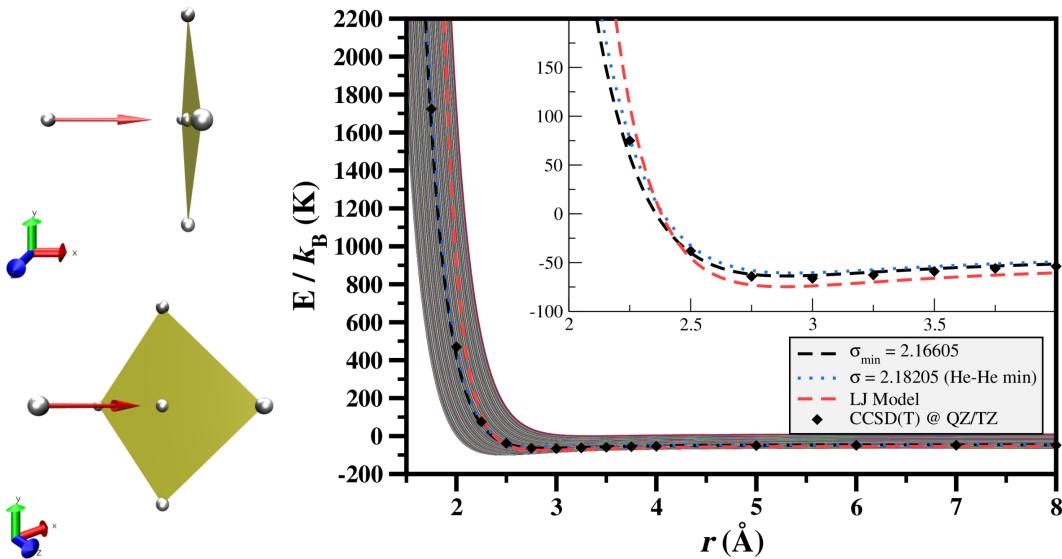


Figure 4.6: A helium atom approaching a plane, or “wall” of helium atoms. Note the 1D trajectory along the x -axis as depicted by the elongated red arrow. This wall shape of heliums is referred to as the **W** configuration. The resulting energy profile for this trajectory is displayed in the right panel.

Table 4.2: The sigma values which map best (i.e. returns the minimum sum of squared errors) to the CCSD(T) surface for the four He clusters with an additional He adsorbing particle.

| System | σ_{min} (\AA) | % Diff from He-He Min |
|-------------|---------------------------------|-----------------------|
| He-He Dimer | 2.18205 | — |
| C | 2.18205 | 0 |
| P | 2.17705 | 0.229 |
| R | 2.14605 | 1.66 |
| W | 2.16605 | 0.736 |

4.4 Conclusions and Future Directions

Four clusters of helium atoms are introduced, their geometries of which are representative of sorption environments commonly encountered in MOFs. An additional helium atom is then simulated to adsorb onto/through each cluster. The energy profile is captured at a high level of theory (CCSD(T) using large basis sets), and our newly defined potential energy function of the Tang-Toennies type is parameterized to reproduce this data. An exhaustive sweep of parameter space is explored for one parameter in particular, sigma, in order to define a relationship between the optimal value and the corresponding cluster geometry.

In addition to simulating adsorption with a helium atom, other alternative noble gases such as neon and argon should be used. Introducing disparate masses stress tests the parameter mixing rules employed, while still being able to maintain the use of high level CCSD(T) calculations. Even better, an explicit hydrogen molecule can serve as the incident particle. Eventually, higher Z atoms can be used to represent the framework fragments, though in this pioneering work we limit ourselves to helium clusters.

Four clusters of helium atoms are introduced, their geometries of which are representative of sorption environments commonly encountered in MOFs. An additional helium atom is then simulated to adsorb onto/through each cluster. The energy profile is captured at a high level of theory (CCSD(T) using large basis sets), and our newly defined potential energy function of the Tang-Toennies type is parameterized to reproduce this data. An exhaustive sweep of parameter space is explored for one parameter in particular, sigma, in order to define a relationship between the optimal value and the corresponding cluster geometry.

In addition to simulating adsorption with a helium atom, other alternative noble gases such as neon and argon should be used. Introducing disparate masses stress tests the parameter mixing rules employed, while still being able to maintain the use of high level CCSD(T) calculations. Even better, an explicit hydrogen molecule can serve as the incident particle. Eventually, higher Z atoms can be used to represent the framework fragments, though in this pioneering work we limit ourselves to helium clusters.

Chapter 5

Characterization of Tunable Radical Metal-Carbene: Key Intermediates in Catalytic Cyclopropanation

5.1 Note to Reader

This chapter contains content previously published in *Organometallics*, **2011**, *30* (10), 2739–2746, and has been reproduced within the guidelines provided by the American Chemical Society.

5.2 Abstract

A new class of radical metal-carbene complex has been characterized as having Fischer-like orbital interactions and adjacent π acceptor stabilization. Density Functional Theory (DFT) along with Natural Bond Orbital (NBO) analysis and Charge Decomposition Analysis (CDA) has given insight into the electronics of this catalytic intermediate in an open-shell cobalt-porphyrin, [Co(Por)], system. The complex has a single bond from the metal to the carbene and has radical character with localized spin density on the carbene carbon. In addition, the carbene carbon is found to be nucleophilic and “tunable” through the introduction of different α -carbon substituents. Finally, based on these findings, rational design strategies are proposed which should lead to the enhancement of catalytic activity.

5.3 Introduction

Recently, a new family of cobalt(II)-porphyrin complexes, [Co(Por)], have been introduced as effective catalysts for stereoselective cyclopropanation.^{113–122} The [Co(Por)]-catalyzed cyclopropa-

nation has a number of distinct advantages as compared to earlier catalytic systems described below. Given this, characterizing the catalytic cycle of this new class of [Co(Por)]-complexes is imperative.¹²³ Based upon experimental results, it has been hypothesized¹¹³ that the reaction proceeds through a stable metallocarbene intermediate with radical character on a nucleophilic α -carbon. These hypotheses are confirmed by extensive electronic structure calculations reported herein and the results explain a trend in experimental substrate reactivities.^{113–119}

Further, population and charge analyses demonstrate that the local “excess” charge density on the carbene carbon is “tunable”, as has been demonstrated for carbenes substituted with aromatic groups.¹²⁴ The magnitude of this charge can take on values from highly negative to nearly neutral (or in one unusual case, even slightly positive) values through the interchange of different carbene substituents. This suggests the ability to control reactivity *via* tuning nucleophilicity. The nature of the bonding in the [Co(Por)]-carbene complex is analyzed and found to represent a new class of metallocarbenes with demonstrated potential to serve as key intermediates for carrying out novel catalytic reactions in a controlled fashion.^{113–119} Moreover, the present study demonstrates that these radical carbene complexes are amenable to rational design in achieving desired electronic and steric characteristics.

Cyclopropanes have found numerous synthetic, biological and practical applications. One example is the use of cyclopropyl moieties to potentiate antibiotics targeted towards both gram-negative and gram-positive bacteria.^{125–130} Given this (and other) extremely important action(s), development of efficient and controllable synthetic methods to their stereoselective formation has been pursued with zeal.¹³¹ These efforts have lead to considerable success with the establishment of a number of catalytic cyclopropanation systems based on Cu-,^{132–135} Rh-,^{136–146} Fe-,¹⁴⁷ and Ru-^{148–155} complexes. Cobalt was first used as an enantioselective catalyst as a part of a chiral Co(II)-dioximato compound¹⁵⁶ and subsequently complexed to salen ligands in a 3+ oxidation state.^{157,158} However, these attempts lacked either diastereomeric or enantiomeric control. Such difficulties were recently overcome by employing [Co(Por)] as catalysts that exhibit both high diastereoisometric control and enantiomeric selectivity^{113–119} (Figure 5.1). In addition, the [Co(Por)]-based catalytic system can be operated in a one-pot reaction without the need for con-

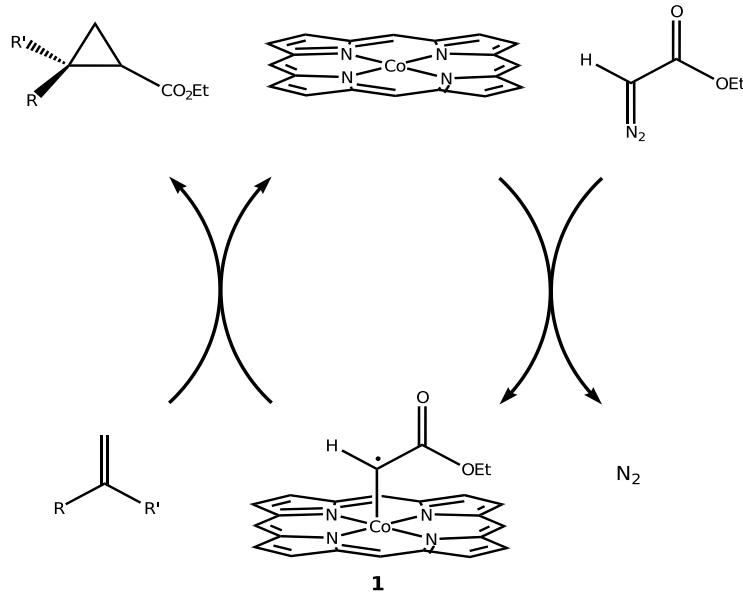


Figure 5.1: Cyclopropanation reaction performed by a cobalt(II)-porphyrin catalyst. The diazoacetate reagent reacts with the porphyrin to form the carbene intermediate studied in this work.

trolled diazo addition and with alkenes as the limiting reagents; this offers a significant advantage over previous metal-catalyzed procedures. Furthermore, with the support of various chiral porphyrin ligands, $[\text{Co}(\text{Por})]$ -catalyzed cyclopropanation reactions have been carried out for different alkenes with a wide range of diazo reagents.^{113–119} In particular, unprecedented catalytic reactivity toward electron-deficient olefins has been demonstrated with $[\text{Co}(\text{Por})]$ catalysts¹¹⁷ (for a highlight on the topic, please see a recent review by Doyle¹⁵⁹). Additionally, $[\text{Co}(\text{Por})]$ has exhibited other highly desirable catalytic capabilities that are fundamentally different from other metal-catalyzed reaction systems, such as the direct preparation of *N*-phosphorus-substituted aziridines from alkenes.^{160,161}

Given the differential reactivity and stereoselectivity of $[\text{Co}(\text{Por})]$ -catalyzed asymmetric cyclopropanation, it is only natural that obtaining a detailed understanding of the underlying catalytic mechanism, especially the nature of the key metallocarbene intermediate, is desirable. Fortunately, the advent of efficient parallel electronic structure codes affords the opportunity to examine the complexes with sufficiently accurate quantum mechanical calculations. For example, Yamada and co-workers have carried out coordinated computational and experimental work to

characterize the nature of the metal–carbene bond in a series of metallo-salen complexes, which are closely related to the metallo-porphyrin complexes. They asserted that the Co(II)-carbene interaction is best characterized as a single bond. It was also concluded that, with the carbene bonded to the metallo-salen, the lone remaining unpaired electron (*i.e.* a single radical) was delocalized along the carbonyl of the methyl diazoacetate ligand. The radical was characterized primarily by observing a 50 cm^{-1} shift in stretching frequency of the diazoacetate carbonyl group upon binding to the metal complex.¹⁶² In contrast, our electronic structure studies have indicated highly localized radicals on [Co(Por)]-complexes, but otherwise our conclusions are consistent with previous observations. Subsequently, they extended their work to [Co(Por)]-complexes and focused on the nature of the metal’s spin state and resulting bond orders.¹⁶³

Currently, there are three well-known classes of metal–carbene complexes: Fischer, Schrock and Grubbs.^{164–167} The former was first synthesized in 1964 and typically includes a low oxidation state metal whereas Schrock carbenes, first reported in 1974, are characterized by higher oxidation states. The bonding pattern in Fischer carbenes can be described as singlet carbenes bonding with singlet metals (Figure 5.2(a)). This is typically thought of as σ donation to the metal center and π backbonding from the metal to the carbene, as proposed in the Dewar-Chatt-Duncanson model.^{168,169} In contrast, Schrock complexes can be depicted as an interaction between triplet carbenes and triplet metals (Figure 5.2(b)). Bonding in these complexes has been characterized both experimentally and computationally with Schrock carbenes having much higher bond orders, with nearly double bonds, while Fischer complexes typically exhibit slightly less than single bond character.¹⁷⁰ Generally speaking, this is illustrated by both net σ and π bonding in Schrock carbenes and largely only effective σ bonding in Fischer carbenes. Detailed analysis does yield a more complicated and complete picture of the bonding in Fischer and Schrock complexes as described previously by Frenking.^{170,171} An additional property, important in the context of the present work, is the charge environment of the carbene center: Fischer carbenes exhibit charge separation between the metal and the carbene carbon (since the magnitude of donation is larger than the backdonation) and an electrophilic carbene carbon whereas Schrock

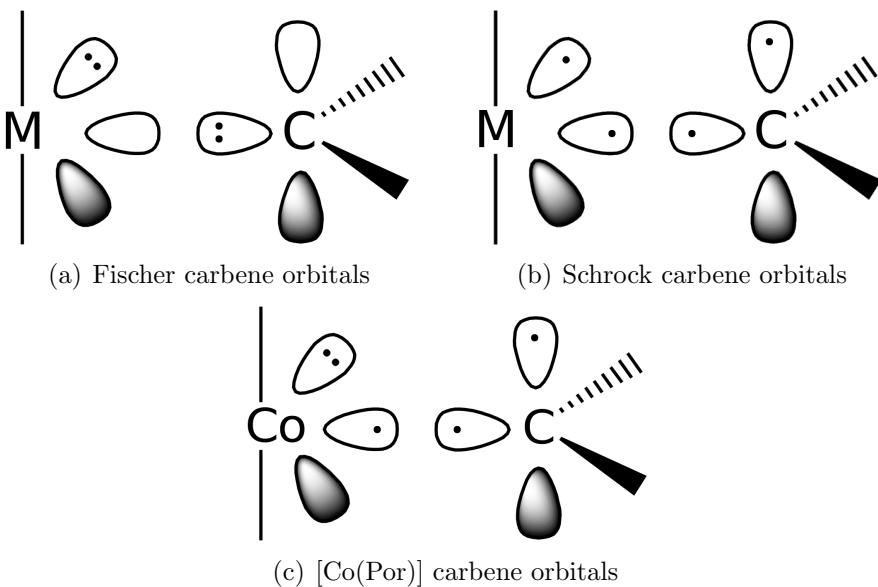


Figure 5.2: Illustration of Fischer, Schrock, and [Co(Por)] carbene orbitals. In Fischer carbenes (a), there is σ donation from the carbene to the metal, and π backdonation from the metal to the carbene. The Schrock (b) scenario shows spin-pairing between adjacent orbitals, thus resulting in a covalent interaction. The [Co(Por)]-carbene (c), presented in this work, consists of a doublet metal into which the carbene σ donates, with π backdonation from the metal to the carbene, in the case of having a good π acceptor on the ligand (as described in the text), with the porphyrin ring also serving as an electron “buffer” to keep the π d -orbitals filled and thus leaving a radical on the carbon. Additional detail can be obtained in a related study.¹²³

carbenes are nucleophilic. This becomes an extremely important property for cyclopropanation reactions, as electron deficient olefins are believed to react at nucleophilic carbene centers.^{113–119}

Quantum chemical methods have seen major improvements recently in both the efficient computation and analysis of transition metal complexes. For example, major strides have been made in the development of relativistic corrections¹⁷² and effective core potentials.^{173,174} These advances coupled with modern density functional theory (DFT), including improved functionals,^{175,176} have made computational treatment of coordination complexes more common.^{177,178} Additionally, powerful analysis tools like natural bond orbital analysis (NBO),^{179–181} charge decomposition analysis (CDA),^{170,171,182} absolutely localized molecular orbitals – energy decomposition analysis (ALMO-EDA),^{183,184} and charge transfer based ALMO-EDA¹⁸⁵ have made possible the interpretation of calculations in chemically meaningful terms.

The two main thrusts of this work are to provide a theoretical description of the presumed carbene intermediate in [Co(Por)]-catalyzed cyclopropanation and to predict new carbene sources to “tune” the property of the carbene intermediate for desired substrate reactivity. To the point, the process of computationally designing catalytic reactions is two fold: first an in-depth understanding of the nature of the key catalytic intermediate must be achieved, and second the interaction of carbene sources (*i.e.* diazoacetate reagents) with the metallo-porphyrin must be explained. The current work explores both the fundamental nature of [Co(Por)]-carbene interactions and highlights future modifications that can be made to control reactivity of asymmetric catalytic cyclopropanation. Calculations employ a simple, unsubstituted Co(II)-porphyrin, depicted in Figure 5.1, to limit computational cost. However, successful controls, listed in the Supporting Information (SI), were performed to ensure that this is a reasonable approximation. This choice allowed for a more detailed and systematic examination of carbene units shown to be important experimentally^{113–119} and to explore possible carbene sources for future experimental studies.

The remainder of this manuscript is arranged as follows. The Methods section outlines the computational approaches employed and their validation as reliable indicators of [Co(Por)]-complex structure and electronics. Furthermore, the analysis tools used to draw many of the conclusions are discussed. The Results and Discussion section examines the [Co(Por)]-carbene intermediate where it becomes clear that this is a fundamentally different type of transition metal-carbene complex from known Fischer- and Schrock-types. A detailed analysis of how diazoacetate and other diazo electronic structure partially controls the cyclopropanation reactivity is presented. In particular, attention is paid to possible alternative carbene sources (and their electronic properties) that can be used to control the reaction. Finally, the last section presents our conclusions, predicts new cobalt carbenes to be catalytically generated and proposes future directions.

5.4 Methods

The ground state of [Co(Por)]-complexes has been shown to be a doublet,^{113–116,163} which was also confirmed in this work by determining that the quartet state was higher in energy. Unrestricted Kohn-Sham DFT¹⁸⁶ was used in conjunction with the M06-L exchange-correlation functional¹⁷⁵ as implemented in the Q-Chem¹⁸⁷ and GAMESS-US¹⁸⁸ quantum chemistry packages. LANL2TZ basis functions and ECPs¹⁸⁹ were applied to the cobalt center of the metal-porphyrin with the remaining atoms being described by the 6-31G* basis set.¹⁹⁰ Structures were first geometry optimized on a coarse (50-194) standard grid¹⁹¹ (SG-1) and then converged with a finer grid (96-302) to obtain the structures for analysis. NBO analysis yielded chemically insightful molecular orbitals for the structures, in addition to the spin density at the nucleus, partial charges and bond orders.

CDA was performed in order to gain insight into the nature of the bonding taking place between the metallo-porphyrin and carbene ligand. Previous analysis of the nature of the bonding in Fischer- and Schrock-type transition metal carbene complexes via CDA¹⁹² is recognized. The following is a brief summary of the various terms calculated within the CDA methodology; please refer to reference¹⁷¹ for details. The charge donation, d , from the ligand (L) to the metal (M) is defined as the canonical overlap of occupied (occ) / unoccupied ($unocc$) orbitals:

$$d_i = \sum_j^{\text{occ}, L} \sum_k^{\text{unocc}, M} m_i c_j c_k \langle \phi_j | \phi_k \rangle \quad (5.1)$$

where ϕ is the canonical molecular orbital, m is the orbital occupation and c is the orbital coefficient. Similarly, the charge back-donation, b , from the metal to the ligand is given by:

$$b_i = \sum_j^{\text{occ}, M} \sum_k^{\text{unocc}, L} m_i c_j c_k \langle \phi_j | \phi_k \rangle \quad (5.2)$$

In addition, we may consider the repulsive polarization, r , (the net flow of electron density in or out of the overlapping region) as:

$$r_i = \sum_j^{\text{occ}, M} \sum_k^{\text{occ}, L} m_i c_j c_k \langle \phi_j | \phi_k \rangle \quad (5.3)$$

And finally, the rest term, Δ , yields the measure of covalent character present in the complex:

$$\Delta_i = \sum_j^{\text{unocc}, M} \sum_k^{\text{unocc}, L} m_i c_j c_k \langle \phi_j | \phi_k \rangle \quad (5.4)$$

where the great utility of the rest term, when coupled with quantifying back donation, is to distinguish between Schrock (covalent like) or Fischer (non-covalent like) carbene bonding as has been previously reported.^{170,171,182} Numerous controls were conducted to validate the computational methods and are described in more detail in the SI. Briefly, the conclusions presented were not changed by use of alternative functionals, higher level basis sets, modeling larger substituted porphyrins that were employed experimentally, or using a dielectric constant in the calculations that is characteristic of the experimental solvent environment. The substitution of the reduced complex for the parent complex was verified through several controls, and it was ultimately found that the reduced complex very accurately represented the full system. It was also checked that the methods reasonably described the individual carbene and porphyrin molecules. It was noted that the commonly used B3LYP¹⁹³ functional did not properly describe the electronics of the bare cobalt-porphyrin,^{194,195} (*e.g.* the HOMO was incorrect) but reasonably described the bonded [Co(Por)]-complex.

5.5 Results and Discussion

Numerous efforts have been made toward a better understanding of the nature of metal-carbene interactions. In particular, most computational efforts have focused on gaining insight into the nature of Fischer-¹⁶⁴ and Schrock-type¹⁶⁵ carbene complexes.^{170,171} It is generally accepted that

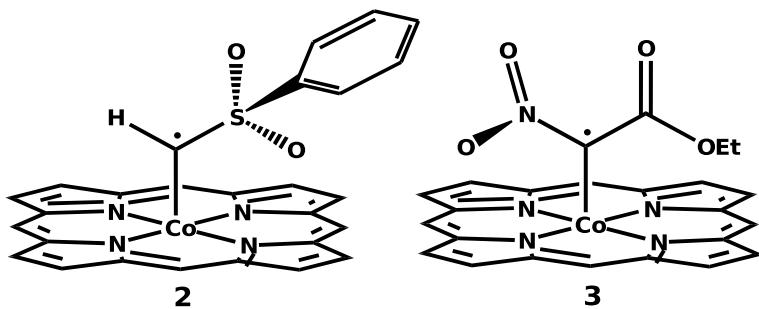


Figure 5.3: Perspective depiction of complexes **2** & **3**.

Fischer complexes are electrophilic with a metal–carbene bond slightly less than unity, whereas Schrock-types are nucleophilic and possess much more double bond character.

The current work is an attempt to characterize a novel metal–carbene complex (*i.e.* [Co(Por)]-complex) that serves as an active intermediate in a highly stereoselective cyclopropanation reaction.^{113–119} It is found that these carbene complexes are radical in character with bond orders similar to Fischer carbenes and nucleophilicity akin, although somewhat smaller in magnitude to Schrock-type complexes (Figure 5.2(c)). The nature and stability of these complexes were assessed through electronic structure analysis tools. First, NBO analysis and CDA were carried out on representative Fischer- and Schrock-type carbene complexes (**FC,SC** in Table 5.1). Subsequently, three [Co(Por)]-carbene complexes (**1–3** in Table 5.1, also see Figures 5.1,5.3) were chosen both to represent a range of experimentally observed reactivities and associated electronic properties. Three additional complexes (**4–6** in Table 5.1) serve to demonstrate the “tunable” nature of not-yet synthesized reactive intermediates. An additional complex, **7**, was included as a counterexample that displays distinct chemical bonding with some Fischer carbene character. Table 5.1 lists bond lengths, bond orders, and the cobalt and carbene α -carbon electrostatic potential (ESP) charges for all complexes examined (note that several similar complexes, including a variety of controls, were also studied with results reported in SI).

Spin densities and ESP charges reported in Table 5.1 clearly show significant radical character localized on nucleophilic carbene carbons, while the metal–carbene bond reveals single bond (or slightly smaller) bond orders. These results hold for all complexes except **7**, which

is clearly qualitatively different (*vide infra*). The most favorable intermediate toward reaction with electron-deficient olefin substrates is formed by reacting a [Co(Por)] with ethyl diazoacetate (*e.g.* ethyl acrylate is a prototypical choice, see Figure 5.1) resulting in **1**.¹¹⁷ This complex has a metal-carbene bond order of 0.90 (1.86 Å bond distance) and displays a significant negative partial charge on the carbene carbon; -0.357 electrons (*e*). While the bond order and bond distances observed are more consistent with those of Fischer carbene complexes¹⁷⁰ as discussed below, it is clearly nucleophilic in nature, consistent with its high reactivity towards electron deficient species. Of the synthesized complexes, **1** has the largest negative charge while the less negatively charged **2-3** are correspondingly less reactive.

From these results it was initially hypothesized that carbene carbon charge may be correlated to both metal-carbene bond character as well as reactivity. Therefore, a series of model compounds were employed (**8-9**, Figure 5.4) to test this proposition. We began with **1**, which has both a large negative charge on the carbon (-0.357 *e*) and a bond distance of 1.86 Å. To selectively adjust the charge on the carbene carbon, a methyl (**8**, -CH₃) group was substituted in place of the hydrogen. As expected, this both increased the charge on the carbene carbon (now -0.027 *e*) and elongated the metal-carbene bond to 1.90 Å. The next logical step was to substitute an ethyl group (**9**, -CH₂CH₃) in place of the methyl group, which again, as suspected, caused the charge to nearly return to that of **1** (-0.349 *e*). However, interestingly the bond length remained 1.90 Å. NBO analysis revealed this was due to hyperconjugation. In fact, significant electron density was observed to be donated from the σ C-H bonding orbitals on the alkyl group into the metal-carbene σ^* orbital thus explaining the maintained elongation of the metal-carbene bond. The trend of substituent stabilization by donation into the metal-carbene σ^* orbital was consistent throughout the complexes examined and correlated perfectly with metal-carbene bond distance and order.

Table 5.1: Electrostatic Potential (ESP) and Natural Bond Orbital (NBO) results. Bond distances (BD), charges (Q), spin densities (SD), and bond orders (BO) of representative metal-carbene complexes (Fischer-type, **FC**, and Schrock-type, **SC**) and a series of $[\text{Co}(\text{Por})]$ -carbene complexes are presented. Charges and spin densities are reported in units of (e) with bond distances in Å and unit-less bond orders, M refers to the metal and C refers to the carbene carbon.

| Label | Complex | M SD | M Q | C SD | C Q | M-C BD | M-C BO |
|-----------|--|---------|-------|-------|---------|--------|--------|
| FC | $(\text{CO})_5\text{W}-\text{C}(\text{CH}_3)\text{OCH}_3$ | - | -1.91 | - | 0.701 | 2.01 | 0.702 |
| | $(\text{C}_5\text{H}_5)_2\text{Ti}-\text{CH}_2$ | - | 0.655 | - | -0.860 | 1.93 | 1.58 |
| 1 | $[\text{Co}(\text{Por})]-\text{CHCO}_2\text{Et}$ | 0.0224 | 0.516 | 0.798 | -0.357 | 1.86 | 0.898 |
| | $[\text{Co}(\text{Por})]-\text{CHSO}_2\text{Ph}$ | 0.0246 | 0.471 | 0.920 | -0.185 | 1.85 | 0.886 |
| 2 | $[\text{Co}(\text{Por})]-\text{C}(\text{NO}_2)\text{CO}_2\text{Et}$ | 0.0725 | 0.502 | 0.649 | -0.113 | 1.85 | 0.858 |
| | $[\text{Co}(\text{Por})]-\text{CHSO}_2\text{H}$ | 0.0770 | 0.427 | 0.863 | -0.439 | 1.86 | 0.882 |
| 3 | $[\text{Co}(\text{Por})]-\text{C}(\text{CF}_3)\text{CO}_2\text{Et}$ | 0.00998 | 0.549 | 0.825 | -0.347 | 1.89 | 0.842 |
| | $[\text{Co}(\text{Por})]-\text{CH}_2$ | 0.109 | 0.396 | 0.931 | -0.425 | 1.86 | 0.947 |
| 4 | $[\text{Co}(\text{Por})]-\text{CF}_2$ | 0.711 | 0.119 | 0.282 | 0.0977 | 1.93 | 0.779 |
| | $[\text{Co}(\text{Por})]-\text{C}(\text{CH}_3)\text{CO}_2\text{Et}$ | 0.0130 | 0.395 | 0.771 | -0.0274 | 1.90 | 0.898 |
| 5 | $[\text{Co}(\text{Por})]-\text{C}(\text{CH}_2\text{CH}_3)\text{CO}_2\text{Et}$ | 0.0380 | 0.430 | 0.793 | -0.349 | 1.90 | 0.849 |
| | | | | | | | |

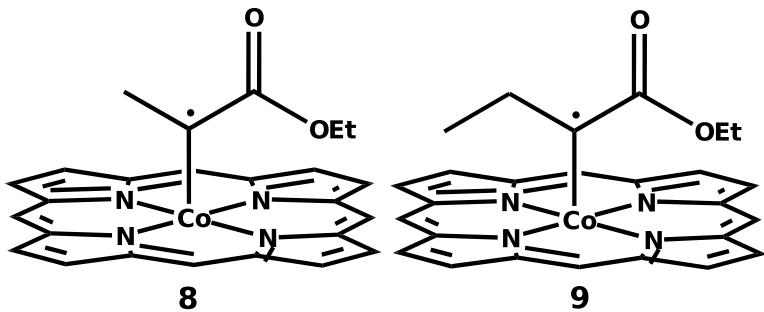


Figure 5.4: Depiction of complexes **8** & **9**.

Complexes **1-6** all exhibit spin densities largely localized on the carbene carbon, in a nearly pure p_y orbital (where the z axis is defined by the metal-carbene bond), with little or no cobalt radical character; this is illustrated in Figure 5.5. The NBO analysis indicates that the radical is stabilized by backdonation to an adjacent π^* orbital on the carbene substituent. This π^* antibonding orbital resides on a carbonyl or sulfonyl group (**1-5**). The carbene carbon spin density varies from 0.649-0.931 (e) but does not obviously correlate with the other structural or electronic properties. However, coupling this property to changes in carbene carbon and cobalt philicity, *via* modification of carbene substituents, may provide a straightforward mechanism for “tuning” selectivity. For example, % yield as a function of electron deficient olefins reveals the following trend: **1** >**2** >**3**.¹¹⁶⁻¹¹⁸ This correlates well with the charge and spin density values calculated for these compounds as shown in Table 5.1, suggesting that the ability to control both charge and radical localization can be a powerful tool for rational catalyst/intermediate design.

Another complex, **7**, having two highly electron-withdrawing fluorine groups attached to the carbene carbon, exhibits significantly different chemical character (Table 5.1). This is confirmed with NBO analysis, which shows that the carbene substituent π^* orbitals, which are responsible for back-stabilization, do not exist in the $-CF_2$ moiety. This leads to radical “migration” from the carbene carbon to the $d_{x^2-y^2}$ orbital on the metal, making it, in a sense, more Fischer-like with an empty carbene p_y orbital and shared electrons between the metal d_{z^2} and carbene sp^2 orbitals (Figure 5.6). This is further confirmed by examining both partial charges and bond orders with the latter indicative of single bond character as is the case in prototypical Fischer

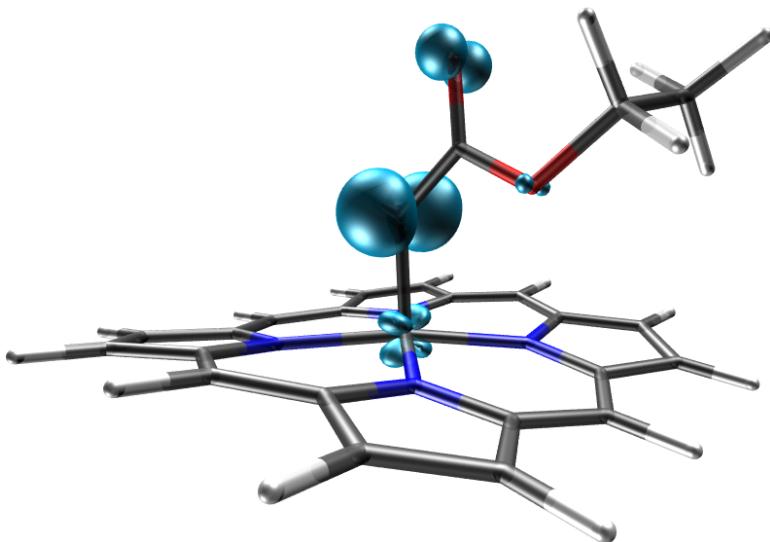


Figure 5.5: Plot of the positive spin density of the $[\text{Co}(\text{Por})]\text{-CHCO}_2\text{Et}$ complex. As can be seen by the magnitude of the isosurface (isovalue = 0.013), the excess spin density is almost entirely localized on the carbene carbon in a nearly pure p atomic orbital in nature.

complexes. As a result of the carbene carbon's slight electrophilic character, it is predicted that **7** should have higher reactivity toward aliphatic and other electron-rich olefins if experimentally accessible.

To further investigate the nature of the carbene complex, CDA was performed on an amphiphilic series of compounds with results reported for **1**, **7**, and representative Fischer- (**FC**), and Schrock-type (**SC**) carbenes (Table 5.2). CDA provides a quantitative representation of the Dewar-Chatt-Duncanson model formalism. The donation and backdonation contributions to the bonding picture are calculated based upon occupied and unoccupied orbital overlap.

In addition to CDA donation and backdonation, the repulsive polarization of the electron density in the vicinity of the bond as well as the rest term (Δ)¹⁷¹ are assessed. The rest term is indicative of the metal-carbene bond character with deviations from zero signifying the nature of covalency. For example, the metal-carbene bond in **FC**, which NBO shows to be slightly less than single, has a Δ value of 0.004 whereas the **SC** complex is essentially half-way between a single and double bond and as expected has a larger rest term (0.165). Concerning individual donation (d) and backdonation (b) values, the physically meaningful quantity, especially for Fischer carbenes, is the d/b ratio as it illustrates the amount of backdonation present in the

Table 5.2: CDA analysis for the [Co(Por)]-CHCO₂Et (**1**), Tungsten (W) Fischer carbene and Titanium (Ti) Schrock carbene complexes. Values shown are donation/backdonation ratio, repulsive charge polarization and rest term (Δ). Significant deviations of Δ from zero are indicative of covalent bonding character.

| Label | Complex | d/b | r | Δ |
|-----------|--|-------|--------|----------|
| FC | (CO) ₅ W-C(CH ₃)OCH ₃ | 2.28 | -0.242 | 0.004 |
| SC | (C ₅ H ₅) ₂ Ti-CH ₂ | 3.71 | -0.294 | 0.165 |
| 1 | [Co(Por)]-CHCO ₂ Et | 2.61 | 0.079 | 0.024 |
| 7 | [Co(Por)]-CF ₂ | 1.94 | -0.182 | 0.010 |

bonding of the complex. Note, attaching physical significance to the d/b ratio makes sense only with small rest term values.¹⁷⁰

CDA results indicate that **1** has a Fischer-like metal-carbene bond with backdonation from the metal to the carbene, stabilizing the complex, however charge and NBO analyses clearly demonstrate non-Fischer character as evident by a nucleophilic carbene carbon. Further, from the nearly zero rest term, there is little covalent character in this complex as opposed to Schrock-types. **7** is even more Fischer-like with a repulsive polarization term, r , that is negative, whereas **1** has an r value that is nearly zero with a smaller Δ rest term.

NBO analysis further reveals the nature of **1**. As noted above, it was determined that a highly localized carbon centered radical exists, which is nearly pure p_y in character (Figure 5.5). σ donation occurs between the sp^2 orbital of the carbene into the d_{z^2} orbital on the metal. Additionally, NBO analysis shows that the π^* orbital on the adjacent carbonyl stabilizes the radical by accepting electron density, thus facilitating favorable backdonation from the metal's d_{xz} and d_{yz} orbitals into the antibonding orbital of the metal-carbene bond. This qualitative bonding picture is illustrated in Figure 5.6. Again, **7** serves as a counterexample with no accessible π^* orbital acceptors.

The necessity of the π^* acceptor (*e.g.* the diazo acetate carbonyl on **1**) was examined using a [Co(Por)]-carbene substituted with an additional electron-withdrawing group (*i.e.* -NO₂, **3**). The presence of this additional group had profound effects on the structure and electronics of the complex. Surprisingly, the carbonyl group, which typically acts as a radical density acceptor,

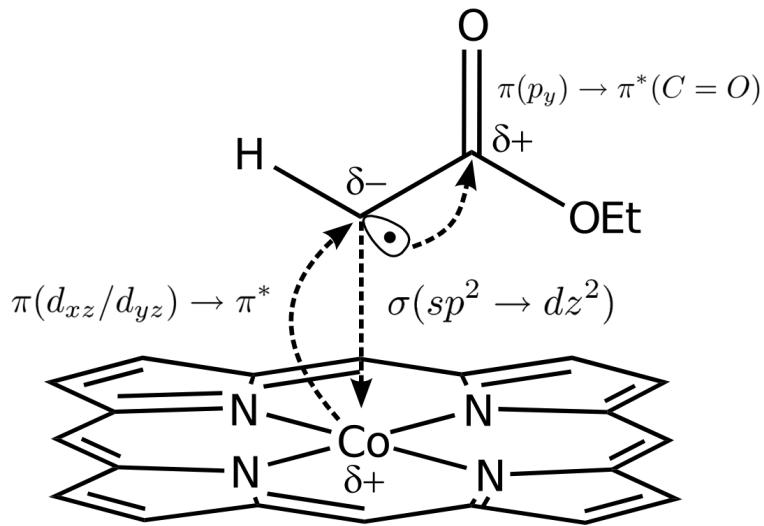


Figure 5.6: Molecular orbital diagram of the cobalt-porphyrin in complex with the carbene ligand $-\text{CHCO}_2\text{Et}$ (complex **1**). The complex is formed through a Fischer-like mechanism in which σ donation occurs between the sp^2 molecular orbital of the ligand and the singly-occupied dz^2 atomic orbital (where the z axis is defined perpendicular to the plane of the porphyrin ring) on the Co(II), with stabilization taking place through a backdonation from the π -symmetry orbitals of the cobalt into the Co-C antibonding molecular orbital (and consequently yielding a bond order less than unity, as is typical in Fischer carbenes). This leaves significant excess electron density (*i.e.* the radical) on the carbene carbon, which is of nearly pure (>99 %) p_y state. Stabilization of the carbene radical occurs through partial relief of the excess electron density *via* donation from the p_y orbital into an adjacent π^* antibonding orbital on the carbonyl (other good π^* acceptors, such as sulfonyl groups, can also fulfill this role).

(when geometry optimized) rotated nearly perpendicular to the plane of the radical with virtually no spin density located on the oxygen ($e \approx 0.0369$). In contrast, the nitro group's π^* orbitals aligned well with the p_y orbital of the radical allowing the NO_2 oxygens to accept 0.269 e of spin density.

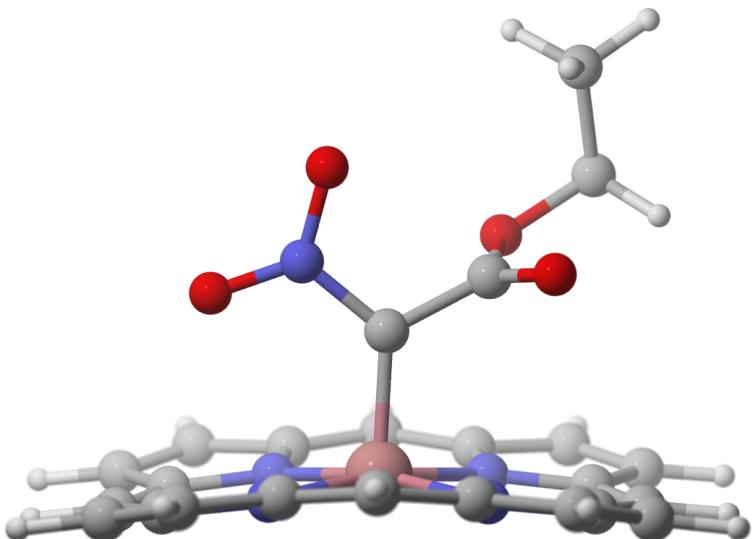
While examining the electronic and steric effects of this rearrangement it was found that at least two minima exist for this structure; the global minimum, with the carbonyl group twisted out of the radical plane (p_y orbital) as shown in Figure 5.7(a), and a second structure with the carbonyl in-plane, but the nitro group rotated perpendicular (Figure 5.7(b)). To understand the electronic and steric preferences of these two conformers, a constrained optimization was performed keeping both the carbonyl $\text{C}=\text{O}$ and the nitro $\text{N}=\text{O}$ in plane with the carbene p_y orbital. Examination of radical stabilization *via* NBO perturbation analysis revealed that an

approximate 4 kcal/mol preference is given to the nitro group as compared to the carbonyl group. It follows that this interaction is favored and the carbonyl group rotates out-of-plane keeping the p_y – π^* interaction intact. However, the total energy difference between these structures was found to be approximately 1 kcal/mol, again with the twisted C=O structure being more stable.

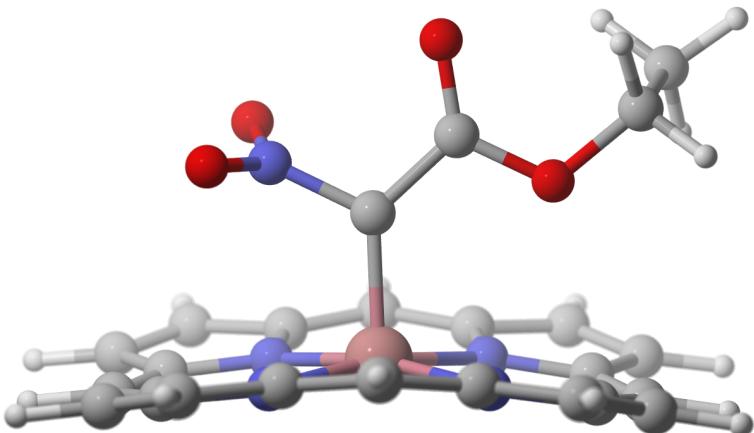
From this analysis interesting hypotheses connecting experimental reactivities can be formulated. Compound **3** has been shown to be the least reactive of the three catalytic intermediates examined (**1** > **2** > **3**) toward electron-deficient olefins. We hypothesize that the reactivity of this species is dependent upon the orientation of the carbonyl group. For example, the more reactive intermediates have their carbonyl group in the plane of the radical p_y orbital and it has been experimentally shown that **3** reacts with a yield of about 65% with appropriate substrates.¹¹⁸ We propose that the reaction proceeds by first converting from the state with the C=O out-of-plane to the slightly higher energy state where the NO₂ is out-of-plane (see Figure 5.7(b)). This transition serves to both position the carbonyl group correctly and simultaneously allows for the necessary electronic properties (*i.e.* radical stabilization by the π acceptor on the carbonyl group, achieving a more nucleophilic carbene carbon, *etc.*). It is not known if this transition is dominated by steric or electronic effects, but from our analysis it seems that the electronics should be the dominant factor in controlling this transition and subsequent reactivity. Using this principle as a strategy for the rational design of selective catalytic intermediates is a logical next step.

5.6 Conclusions

The electronic structure calculations and subsequent analyses presented identify and characterize a new class of metal-carbene complexes that are key intermediates in catalytic cyclopropanation by [Co(Por)]. The bonding of these intermediates is reminiscent of Fischer carbenes, with the metal existing in a low oxidation state and metal-carbene bond order being described as slightly less than one. However, the carbene α -carbon is both “tunable” and nucleophilic which is more



(a) carbonyl out-of-plane



(b) nitro out-of-plane

Figure 5.7: Depiction of two low energy conformers of **3**. The global minimum structure, (a), is shown with its carbonyl group ($-C=O$) rotated nearly perpendicular relative to the carbene p_y orbital, while the nitro ($-NO_2$) group is in-plane. A slightly higher energy structure, (b), has its nitro group rotated out-of-plane with respect to the p_y orbital and the carbonyl aligned.

reminiscent of Schrock-type complexes. Additionally, it is found that significant radical character is localized on the carbene carbon.

Calculated properties correlate well with experimental observations; reactivity trends strongly suggested a localized radical intermediate that readily reacts with electrophilic substrates. The partial charge of the carbene carbon is found to take on values from slightly positive to approximately -0.5 e , again suggesting this can be used to “tune” reactivity toward different olefin substrates. For example, the carbene complex with an approximately neutral carbene carbon may serve as an effective reactive intermediate in catalytic cyclopropanation of less electrophilic substrates *via* this type of radical reaction mechanism. Further evidence of this tunability is demonstrated by alternating alkyl chain length of the carbene substituents which not only controls the charge of the radical carbon, but also stabilizes it *via* hyperconjugation.

The radical carbene complex is found to be critically stabilized by either backdonation into a π^* acceptor on the carbene substituents, or through hyperconjugation. Indeed, a complex such as $[\text{Co}(\text{Por})]\text{-CF}_2$, which lacks available orbital acceptors, exhibits a very different bonding pattern with a radical largely localized on the cobalt metal center and Fischer-like characteristics that include an electrophilic carbene carbon and similar backdonation characteristics as judged by CDA analysis.

While the present catalytic mechanism involves a doublet $[\text{Co}(\text{Por})]$ -catalyst reacting with ground state triplet carbenes, the present results suggest that open-shell singlet carbenes, where the radical electrons are spin-paired but in spatially distinct locations, may also be catalytically accessible; this is hypothesized to be a fundamentally different mechanism than that of current Fischer and Schrock carbenes. Interestingly, such open-shell singlet states may be photoaccessible under certain reaction conditions. Additionally, an explanation of the experimental reactivity of intermediate **3** is offered. We hypothesize that reactivity is controlled by conformational changes associated with the adjacent carbene substituent; these conformational changes are in turn controlled by electronic stabilization of the localized carbene radical into either the carbonyl or nitro group. In fact, it is found that the in-plane nitro group (with the carbonyl rotated out-of-plane) preferentially stabilizes the radical and is therefore preferred ($\Delta E \approx 1.46\text{ kcal/mol}$).

The reaction is thus concluded to proceed only after conformational changes align the carbonyl group back in-plane with the radical. This structural feature resembles the carbonyl and sulfonyl moieties on **1** and **2** which are known to possess larger yields and higher reactivities.

In summary, the nature of the key intermediate for [Co(Por)]-catalyzed cyclopropanation was explored *via* electronic structure calculations and subsequent analysis. The results also permit the prediction and design of novel catalytic carbene intermediates that can be “tuned” to achieve desired reactivity. Future studies will include attempts to synthesize novel [Co(Por)]-catalysts and to confirm the predicted reactivity (*e.g.* a catalytic system involving intermediate **5** should be comparable to that of **1** in reactivity) based on the calculated properties.

5.7 Acknowledgments

HLW3 would like to acknowledge NIH (1K22HL088341-01A1) and the University of South Florida (start-up) for funding. Computations were performed on the NCSA Cobalt supercomputer under a Teragrid Grant (Grant No. TG-DMR090028) to B.S., and also at the USF Research Computing Center where NSF-funded computational resources (under Grant No. CHE-0722887) were greatly appreciated. BS and XPZ acknowledge funding from the U.S. Department of Energy (Grant No. DE0GG02-07ER46470), the U.S. Department of Defense (Grant No. HDTRA1-08-C-0035), the National Science Foundation (Grant No. CHE-0711024) and Draper Laboratory (Grant No. URAD2010187). The authors also thank the Space Foundation (Basic and Applied Research) for partial support.

5.8 Supporting Information

Details of the electronic structure calculations, computational controls, table of properties and coordinates for several complexes. This material is available free of charge via the Internet at <http://pubs.acs.org>.

Chapter 6

Virtually Engineering Dual Detection & Decontamination Materials for Chemical Warfare Agents: Computational Insights

6.1 Introduction

Chemical warfare agents (CWAs) (nerve, blister, choking, etc., a few of which are illustrated in Figure 6.1) and other toxic industrial chemicals (TICs), e.g. ammonia, chlorine, fluorine, etc., pose a significant threat to both civilian and military populations. Ensuring the safety of those within close proximity of exposure zones is therefore critical. Due to the ongoing threat of CWAs and omnipresence of TICs, predicting an act of war or accidental exposure event is oftentimes difficult. Thus having a readily exercisable strategy in place for air purification and, ideally, chemical degradation of such threats is of high importance. Currently this is achieved by the use of a gas mask equipped with the proper media to separate the airflow and contain any harmful chemicals. The type of media may vary, and can be selective for a specific type of threat. Engineering enhanced filtration materials can be challenging, however, as oftentimes the chemical identity at an exposure site is unknown. Two very common masks widely employed by the U.S. military are the M40 (for the U.S. Army and U.S. Marine Corps) and the MCU-2A/P (for the U.S. Air Force and U.S. Navy). Both are capable of providing limited protection against a broad range of chemical and biological warfare agents, and thus should only be used as escape devices rather than for prolonged protection.¹⁹⁶ Clearly, there exists a tradeoff between designing a material to be highly selective towards a single target and designing a material to remove a broad range of chemical threats, with less protection against specific targets.¹⁹⁷

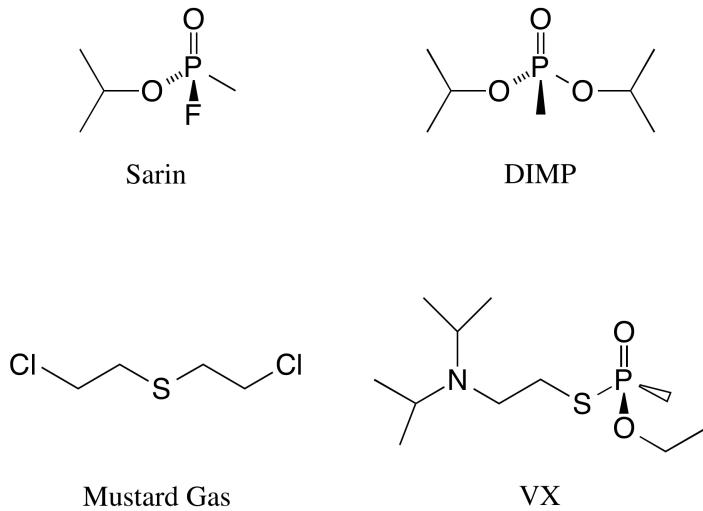


Figure 6.1: Structures of (**top**) the nerve agent sarin, alongside its simulant, diisopropyl methylphosphonate (DIMP) and (**bottom**) two additional chemical warfare agents, bis(2-chloroethyl) sulfide (mustard gas) and VX. Only sarin and DIMP are studied in this work.

In this work, a series of Zn(II) metallated derivatized di- and tetraphenyl porphyrins (Figure 6.2) are virtually screened for their ability to (quantitatively and selectively) bind the nerve agent sarin as well as its simulant, diisopropyl methylphosphonate (DIMP). The calculations performed offer insight into the nature of binding occurring between sensor and targets. Further, *in silico* chemical modifications of one of the sensors in particular are made and in turn successfully increase the binding affinity while simultaneously reversing the trend of specificity for a target of interest, thus demonstrating the role that simulation can play in CWA sensing and containment.

The efforts herein focus on engineering a molecular sensor to more specifically identify and selectively capture a target threat agent, in specific sarin versus DIMP. As a practical application, the synthesized sensors can be strategically integrated into porous materials, such as metal–organic frameworks (MOFs), whereby CWA/TIC encapsulation can occur. Further equipping such a smart material with on-board photocatalytic degradation capabilities is possible, and of course desirable.

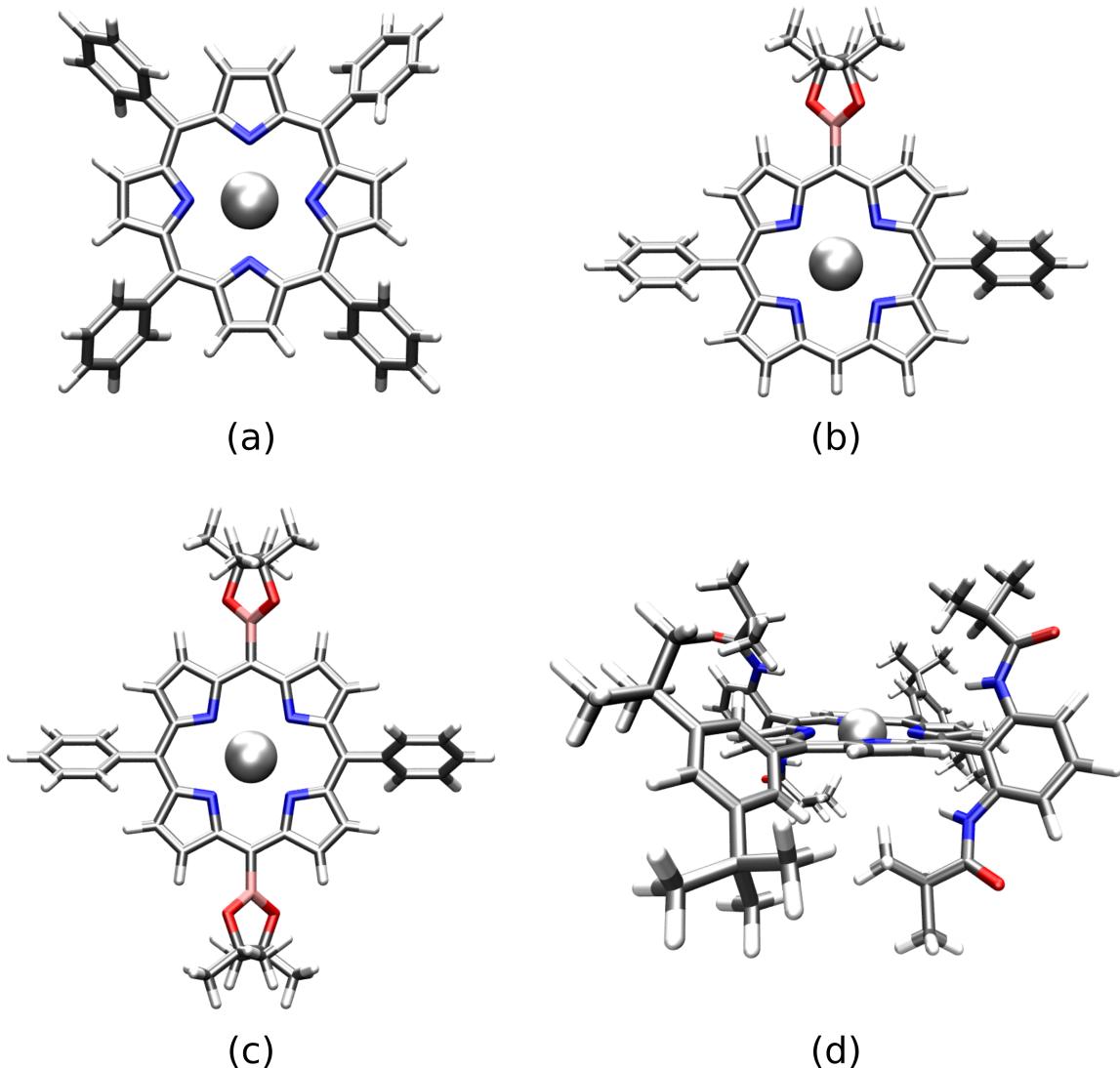


Figure 6.2: Rendered structures of the $[Zn(\text{Por})]$ -based sensors investigated: (a) ZnTPP, (b) ZnMBDPP, (c) ZnDBDPP, and (d) ZnCS1. The atomic color map is as follows: B = Pink, C = Gray, H = White, N = Blue, O = Red, Zn = Silver.

6.2 Methods

6.2.1 Minimization and Complex Formation

The sensors employed in this investigation are derivatized di- and tetraphenyl porphyrins, and are depicted in Figure 6.2. In order of increasing structural complexity, the four sensors used are **(a)** tetraphenylporphyrin (TPP), **(b)** 10-monoborano-5,15-diphenylporphyrin (MBDPP), **(c)** 5,10-diborano-15,20-diphenylporphyrin (DBDPP), and **(d)** 3,5-di^tBu-IbuPhyrin, (CS1). Similarly, the target binding agents are sarin and its simulant DIMP (see Figure 6.1). Initial geometry optimizations of each sensor and target were performed independently in vacuum via restricted Kohn-Sham density functional theory (DFT),¹⁸⁶ using the quantum chemistry package Q-Chem (version 3.2).¹⁸⁷ The M06-L exchange-correlation functional of Truhlar,¹⁷⁵ and a mixed basis set approach was used, applying the LANL2TZ basis functions and effective core potential (ECP)¹⁸⁹ to the zinc center of the metalloporphyrin and treating all remaining atoms with the 6-31G* basis set.¹⁹⁰ Minimizations first made use of a coarse (50-194) standard grid¹⁹¹ (SG-1), followed by a finer grid (96-302).

To simulate binding of the target to the sensor, the target was initially placed in a random orientation at a distance between 4-8 Å orthogonal to the plane of the porphyrin ring. A minimum of three initial target orientations were used, to increase the spatial trajectories explored and to ensure a global minimum was reached. In all cases (each combination of sensor with target), the target explored a trajectory which terminated with the target's phosphoryl oxygen weakly interacting with the Zn metal center of the porphyrin. The expression for computing the binding energy is

$$E_{Bind} = E_{Complex} - (E_{Sensor}^{r=\infty} + E_{Target}^{r=\infty}) \quad (6.1)$$

That is, the energy of binding is calculated as the difference between the final energy of the optimized complex and the sum of the individual monomers, geometry optimized at infinite separation. The binding energies of all the complexes are provided in Table 6.1.

Table 6.1: Calculated binding energies (in $\text{kcal}\cdot\text{mol}^{-1}$) of the sensor/target complexes.

| Sensor | Target | Binding Energy |
|---------|--------|----------------|
| ZnTPP | DIMP | -26.36 |
| | Sarin | -19.33 |
| ZnMBDPP | DIMP | -25.65 |
| | Sarin | -16.41 |
| ZnDBDPP | DIMP | -22.27 |
| | Sarin | -21.55 |
| ZnCS1 | DIMP | -31.22 |
| | Sarin | -27.76 |

6.2.2 Sensor Engineering

The trend observed in Table 6.1 indicates that DIMP is the preferred target for each of the four [Zn(Por)] sensors. It is desirable, however, for the sensor to preferentially complex with the nerve agent sarin. Of the four sensors, the platform with the most functionalized peripheral ligands by far is ZnCS1. Using this structure as a template, *in silico* modifications of the decorated ligands were made. The exact chemical mutations are illustrated in Figure 6.3. Attempts were made to maximize further stabilization between the sarin molecule and the tetraphenyl ligands of ZnCS1 (based on the orientation of sarin in the optimized complex), while minimizing the free volume in the binding space above the Zn core (at the fifth coordination site) since sarin occupies less volume than its simulant DIMP.

Post-modification geometry optimization of the sensor with both DIMP and sarin revealed interesting results. Energetically, the sensor now selectively binds sarin, effectively reversing the trend of DIMP preference (see Table 6.2). Further, the modifications were successful in stabilizing the sarin, as a new $\text{NH} \cdots \text{F}$ weak bond is realized (illustrated in Figure 6.4). The interior space at the binding site is now crowded enough to disrupt the typical phosphoryl oxygen–Zn interaction, thus resulting in a less favorable energetic interaction for DIMP (shown in Figure 6.5).

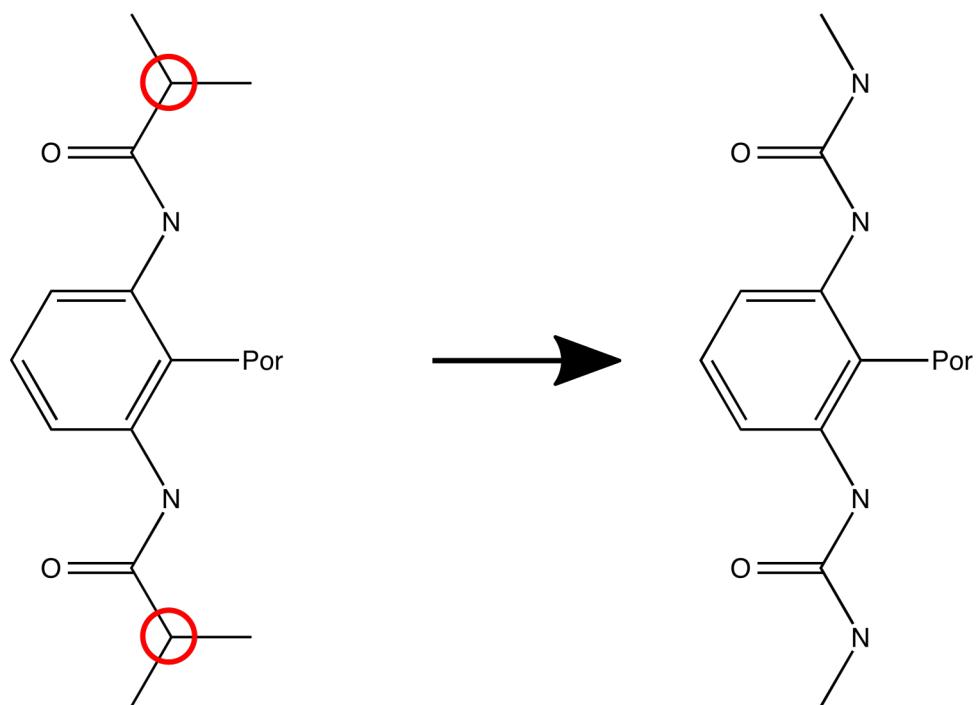
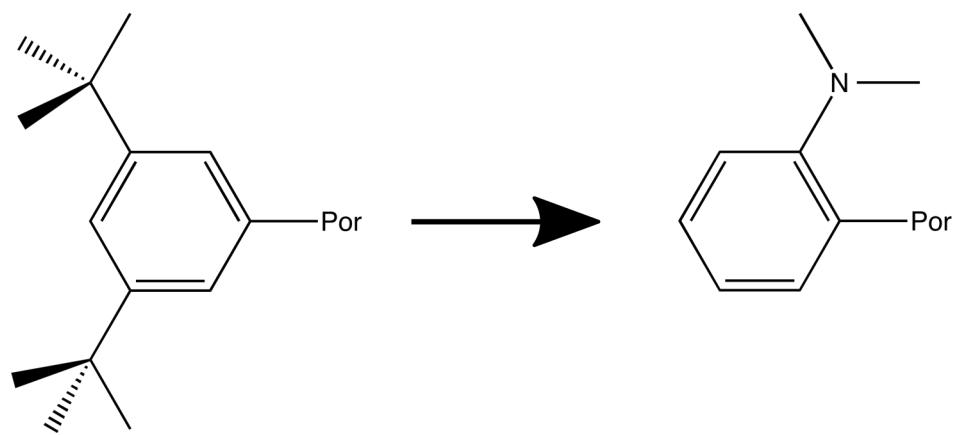


Figure 6.3: The modifications made to both types of functionalized phenyl substituents. **(top)** The isopropyl substituents are removed from the 3- and 5-positions of the phenyl rings, and an amino group is inserted at the 2-position. This further crowds the free volume available at the Zn coordination site. **(bottom)** The carbon atom α to the carbonyl (circled in red) on each side of both phenyl rings is replaced with a nitrogen atom, thus creating a carbamide group. This allows for further potential hydrogen bonding.

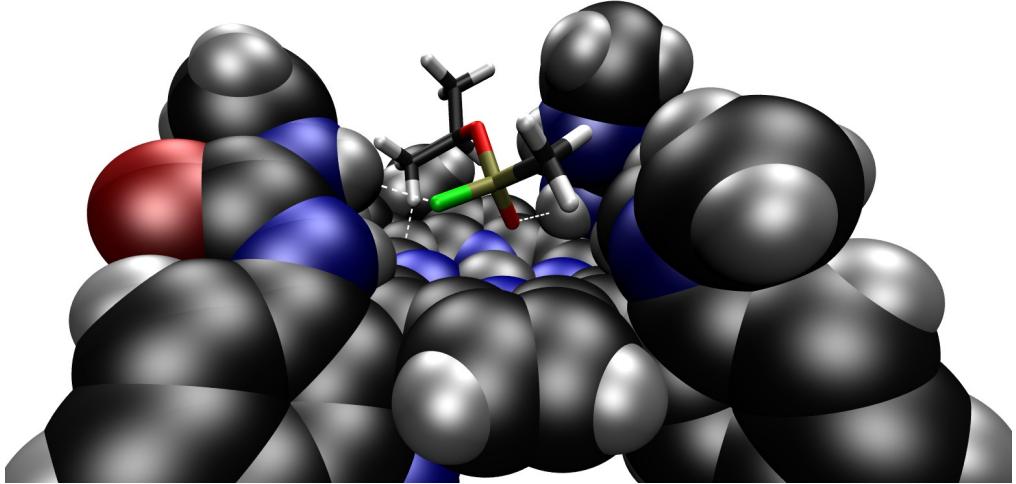


Figure 6.4: A functionalized porphyrin that has been found to tightly bind the nerve agent sarin selectively over the simulant DIMP, thus demonstrating the validity of a computational approach enabling specific binding of chemical agents. This was achieved through *in silico* mutations of the functionalized periphery so as to enhance hydrogen binding with the target (note the hydrogen bonds depicted above by the dashed lines). Calculations were performed using the M06-L exchange correlation functional.¹⁷⁵

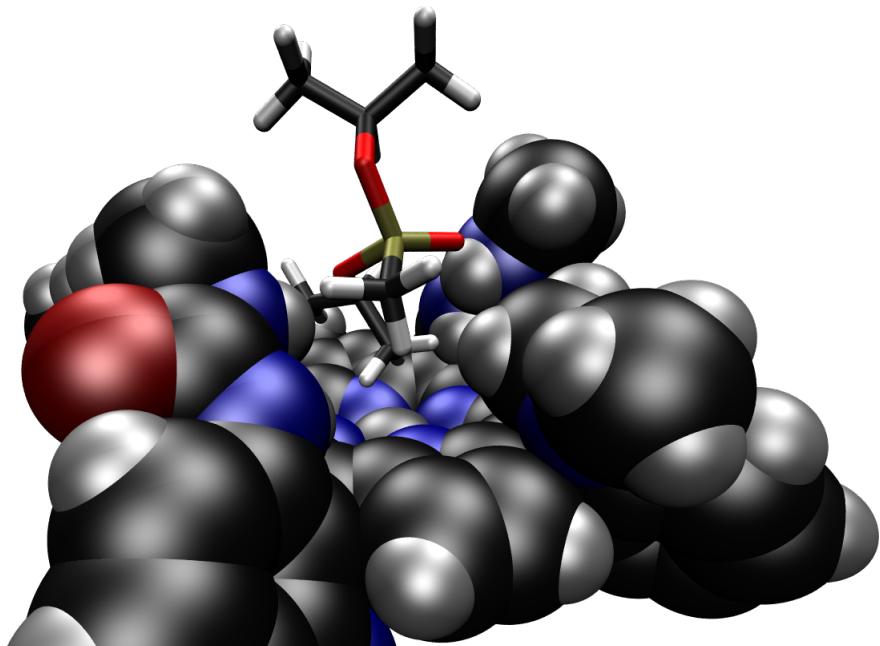


Figure 6.5: The newly crowded space at the distal position of the Zn center of the modified ZnCS1 sensor does not allow for interaction between the phosphoryl oxygen of DIMP to interact with the Zn center, thus resulting in a sensor which exhibits specificity for sarin over DIMP.

Table 6.2: Calculated binding energies (in $\text{kcal}\cdot\text{mol}^{-1}$) of sarin and DIMP with the modified ZnCS1 sensor. Virtual modification of the platform proved successful at reversing the preferential DIMP binding that was observed in Table 6.1.

| Sensor | Target | Binding Energy |
|------------------|--------|----------------|
| ZnCS1 - Modified | DIMP | -14.44 |
| | Sarin | -17.42 |

6.3 Conclusions and Future Directions

Four Zn(II) derivatized di- and tetraphenyl porphyrins have been virtually screened for binding with the chemical warfare (nerve) agent sarin and its simulant DIMP. Note that the experimental results (for DIMP only) have been reported.¹⁹⁸ Calculated binding energies revealed a trend of preferential DIMP binding. *In silico* chemical modifications to one of the sensors, ZnCS1, achieved reversal of said trend, demonstrating the role that theory can play in CWA (and more broadly CBRN – chemical, biological, radiological and nuclear) detection studies. In addition to the biological (i.e. toxicity) advantages, molecular simulation investigations are more cost effective than experiments, which demand clean rooms (at, for example, Edgewood Chemical and Biological Center – ECBC) and paid security personnel.

Future work should encompass the screening of additional CWAs such as sulfur mustard, VX, tabun, soman, etc., as well as target TICs. The sensors can each be engineered for preferential binding with a target of interest. Further, the MOFs that shall be used to house the molecular sensors can themselves be rationally designed for enhanced CWA/TIC stabilization, leading to superior materials with excellent sequestering and photocatalytic degradation abilities. Efforts by others are already underway to design such “smart” materials.^{199–204}

6.4 Acknowledgements

I would like to thank the Defense Threat Reduction Agency and the Charles Stark Draper Laboratory for supporting this research through an award (HDTRA1-08-C-0035) granted to the University of South Florida.

Chapter 7

Conclusions

In this dissertation, computational investigations of potential energy function development for metal–organic framework simulations, metal carbenes, and chemical warfare agents has been performed and reported in detail. The major focus of this work is the development of accurate and transferable potential energy functions for usage in materials simulations, such as sorption in MOFs. The models developed are part of the PHAST (Potentials with High Accuracy, Speed and Transferability) family, which accounts for repulsion/dispersion, electrostatic and many-body polarization contributions to the total energy. Parameterized only to very high-level ab initio energies (CCSD(T) with large basis sets extrapolated to the complete basis set limit), our models are built via a “bottom-up” approach. They have demonstrated to be robust and have predicted sorption isotherms for various sorbates in a variety of MOFs (i.e. sorbents). In particular, models for two sorbates of interest are developed and reported, that of N₂ and CH₄.

Through rigorous testing, and ultimately the realization of limitation, the original PHAST models which incorporated the Lennard-Jones 12-6 function for calculating the repulsion / dispersion energy was shown to be incapable of reproducing CCSD(T) energies for heterogeneous systems. The supplanting of the LJ function for exponential repulsion and dispersion realized classical energies in excellent quantitative agreement for both heterogeneous systems as well as homogeneous rare-gas clusters of helium atoms, especially at short pair separation distances. Even for small systems ($N \leq 7$) of noble gases (specifically, helium), the Lennard-Jones 12-6 function is inadequate, and the adoption of a “softer” function is required.

The failure of the original PHAST potential, and thus its evolution to that which is of the Tang-Toennies type, is perhaps the most impactful result of this body of work. Although often-times understressed, it is this targeted critiquing that sheds light on the boundaries of what is possible. To this end, failure is a blessing in disguise. Upon inclusion of exponential repulsions and reciprocal six, eight and ten terms for dispersion (the modeling of polarization through the use of induced dipoles and representing the permanent electrostatics as point charges being retained), the new PHAST model is demonstrated to be more applicable to both homogeneous and heterogeneous systems. Given this, the future of model development and thus the simulation of sorption in nanomaterials remains bright. Coupled with the explosive growth of supercomputing (in both hardware and software applications), and the ultimate stride towards exascale computing, it is an extremely exciting time for data science and thus nanomaterials simulation. Using highly accurate classical potentials, a large number of porous materials can be scanned in a high-throughput fashion for their ability to selectively filter and retain targets of interest, which was simply not possible five to ten years ago.

Using simulation to guide experiment as applied to materials engineering is a relatively new paradigm that is sure to continually grow. Made possible by the intricate potential energy function development, it is this attention to detail and stress testing (verification and validation) of the models to which credit is due. This is important, as blind application of the inadequate models can lead to mistakenly correct results (this is proven in Chapter 3 where multiple parameter sets lead to seemingly similar results, yet not all are adequate).

In addition to the extensive model development presented, the computational characterization of a new class of radical metal–carbene complex was performed. Here, a cobalt(II)–porphyrin complex, [Co(Por)], was investigated to understand its role as an effective catalyst in stereoselective cyclopropanation of a diazoacetate reagent. Density functional theory along with natural bond order analysis and charge decomposition analysis gave insight into the electronics of the catalytic intermediate. The bonding pattern unveiled a new class of radical metal–carbene complex, with a doublet cobalt into which a triplet carbene σ donates, and subsequent back-bonding

occurs into a π^* antibonding orbital. This is a different type of interaction not seen in the three existing classes of metal–carbene complexes, namely Fischer, Schrock, and Grubbs.

Through a series of functional group replacement adjacent to the carbene α -carbon, it was shown that the presence of a neighboring π^* acceptor (such as a carbonyl group, C=O) is necessary to localize the spin density and allow stabilization of the radical. In fact, when not present, radical migration down into the electron-rich porphyrin macrocycle was observed. When this is the case, the catalytic intermediate is much less effective which in turn leads to a much lower yield in the cyclopropanation reaction.

Lastly, the virtual engineering of chemical warfare agent sensors was investigated. This work was supported by the U.S. Department of Defense (via DTRA and Draper Laboratory). *In silico* chemical modifications to a previously synthesized zinc–porphyrin, ZnCS1, were made to attempt to achieve preferential binding of the nerve agent sarin versus its simulant, DIMP (diisopropyl methylphosphonate). Prior to modification, DIMP was the preferred binding agent. Post-modification, a combination of steric effects and induced hydrogen bonding allowed for the selective binding of sarin. The success of this work demonstrates the role that high performance computing can play in national security research, without the associated costs and high security required for experimentation.

Lastly, the virtual engineering of chemical warfare agent sensors was investigated. This work was supported by the U.S. Department of Defense (via DTRA and Draper Laboratory). *In silico* chemical modifications to a previously synthesized zinc–porphyrin, ZnCS1, were made to attempt to achieve preferential binding of the nerve agent sarin versus its simulant, DIMP (diisopropyl methylphosphonate). Prior to modification, DIMP was the preferred binding agent. Post-modification, a combination of steric effects and induced hydrogen bonding allowed for the selective binding of sarin. The success of this work demonstrates the role that high performance computing can play in national security research, without the associated costs and high security required for experimentation.

References

- [1] Snurr, R. Q.; Hupp, J. T.; Nguyen, S. T. Prospects for nanoporous metal-organic materials in advanced separations processes. *AICHE Journal* **2004**, *50*, 1090–1095.
- [2] Delgado, J. A.; Uguina, M. A.; Gómez, J. M.; Ortega, L. Adsorption equilibrium of carbon dioxide, methane and nitrogen onto Na- and H-mordenite at high pressures. *Separation and Purification Technology* **2006**, *48*, 223 – 228.
- [3] Moulton, B.; Zaworotko, M. J. From molecules to crystal engineering: supramolecular isomerism and polymorphism in network solids. *Chem. Rev.* **2001**, *101*, 1629–1658.
- [4] Farha, O. K.; zgr A. Yazaydn,; Eryazici, I.; Malliakas, C. D.; Hauser, B. G.; Kanatzidis, M. G.; Nguyen, S. T.; Snurr, R. Q.; Hupp, J. T. De novo synthesis of a met-alorganic framework material featuring ultrahigh surface area and gas storage capacities. *Nature Chemistry* **2010**, *2*, 944–948.
- [5] Eddaoudi, M.; Moler, D. B.; Li, H.; Chen, B.; Reineke, T. M.; O'Keeffe, M.; Yaghi, O. M. Modular Chemistry: Secondary Building Units as a Basis for the Design of Highly Porous and Robust Metal-Organic Carboxylate Frameworks. *Accounts of Chemical Research* **2001**, *34*, 319–330.
- [6] Zheng, B.; Bai, J.; Duan, J.; Wojtas, L.; Zaworotko, M. J. Enhanced CO₂ Binding Affinity of a High-Uptake rht-Type Metal Organic Framework Decorated with Acylamide Groups. *Journal of the American Chemical Society* **2011**, *133*, 748–751.

- [7] Mohamed, M. H.; Elsaidi, S. K.; Wojtas, L.; Pham, T.; Forrest, K. A.; Tudor, B.; Space, B.; Zaworotko, M. J. Highly Selective CO₂ Uptake in Uninodal 6-Connected mmo Nets Based upon MO42 (M = Cr, Mo) Pillars. *Journal of the American Chemical Society* **2012**, *134*, 19556–19559.
- [8] Nugent, P.; Belmabkhout, Y.; Burd, S.; Cairns, A.; Luebke, R.; Forrest, K.; Pham, T.; Ma, S.; Space, B.; Wojtas, L.; Eddaoudi, M.; Zaworotko, M. Porous materials with optimal adsorption thermodynamics and kinetics for CO₂ separations. *Nature* **2013**, *495*, 80–84.
- [9] Britt, D.; Furukawa, H.; Wang, B.; Glover, T. G.; Yaghi, O. M. Highly efficient separation of carbon dioxide by a metal-organic framework replete with open metal sites. *Proceedings of the National Academy of Sciences* **2009**, *106*, 20637–20640.
- [10] Millward, A. R.; Yaghi, O. M. MetalOrganic Frameworks with Exceptionally High Capacity for Storage of Carbon Dioxide at Room Temperature. *Journal of the American Chemical Society* **2005**, *127*, 17998–17999.
- [11] Banerjee, R.; Phan, A.; Wang, B.; Knobler, C.; Furukawa, H.; O'Keeffe, M.; Yaghi, O. M. High-Throughput Synthesis of Zeolitic Imidazolate Frameworks and Application to CO₂ Capture. *Science* **2008**, *319*, 939–943.
- [12] Snurr, R. Q.; Yazaydin, A. O.; Dubbeldam, D.; Frost, H. *Metal-Organic Frameworks*; John Wiley & Sons, Inc., 2010; pp 313–339.
- [13] Bock, S.; Bich, E.; Vogel, E. A new intermolecular potential energy surface for carbon dioxide from ab initio calculations. *Chemical Physics* **2000**, *257*, 147 – 156.
- [14] Hellmann, R.; Bich, E.; Vogel, E. Ab initio intermolecular potential energy surface and second pressure virial coefficients of methane. *The Journal of Chemical Physics* **2008**, *128*, 214303.
- [15] Forrest, K. A.; Pham, T.; McLaughlin, K.; Belof, J. L.; Stern, A. C.; Zaworotko, M. J.; Space, B. *The Journal of Physical Chemistry C* **2012**, *116*, 15538–15549.

- [16] Pham, T.; Forrest, K. A.; McLaughlin, K.; Tudor, B.; Nugent, P.; Hogan, A.; Mullen, A.; Cioce, C. R.; Zaworotko, M. J.; Space, B. Theoretical Investigations of CO₂ and H₂ Sorption in an Interpenetrated Square-Pillared MetalOrganic Material. *The Journal of Physical Chemistry C* **2013**, *117*, 9970–9982.
- [17] McLaughlin, K.; Cioce, C. R.; Belof, J. L.; Space, B. A molecular H₂ potential for heterogeneous simulations including polarization and many-body van der Waals interactions. *The Journal of Chemical Physics* **2012**, *136*, 194302.
- [18] Belof, J. L.; Stern, A. C.; Space, B. An Accurate and Transferable Intermolecular Diatomic Hydrogen Potential for Condensed Phase Simulation. *Journal of Chemical Theory and Computation* **2008**, *4*, 1332–1337.
- [19] Belof, J. L.; Stern, A. C.; Space, B. A Predictive Model of Hydrogen Sorption for Metal–Organic Materials. *The Journal of Physical Chemistry C* **2009**, *113*, 9316–9320.
- [20] Mullen, A. L.; Pham, T.; Forrest, K. A.; Cioce, C. R.; McLaughlin, K.; Space, B. A Polarizable and Transferable PHAST CO₂ Potential for Materials Simulation. *Journal of Chemical Theory and Computation* **2013**, *9*, 5421–5429.
- [21] Cioce, C. R.; Tudor, B.; McLaughlin, K.; Belof, J. L.; Space, B. in preparation.
- [22] Potoff, J. J.; Siepmann, J. I. Vaporliquid equilibria of mixtures containing alkanes, carbon dioxide, and nitrogen. *AIChE Journal* **2001**, *47*, 1676–1682.
- [23] Liu, Y.; Panagiotopoulos, A. Z.; Debenedetti, P. G. Monte Carlo Simulations of High-Pressure Phase Equilibria of CO₂H₂O Mixtures. *The Journal of Physical Chemistry B* **2011**, *115*, 6629–6635.
- [24] Yu, K.; McDaniel, J. G.; Schmidt, J. R. Physically Motivated, Robust, ab Initio Force Fields for CO₂ and N₂. *The Journal of Physical Chemistry B* **2011**, *115*, 10054–10063.
- [25] Murthy, C.; O’Shea, S.; McDonald, I. Electrostatic interactions in molecular crystals. *Molecular Physics* **1983**, *50*, 531–541.

- [26] Huber, K. P.; Herzberg, G. *Molecular Spectra and Molecular Structure. IV. Constants of Diatomic Molecules*; Van Nostrand Reinhold Company: New York, NY, 1979.
- [27] NIST Computational Chemistry Comparison and Benchmark Database, NIST Standard Reference Database Number 101. <http://cccbdb.nist.gov/>, Release 15b, August 2011, Editor: Russell D. Johnson III.
- [28] Mankoo, P. K.; Keyes, T. Classical Molecular Electrostatics: Recognition of Ligands in Proteins and the Vibrational Stark Effect. *The Journal of Physical Chemistry B* **2006**, *110*, 25074–25079.
- [29] Keyes, T.; Napoleon, R. L. Extending Classical Molecular Theory with Polarization. *The Journal of Physical Chemistry B* **2011**, *115*, 522–531.
- [30] Diep, P.; Johnson, J. K. An accurate H₂–H₂ interaction potential from first principles. *The Journal of Chemical Physics* **2000**, *112*, 4465–4473.
- [31] Aquilanti, V.; Bartolomei, M.; Cappelletti, D.; Carmona-Novillo, E.; Pirani, F. The N₂–N₂ system: An experimental potential energy surface and calculated rotovibrational levels of the molecular nitrogen dimer. *The Journal of Chemical Physics* **2002**, *117*, 615–627.
- [32] others,, et al. MOLPRO, version 2010.1, a package of ab initio programs. 2010; see <http://www.molpro.net>.
- [33] Raghavachari, K.; Trucks, G. W.; Pople, J. A.; Head-Gordon, M. A fifth-order perturbation comparison of electron correlation theories. *Chemical Physics Letters* **1989**, *157*, 479 – 483.
- [34] Thom H. Dunning, J. Gaussian basis sets for use in correlated molecular calculations. I. The atoms boron through neon and hydrogen. *The Journal of Chemical Physics* **1989**, *90*, 1007–1023.

- [35] Kendall, R. A.; Thom H. Dunning, J.; Harrison, R. J. Electron affinities of the first-row atoms revisited. Systematic basis sets and wave functions. *The Journal of Chemical Physics* **1992**, *96*, 6796–6806.
- [36] Boys, S.; Bernardi, F. The calculation of small molecular interactions by the differences of separate total energies. Some procedures with reduced errors. *Molecular Physics* **1970**, *19*, 553–566.
- [37] Huh, S. B.; Lee, J. S. Basis set and correlation dependent extrapolation of correlation energy. *The Journal of Chemical Physics* **2003**, *118*, 3035–3042.
- [38] Babin, V.; Medders, G. R.; Paesani, F. Toward a Universal Water Model: First Principles Simulations from the Dimer to the Liquid Phase. *The Journal of Physical Chemistry Letters* **2012**, *3*, 3765–3769.
- [39] Sundberg, K. A group-dipole interaction model of the molecular polarizability and the molecular first and second hyperpolarizabilities. *J. Chem. Phys.* **1977**, *66*, 114.
- [40] Bode, K. A.; Applequist, J. A New Optimization of Atom Polarizabilities in Halomethanes, Aldehydes, Ketones, and Amides by Way of the Atom Dipole Interaction Model. *The Journal of Physical Chemistry* **1996**, *100*, 17820–17824.
- [41] Thole, B. Molecular polarizabilities calculated with a modified dipole interaction. *Chemical Physics* **1981**, *59*, 341 – 350.
- [42] van Duijnen, P. T.; Swart, M. Molecular and Atomic Polarizabilities: Thole's Model Revisited. *The Journal of Physical Chemistry A* **1998**, *102*, 2399–2407.
- [43] Yonezawa, Y. A long-range electrostatic potential based on the Wolf method charge-neutral condition. *The Journal of Chemical Physics* **2012**, *136*, 244103.
- [44] Tudor, B.; Space, B. *Journal of Computational Science Education*, Submitted.

- [45] Valiev, M.; Bylaska, E.; Govind, N.; Kowalski, K.; Straatsma, T.; Dam, H. V.; Wang, D.; Nieplocha, J.; Apra, E.; Windus, T.; de Jong, W. NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Computer Physics Communications* **2010**, *181*, 1477 – 1489.
- [46] Cohen, H. D.; Roothaan, C. C. J. Electric Dipole Polarizability of Atoms by the Hartree–Fock Method. I. Theory for Closed-Shell Systems. *The Journal of Chemical Physics* **1965**, *43*, S34–S39.
- [47] Graham, C.; Imrie, D. A.; Raab, R. E. Measurement of the electric quadrupole moments of CO₂, CO, N₂, Cl₂ and BF₃. *Molecular Physics* **1998**, *93*, 49–56.
- [48] Huot, J.; Bose, T. K. Determination of the quadrupole moment of nitrogen from the dielectric second virial coefficient. *The Journal of Chemical Physics* **1991**, *94*, 3849–3854.
- [49] Roco, J. M. M.; Hernández, A. C.; Velasco, S.; Medina, A. Estimation of the quadrupole and hexadecapole moments of N₂ from the far-infrared spectrum of a N₂–Xe gaseous mixture. *The Journal of Chemical Physics* **1999**, *110*, 5218–5223.
- [50] Buckingham, A.; Graham, C.; Williams, J. Electric field-gradient-induced birefringence in N₂, C₂H₆, C₃H₆, Cl₂, N₂O and CH₃F. *Molecular Physics* **1983**, *49*, 703–710.
- [51] Maroulis, G. Accurate electric multipole moment, static polarizability and hyperpolarizability derivatives for N₂. *The Journal of Chemical Physics* **2003**, *118*, 2673–2687.
- [52] Buckingham, A. D. Molecular quadrupole moments. *Q. Rev. Chem. Soc.* **1959**, *13*, 183–214.
- [53] Waldman, M.; Hagler, A. New combining rules for rare gas van der waals parameters. *Journal of Computational Chemistry* **1993**, *14*, 1077–1084.
- [54] White, A. *Intermolecular Potentials of Mixed Systems: Testing the Lorentz-Berthelot Mixing Rules with Ab Initio Calculations*; 2000.

- [55] Al-Matar, A. K.; Rockstraw, D. A. A generating equation for mixing rules and two new mixing rules for interatomic potential energy parameters. *Journal of Computational Chemistry* **2004**, *25*, 660–668.
- [56] Massively Parallel Monte Carlo (MPMC). <http://code.google.com/p/mpmc>, MPMC is an in-house code developed by Jonathan Belof and maintained by Keith McLaughlin, Christian R. Cioce and Brant Tudor. MPMC is released under GNU Public License and is available on Google Code <http://code.google.com/p/mpmc>.
- [57] Lemmon, E.; McLinden, M.; Friend, D. In *NIST Chemistry WebBook, NIST Standard Reference Database Number 69*; Linstrom, P., Mallard, W., Eds.; National Institute of Standards and Technology: Gaithersburg, MD, 2005.
- [58] Kuan, T.-S.; Warshel, A.; Schnepp, O. Intermolecular Potentials for N₂ Molecules and the Lattice Vibrations of Solid alpha-N₂. *The Journal of Chemical Physics* **1970**, *52*, 3012–3020.
- [59] Erba, A.; Maschio, L.; Salustro, S.; Casassa, S. A post-Hartree–Fock study of pressure-induced phase transitions in solid nitrogen: The case of the alpha, gamma, and epsilon low-pressure phases. *The Journal of Chemical Physics* **2011**, *134*, 074502.
- [60] van der Avoird, A.; Wormer, P. E. S.; Jansen, A. P. J. An improved intermolecular potential for nitrogen. *The Journal of Chemical Physics* **1986**, *84*, 1629–1635.
- [61] Mulder, A.; Michels, J. P. J.; Schouten, J. A. The importance of the anisotropic energy term for the structure of the solid phases of nitrogen. *The Journal of Chemical Physics* **1996**, *105*, 3235–3244.
- [62] Scott, T. Solid and liquid nitrogen. *Physics Reports* **1976**, *27*, 89 – 157.
- [63] Jordan, P. C.; van Maaren, P. J.; Mavri, J.; van der Spoel, D.; Berendsen, H. J. C. Towards phase transferable potential functions: Methodology and application to nitrogen. *The Journal of Chemical Physics* **1995**, *103*, 2272–2285.

- [64] Wyckoff, R. W. G. *Crystal Structures*; John Wiley & Sons, 1963; Vol. 1; pp 29–30.
- [65] Donnay, J. D. H., Donnay, G., Eds. *Crystal Data Determinative Tables*, 2nd ed.; American Crystallographic Association, 1963; p 884.
- [66] Raich, J. C.; Mills, R. L. alpha-gamma Transition in Solid Nitrogen and Carbon Monoxide at High Pressure. *The Journal of Chemical Physics* **1971**, *55*, 1811–1817.
- [67] Mills, R. L.; Schuch, A. F. Crystal Structure of Gamma Nitrogen. *Phys. Rev. Lett.* **1969**, *23*, 1154–1156.
- [68] Schuch, A. F.; Mills, R. L. Crystal Structures of the Three Modifications of Nitrogen 14 and Nitrogen 15 at High Pressure. *The Journal of Chemical Physics* **1970**, *52*, 6000–6008.
- [69] Pickard, C. J.; Needs, R. J. High-Pressure Phases of Nitrogen. *Phys. Rev. Lett.* **2009**, *102*, 125702.
- [70] McQuarrie, D. A.; Simon, J. D. *Physical chemistry: a molecular approach*; University Science Books: Sausalito, CA, 1997; p 663.
- [71] Scholz, M.; Melin, T.; Wessling, M. Transforming biogas into biomethane using membrane technology. *Renewable and Sustainable Energy Reviews* **2013**, *17*, 199 – 212.
- [72] Duan, X.; Zhang, Q.; Cai, J.; Yang, Y.; Cui, Y.; He, Y.; Wu, C.; Krishna, R.; Chen, B.; Qian, G. A new metal-organic framework with potential for adsorptive separation of methane from carbon dioxide, acetylene, ethylene, and ethane established by simulated breakthrough experiments. *J. Mater. Chem. A* **2014**, *2*, 2628–2633.
- [73] Pham, T.; Forrest, K. A.; Tudor, B.; Elsaidi, S. K.; Mohamed, M. H.; McLaughlin, K.; Cioce, C. R.; Zaworotko, M. J.; Space, B. Theoretical Investigations of CO₂ and CH₄ Sorption in an Interpenetrated Diamondoid Metal–Organic Material. *Langmuir* **2014**, *30*, 6454–6462.

- [74] Elsaidi, S. K.; Mohamed, M. H.; Wojtas, L.; Chanthapally, A.; Pham, T.; Space, B.; Vittal, J. J.; Zaworotko, M. J. Putting the Squeeze on CH₄ and CO₂ through Control over Interpenetration in Diamondoid Nets. *Journal of the American Chemical Society* **2014**, *136*, 5072–5077, PMID: 24611507.
- [75] Ma, S.; Sun, D.; Simmons, J. M.; Collier, C. D.; Yuan, D.; Zhou, H.-C. Metal-Organic Framework from an Anthracene Derivative Containing Nanoscopic Cages Exhibiting High Methane Uptake. *Journal of the American Chemical Society* **2008**, *130*, 1012–1016, PMID: 18163628.
- [76] Cioce, C. R.; McLaughlin, K.; Belof, J. L.; Space, B. A Polarizable and Transferable PHAST N₂ Potential for Use in Materials Simulation. *Journal of Chemical Theory and Computation* **2013**, *9*, 5550–5557.
- [77] Do, H.; Wheatley, R. J.; Hirst, J. D. Gibbs Ensemble Monte Carlo Simulations of Binary Mixtures of Methane, Difluoromethane, and Carbon Dioxide. *The Journal of Physical Chemistry B* **2010**, *114*, 3879–3886, PMID: 20184300.
- [78] Yang, Q.; Zhong, C. Molecular Simulation of Carbon Dioxide/Methane/Hydrogen Mixture Adsorption in MetalOrganic Frameworks. *The Journal of Physical Chemistry B* **2006**, *110*, 17776–17783, PMID: 16956262.
- [79] Düren, T.; Snurr, R. Q. Assessment of Isoreticular MetalOrganic Frameworks for Adsorption Separations: A Molecular Simulation Study of Methane/n-Butane Mixtures. *The Journal of Physical Chemistry B* **2004**, *108*, 15703–15708.
- [80] Ohkubo, T.; Miyawaki, J.; Kaneko, K.; Ryoo, R.; Seaton, N. A. Adsorption Properties of Tempered Mesoporous Carbon (CMK-1) for Nitrogen and Supercritical MethaneExperiment and GCMC Simulation. *The Journal of Physical Chemistry B* **2002**, *106*, 6523–6528.
- [81] Lyubartsev, A. P.; Førriisdahl, O. K.; Laaksonen, A. Solvation free energies of methane and alkali halide ion pairs: An expanded ensemble molecular dynamics simulation study. *The Journal of Chemical Physics* **1998**, *108*.

- [82] Tang, K. T.; Toennies, J. P. An improved simple model for the van der Waals potential based on universal damping functions for the dispersion coefficients. *The Journal of Chemical Physics* **1984**, *80*, 3726–3741.
- [83] Bini, R.; Pratesi, G. High-pressure infrared study of solid methane: Phase diagram up to 30 GPa. *Phys. Rev. B* **1997**, *55*, 14800–14809.
- [84] Neumann, M. A.; Press, W.; Nldeke, C.; Asmussen, B.; Prager, M.; Ibberson, R. M. The crystal structure of methane phase III. *The Journal of Chemical Physics* **2003**, *119*.
- [85] Kumar, R.; Keyes, T. Classical Simulations with the POLIR Potential Describe the Vibrational Spectroscopy and Energetics of Hydration: Divalent Cations, from Solvation to Coordination Complex. *Journal of the American Chemical Society* **2011**, *133*, 9441–9450.
- [86] Medders, G. R.; Babin, V.; Paesani, F. A critical assessment of two-body and three-body interactions in water. *Journal of Chemical Theory and Computation* **2013**, *9*, 1103–1114.
- [87] Belof, J. L.; Stern, A. C.; Eddaoudi, M.; Space, B. On the Mechanism of Hydrogen Storage in a Metal–Organic Framework Material. *Journal of the American Chemical Society* **2007**, *129*, 15202–15210, PMID: 17999501.
- [88] Stern, A. C.; Belof, J. L.; Eddaoudi, M.; Space, B. Understanding hydrogen sorption in a polar metal–organic framework with constricted channels. *Journal of Chemical Physics* **2012**, *136*, 034705.
- [89] Pham, T.; Forrest, K. A.; McLaughlin, K.; Tudor, B.; Nugent, P.; Hogan, A.; Mullen, A.; Cioce, C. R.; Zaworotko, M. J.; Space, B. Theoretical Investigations of CO₂ and H₂ Sorption in an Interpenetrated Square-Pillared Metal–Organic Material. *The Journal of Physical Chemistry C* **2013**, *117*, 9970–9982.
- [90] Forrest, K. A.; et al., Computational Studies of CO₂ Sorption and Separation in an Ultramicroporous Metal–Organic Material. *The Journal of Physical Chemistry C* **2013**, *117*, 17687–17698.

- [91] Forrest, K. A.; Pham, T.; Nugent, P.; Burd, S. D.; Mullen, A.; Wojtas, L.; Zaworotko, M. J.; Space, B. Examining the Effects of Different Ring Configurations and Equatorial Fluorine Atom Positions on CO₂ Sorption in [Cu(bpy)₂SiF₆]. *Crystal Growth & Design* **2013**, *13*, 4542–4548.
- [92] Pham, T.; Forrest, K. A.; Hogan, A.; McLaughlin, K.; Belof, J. L.; Eckert, J.; Space, B. Simulations of Hydrogen Sorption in *rht*-MOF-1: Identifying the Binding Sites Through Explicit Polarization and Quantum Rotation Calculations. *Journal of Materials Chemistry A* **2014**, *2*, 2088–2100.
- [93] Pham, T.; Forrest, K. A.; Eckert, J.; Georgiev, P. A.; Mullen, A.; Luebke, R.; Cairns, A. J.; Belmabkhout, Y.; Eubank, J. F.; McLaughlin, K.; Lohstroh, W.; Eddaoudi, M.; Space, B. Investigating the Gas Sorption Mechanism in an *rht*-Metal–Organic Framework through Computational Studies. *The Journal of Physical Chemistry C* **2014**, *118*, 439–456.
- [94] Forrest, K. A.; Pham, T.; McLaughlin, K.; Hogan, A.; Space, B. Insights into an intriguing gas sorption mechanism in a polar metal–organic framework with open-metal sites and narrow channels. *Chemical Communications* **2014**, *50*, 7283–7286.
- [95] Nugent, P.; Pham, T.; McLaughlin, K.; Georgiev, P. A.; Lohstroh, W.; Embs, J. P.; Zaworotko, M. J.; Space, B.; Eckert, J. Dramatic effect of pore size reduction on the dynamics of hydrogen adsorbed in metal–organic materials. *Journal of Materials Chemistry A* **2014**, *2*, 13884–13891.
- [96] Pham, T.; Forrest, K. A.; McLaughlin, K.; Eckert, J.; Space, B. Capturing the H₂–Metal Interaction in Mg-MOF-74 Using Classical Polarization. *The Journal of Physical Chemistry C* **2014**, *118*, 22683–22690.
- [97] Pham, T.; Forrest, K. A.; Georgiev, P. A.; Lohstroh, W.; Xue, D.-X.; Hogan, A.; Eddaoudi, M.; Space, B.; Eckert, J. A high rotational barrier for physisorbed hydrogen in an fcu-metal–organic framework. *Chemical Communications* **2014**, *50*, 14109–14112.

- [98] Pham, T.; Forrest, K. A.; McDonald, K.; Space, B. Modeling PCN-61 and PCN-66: Isostructural *rht*-Metal–Organic Frameworks with Distinct CO₂ Sorption Mechanisms. *Crystal Growth & Design* **2014**, DOI: 10.1021/cg500860t.
- [99] Allen, M. P.; Tildesley, D. J. *Computer Simulation of Liquids*; Oxford University Press: Oxford, United Kingdom, 1989; pp. 21.
- [100] Waldman, M.; Hagler, A. New combining rules for rare gas van der Waals parameters. *Journal of Computational Chemistry* **1993**, *14*, 1077–1084.
- [101] Yu, K.; McDaniel, J. G.; Schmidt, J. Physically motivated, robust, ab initio force fields for CO₂ and N₂. *The Journal of Physical Chemistry B* **2011**, *115*, 10054–10063.
- [102] Hudson, G. H.; McCoubrey, J. C. Intermolecular forces between unlike molecules. A more complete form of the combining rules. *Transactions of the Faraday Society* **1960**, *56*, 761–766.
- [103] McDaniel, J. G.; Yu, K.; Schmidt, J. R. Ab Initio, Physically Motivated Force Fields for CO₂ Adsorption in Zeolitic Imidazolate Frameworks. *The Journal of Physical Chemistry C* **2012**, *116*, 1892–1903.
- [104] McDaniel, J. G.; Schmidt, J. R. Robust, Transferable, and Physically Motivated Force Fields for Gas Adsorption in Functionalized Zeolitic Imidazolate Frameworks. *The Journal of Physical Chemistry C* **2012**, *116*, 14031–14039.
- [105] McDaniel, J. G.; Schmidt, J. Physically-Motivated Force Fields from Symmetry-Adapted Perturbation Theory. *The Journal of Physical Chemistry A* **2013**, *117*, 2053–2066, PMID: 23343200.
- [106] Allinger, N. L.; Yuh, Y. H.; Lii, J. H. Molecular mechanics. The MM3 force field for hydrocarbons. 1. *Journal of the American Chemical Society* **1989**, *111*, 8551–8566.
- [107] Hart, J.; Rappe, A. van der Waals functional forms for molecular simulations. *The Journal of Chemical Physics* **1992**, *97*, 1109–1115.

- [108] Stone, A.; Misquitta, A. Atom–atom potentials from ab initio calculations. *International Reviews in Physical Chemistry* **2007**, *26*, 193–222.
- [109] Casimir, H. B. G.; Polder, D. The Influence of Retardation on the London-van der Waals Forces. *Physical Review* **1948**, *73*, 360–372.
- [110] Abramowitz, M. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*; Dover Publications, Inc.: Mineola, NY, 1965; Chapt. 25, Sect. 4, pp. 887.
- [111] Thakkar, A. J. Higher dispersion coefficients: Accurate values for hydrogen atoms and simple estimates for other systems. *The Journal of Chemical Physics* **1988**, *89*, 2092–2098.
- [112] Hirschfelder,; Curtiss,; Bird,; Spotz, *The Molecular Theory of Gases and Liquids*; John Wiley and Sons, Inc.: New York, NY, 1954.
- [113] Huang, L.; Chen, Y.; Gao, G.-Y.; Zhang, X. P. Diastereoselective and enantioselective cyclopropanation of alkenes catalyzed by cobalt porphyrins. *J. Org. Chem.* **2003**, *68*, 8179.
- [114] Chen, Y.; Zhang, X. P. Asymmetric Cyclopropanation of Styrenes Catalyzed by Metal Complexes of D2-Symmetrical Chiral Porphyrin: Superiority of Cobalt over Iron. *J. Org. Chem.* **2007**, *72*, 5931.
- [115] Ruppel, J. V.; Gauthier, T. J.; Snyder, N. L.; Perman, J. A.; Zhang, X. P. Asymmetric Co(II)-Catalyzed Cyclopropanation with Succinimidyl Diazoacetate: General Synthesis of Chiral Cyclopropyl Carboxamides. *Org. Lett.* **2009**, *11*, 2273.
- [116] Zhu, S.; Ruppel, J. V.; Lu, H.; Wojtas, L.; Zhang, X. P. Cobalt-Catalyzed Asymmetric Cyclopropanation with Diazosulfones: Rigidification and Polarization of Ligand Chiral Environment via Hydrogen Bonding and Cyclization. *J. Am. Chem. Soc.* **2008**, *130*, 5042.
- [117] Chen, Y.; Ruppel, J. V.; Zhang, X. P. Cobalt-catalyzed asymmetric cyclopropanation of electron-deficient olefins. *J. Am. Chem. Soc.* **2007**, *129*, 12074.

- [118] Zhu, S.; Perman, J. A.; Zhang, X. P. Acceptor/Acceptor-Substituted Diazo Reagents for Carbene Transfers: Cobalt-Catalyzed Asymmetric *Z*-Cyclopropanation of Alkenes with α -Nitrodiazoacetates. *Angew. Chem. Int. Ed.* **2008**, *47*, 8460.
- [119] Chen, Y.; Fields, K. B.; Zhang, X. P. Bromoporphyrins as Versatile Synthons for Modular Construction of Chiral Porphyrins: Cobalt-Catalyzed Highly Enantioselective and Diastereoselective Cyclopropanation. *J. Am. Chem. Soc.* **2004**, *126*, 14718.
- [120] Penoni, A.; Wanke, R.; Tollari, S.; Gallo, E.; Musella, D.; Regaini, F.; Demartin, F.; Cenini, S. Cyclopropanation of Olefins with Diazoalkanes, Catalyzed by CoII(porphyrin) Complexes - A Synthetic and Mechanistic Investigation and the Molecular Structure of CoIII(TPP)(CH₂CO₂Et). *Eur. J. Inorg. Chem.* **2003**, *2003*, 1452.
- [121] Caselli, A.; Gallo, E.; Ragagni, F.; Ricatto, F.; Abbiati, G.; Cenini, S. Chiral Porphyrin complexes of cobalt(II) and ruthenium(II) in catalytic cyclopropanation and animation reactions. *Inorg. Chim. Acta* **2006**, *359*, 2924.
- [122] Fantauzzi, S.; Gallo, E.; Rose, E.; Raoul, N.; Caselli, A.; Issa, S.; Ragagni, F.; Cenini, S. Asymmetric Cyclopropanation of Olefins Catalyzed by Chiral Cobalt(II)-Binaphthyl Porphyrins. *Organometallics* **2008**, *27*, 6143.
- [123] Dzik, W. I.; Xu, X.; Zhang, X. P.; Reek, J. N. H.; de Bruin, B. Carbene Radicals in CoII(por)-Catalyzed Olefin Cyclopropanation. *J. Am. Chem. Soc.* **2010**, *132*, 10891.
- [124] Woodcock, H. L.; Moran, D.; Brooks, B. R.; Schleyer, P. V.; Schaefer, H. F. Carbene stabilization by aryl substituents. Is bigger better? *J. Am. Chem. Soc.* **2007**, *129*, 3763.
- [125] Salaun, J. In *Small Ring Compounds in Organic Synthesis VI*; de Meijere, A., Ed.; Top. Curr. Chem.; Springer Berlin / Heidelberg, 2000; Vol. 207; p 1.
- [126] Im, W.; et al., Differential Affinity Of Dihydroimidazoquinoxalines And Diimidazoquinoxalines To The Alpha 1 Beta 2 Gamma 2 And Alpha 6 Beta 2 Gamma 2 Subtypes Of Cloned GABAA Receptors. *Br. J. Pharmacol.* **1993**, *110*, 677.

- [127] Tenbrink, R.; IM, W.; Sethy, V.; Tang, A.; Carter, D. Antagonist, Partial Agonist, And Full Agonist Imidazo[1,5-a]quinoxaline Amides And Carbamates Acting Through The GABAA/benzodiazepine Receptor. *J. Med. Chem.* **1994**, *37*, 758.
- [128] Ishikura, K.; Kubota, T.; Minami, K.; Hamashima, Y.; Nakashimizu, H.; Motokawa, K.; Yoshida, T. Synthesis And Structure-Activity Relationships Of 7- β -[(Z)-2-(2-aminothiazol-4-YL)-3-(substituted)-2-propenoylamino]- 3-desacetoxyethylcephalosporins. *J. Antibiot.* **1994**, *47*, 453.
- [129] Jacobsen, E.; Tenbrink, R.; Stelzer, L.; Belonga, K.; Carter, D.; Im, H.; Im, W.; Sethy, V.; Tang, A.; VonVoigtlander, P.; Petke, J. High-affinity partial agonist imidazol[1,5-a]quinoxaline amides, carbamates, and ureas at the γ -aminobutyric acid A/benzodiazepine receptor complex. *J. Med. Chem.* **1996**, *39*, 158.
- [130] Mickelson, J.; Jacobsen, E.; Carter, D.; Im, H.; Im, W.; Schreur, P.; Sethy, V.; Tang, A.; McGee, J.; Petke, J. High-affinity α -aminobutyric acid A/benzodiazepine ligands: Synthesis and structure-activity relationship studies of a new series of tetracyclic imidazoquinolines. *J. Med. Chem.* **1996**, *39*, 4654.
- [131] Doyle, M. P.; McKervey, M. A.; Ye, T. *Modern Catalytic Methods for Organic Synthesis with Diazo Compounds: From Cyclopropanes to Ylides*; Wiley-Interscience, 1998.
- [132] Nozaki, H.; Moriuti, S.; Takaya, H.; Noyori, R. Asymmetric Induction In Carbenoid Reaction By Means Of A Dissymmetric Copper Chelate. *Tetrahedron Lett.* **1966**, 5239.
- [133] Fritschi, H.; Leutenegger, U.; Pfaltz, A. Chiral Copper Semicorrin Complexes As Enantioselective Catalysts For The Cyclopropanation Of Olefins By Diazo-Compounds. *Angew. Chem. Int. Ed.* **1986**, *25*, 1005.
- [134] Lowenthal, R.; Abiko, A.; Masamune, S. Asymmetric Catalytic Cyclopropanation Of Olefins: Bis-Oxazoline Copper Complexes. *Tetrahedron Lett.* **1990**, *31*, 6005.

- [135] Evans, D.; Woerpel, K.; Hinman, M.; Faul, M. Bis(oxazolines) As Chiral Ligands In Metal-Catalyzed Asymmetric Reactions. Catalytic, Asymmetric Cyclopropanation Of Olefins. *J. Am. Chem. Soc.* **1991**, *113*, 726.
- [136] Callot, H.; Piechocki, C. Cyclopropanation Using rhodium (III) Porphyrins: Large *cis* vs *trans* Selectivity. *Tetrahedron Lett.* **1980**, *21*, 3489.
- [137] Callot, H.; Metz, F.; Piechocki, C. Sterically Crowded Cyclopropanation Catalysts. *Syn-Selectivity Using Rhodium(III)Porphyrins. Tetrahedron* **1982**, *38*, 2365.
- [138] Omalley, S.; Kodadek, T. Asymmetric Cyclopropanation Of Alkenes Catalyzed By A Chiral Wall Porphyrin. *Tetrahedron Lett.* **1991**, *32*, 2445.
- [139] Maxwell, J.; Kodadek, T. Organometallic Chemistry Of Porphyrins: Spectroscopic And Chemical Characterization Of A Rhodium Porphyrin-Ethyl Diazoacetate Adduct. *Organometallics* **1991**, *10*, 4.
- [140] Brown, K.; Kodadek, T. A Transition-State Model For The Rhodium Porphyrin-Catalyzed Cyclopropanation Of Alkenes By Diazo Esters. *J. Am. Chem. Soc.* **1992**, *114*, 8336.
- [141] Omalley, S.; Kodadek, T. Asymmetric Cyclopropanation Of Alkenes Catalyzed By A Rhodium Chiral Fortress Porphyrin. *Organometallics* **1992**, *11*, 2299.
- [142] Maxwell, J.; Brown, K.; Bartley, D.; Kodadek, T. Mechanism Of The Rhodium Porphyrin Catalyzed Cyclopropanation Of Alkenes. *Science* **1992**, *256*, 1544.
- [143] Maxwell, J.; Omalley, S.; Brown, K.; Kodadek, T. Shape-Selective And Asymmetric Cyclopropanation Of Alkenes Catalyzed By Rhodium Porphyrins. *Organometallics* **1992**, *11*, 645.
- [144] Bartley, D.; Kodadek, T. Identification Of The Active Catalyst In The Rhodium Porphyrin-Mediated Cyclopropanation Of Alkenes. *J. Am. Chem. Soc.* **1993**, *115*, 1656.

- [145] Doyle, M.; Pieters, R.; Martin, S.; Austin, R.; Oalmann, C.; Muller, P. High Enantioselectivity In The Intramolecular Cyclopropanation Of Allyl Diazoacetates Using A Novel rhodium(II) Catalyst. *J. Am. Chem. Soc.* **1991**, *113*, 1423.
- [146] others,, et al. Enantioselective Intramolecular Cyclopropanations Of Allylic And Homoallylic Diazoacetates And Diazoacetamides Using Chiral Dirhodium(II) Carboxamide Catalysts. *J. Am. Chem. Soc.* **1995**, *117*, 5763.
- [147] Wolf, J.; Hamaker, C.; Djukic, J.; Kodadek, T.; Woo, L. Shape And Stereoselective Cyclopropanation Of Alkenes Catalyzed By Iron Porphyrins. *J. Am. Chem. Soc.* **1995**, *117*, 9194.
- [148] Lo, W.; Che, C.; Cheng, K.; Mak, T. Catalytic and asymmetric cyclopropanation of styrenes catalysed by ruthenium porphyrin and porphycene complexes. *Chem. Comm.* **1997**, 1205.
- [149] Che, C.; Huang, J.; Lee, F.; Li, Y.; Lai, T.; Kwong, H.; Teng, P.; Lee, W.; Lo, W.; Peng, S.; Zhou, Z. Asymmetric inter- and intramolecular cyclopropanation of alkenes catalyzed by chiral ruthenium porphyrins. Synthesis and crystal structure of a chiral metalloporphyrin carbene complex. *J. Am. Chem. Soc.* **2001**, *123*, 4119.
- [150] Teng, P.; Lai, T.; Kwong, H.; Che, C. Asymmetric inter- and intramolecular cyclopropanations of alkenes catalyzed by rhodium D_4 -porphyrin: A comparison of rhodium- and ruthenium-centred catalysts. *Tetrahedron-Asymmetr.* **2003**, *14*, 837.
- [151] Galardon, E.; LeMaux, P.; Simonneaux, G. Cyclopropanation of alkenes with ethyl diazoacetate catalysed by ruthenium porphyrin complexes. *Chem. Comm.* **1997**, 927.
- [152] Galardon, E.; Roue, S.; Le Maux, P.; Simonneaux, G. Asymmetric cyclopropanation of alkenes and diazocarbonyl insertion into S–H bonds catalyzed by a chiral porphyrin Ru(II) complex. *Tetrahedron Lett.* **1998**, *39*, 2333.

- [153] Paul-Roth, C.; De Montigny, F.; Rehtore, G.; Simonneaux, G.; Gulea, M.; Masson, S. Cyclopropanation of alkenes with diisopropyl diazomethylphosphonate catalysed by ruthenium porphyrin complexes. *J. Mol. Catal. A.: Chem.* **2003**, *201*, 79.
- [154] Frauenkron, M.; Berkessel, A. A novel chiral ruthenium porphyrin as highly efficient and selective catalyst for asymmetric cyclopropanations. *Tetrahedron Lett.* **1997**, *38*, 7175.
- [155] Gross, Z.; Galili, N.; Simkhovich, L. Metalloporphyrin catalyzed asymmetric cyclopropanation of olefins. *Tetrahedron Lett.* **1999**, *40*, 1571.
- [156] Nakamura, A.; Konishi, A.; Tatsuno, Y.; Otsuka, S. A Highly Enantioselective Synthesis Of Cyclopropane Derivatives Through Chiral cobalt(II) Complex Catalyzed Carbenoid Reaction. General Scope And Factors Determining the Enantioselectivity. *J. Am. Chem. Soc.* **1978**, *100*, 3443.
- [157] Niimi, T.; Uchida, T.; Irie, R.; Katsuki, T. Highly enantioselective cyclopropanation with Co(II)-salen complexes: Control of cis- and trans-selectivity by rational ligand-design. *Adv. Synth. Catal.* **2001**, *343*, 79.
- [158] Ikeno, T.; Sato, M.; Sekino, H.; Nishizuka, A.; Yamada, T. Highly enantioselective cyclopropanation of styrenes and diazoacetates catalyzed by 3-oxobutylideneaminatocobalt(II) complexes, Part 1. Designs of cobalt complex catalysts and the effects of donating ligands. *Bull. Chem. Soc. Japan* **2001**, *74*, 2139.
- [159] Doyle, M. Exceptional Selectivity in Cyclopropanation Reactions Catalyzed by Chiral Cobalt(II)-Porphyrin Catalysts. *Angew. Chem. Int. Ed.* **2009**, *48*, 850.
- [160] Ruppel, J. V.; Jones, J. E.; Huff, C. A.; Kamble, R. M.; Chen, Y.; Zhang, X. P. A highly effective cobalt catalyst for olefin aziridination with azides: Hydrogen bonding guided catalyst design. *Org. Lett.* **2008**, *10*, 1995.

- [161] Jones, J. E.; Ruppel, J. V.; Gao, G. Y.; Moore, T. M.; Zhang, X. P. Cobalt-catalyzed asymmetric olefin aziridination with diphenylphosphoryl azide. *J. Org. Chem.* **2008**, *73*, 7260.
- [162] Ikeno, T.; Iwakura, I.; Yamada, T. Cobalt-Carbene Complex with Single-Bond Character: Intermediate for the Cobalt Complex-Catalyzed Cyclopropanation. *J. Am. Chem. Soc.* **2002**, *124*, 15152.
- [163] Iwakura, I.; Tanaka, H.; Taketo, I.; Yamada, T. FT-IR and Theoretical Analysis of the Characteristic Bonding Properties in the Multiplet Metal Porphyrin Carbene Complexes. *Chem. Lett.* **2004**, *33*, 140.
- [164] Fischer, E. O.; Maasböl, A. On the Existence of a Tungsten Carbonyl Carbene Complex. *Angew. Chem. Int. Ed.* **1964**, *3*, 580.
- [165] Schrock, R. R. Alkylcarbene complex of tantalum by intramolecular α -hydrogen abstraction. *J. Am. Chem. Soc.* **1974**, *96*, 6796.
- [166] Wikipedia, Transition metal carbene complex. 2006; http://en.wikipedia.org/wiki/Transition_metal_carbene_complex, [Online; accessed 16-June-2010].
- [167] Dias, E. L.; Nguyen, S. T.; Grubbs, R. H. Well-Defined Ruthenium Olefin Metathesis Catalysts: Mechanism and Activity. *J. Am. Chem. Soc.* **1997**, *119*, 3887.
- [168] Dewar, M. J. S. A review of the p-complex theory. *Bull. Soc. Chim. Fr.* **1951**, *18*, C71.
- [169] Chatt, J.; Duncanson, L. Olefin co-ordination compounds. Part III. Infra-red spectra and structure: Attempted preparation of acetylene complexes. *J. Chem. Soc.* **1953**, *2939*, 2947.
- [170] Frenking, G.; Sola, M.; Vyboishchikov, S. Chemical bonding in transition metal carbene complexes. *J. Organomet. Chem.* **2005**, *690*, 6178.
- [171] Ehlers, A.; Dapprich, S.; Vyboishchikov, S.; Frenking, G. Structure and Bonding of the Transition-Metal Carbonyl Complexes $M(CO)_5L$ and $M(CO)_3L$. *Organometallics* **1996**, *15*, 105.

- [172] Ziegler, T. Approximate Density Functional Theory As A Practical Tool In Molecular Energetics And Dynamics. *Chem. Rev.* **1991**, *91*, 651.
- [173] Cundari, T. R.; Benson, M. T.; Lutz, M. L.; Sommerer, S. O. *Reviews in Computational Chemistry*; John Wiley & Sons, Inc.: Hoboken, 1996; p 145.
- [174] Frenking, G.; Antes, I.; Bhme, M.; Dapprich, S.; Ehlers, A. W.; Jonas, V.; Neuhaus, A.; Otto, M.; Stegmann, R.; Veldkamp, A.; Vyboishchikov, S. F. *Reviews in Computational Chemistry*; John Wiley & Sons, Inc.: Hoboken, 1996; p 63.
- [175] Zhao, Y.; Truhlar, D. G. A new local density functional for main-group thermochemistry, transition metal bonding, thermochemical kinetics, and noncovalent interactions. *J. Chem. Phys.* **2006**, *125*, 194101.
- [176] Woodcock, H. L.; Schaefer, H. F.; Schreiner, P. R. Problematic energy differences between cumulenes and poly-ynes: Does this point to a systematic improvement of density functional theory? *J. Phys. Chem. A* **2002**, *106*, 11923.
- [177] Furche, F.; Perdew, J. The performance of semilocal and hybrid density functionals in 3d transition-metal chemistry. *J. Chem. Phys.* **2006**, *124*.
- [178] Cobar, E. A.; Khaliullin, R. Z.; Bergman, R. G.; Head-Gordon, M. Theoretical study of the rhenium-alkane interaction in transition metal-alkane sigma-complexes. *Proc. Natl. Acad. Sci. U.S.A.* **2007**, *104*, 6963.
- [179] Foster, J.; Weinhold, F. Natural Hybrid Orbitals. *J. Am. Chem. Soc.* **1980**, *102*, 7211.
- [180] Glendening, E. D.; Badenhoop, J. K.; Reed, A. E.; Carpenter, J. E.; Bohmann, J. A.; Morales, C. M.; Weinhold, F. Natural Bond Orbital Analysis. 2001.
- [181] Weinhold, F.; Landis, C. Natural bond orbitals and extensions of localized bonding concepts. *Chem. Ed.* **2001**, *2*, 91.
- [182] Dapprich, S.; Frenking, G. Investigation of Donor-Acceptor Interactions: A Charge Decomposition Analysis Using Fragment Molecular Orbitals. *J. Phys. Chem.* **1995**, *99*, 9352.

- [183] Khaliullin, R.; Head-Gordon, M.; Bell, A. An efficient self-consistent field method for large systems of weakly interacting components. *J. Chem. Phys.* **2006**, *124*.
- [184] Khaliullin, R. Z.; Cobar, E. A.; Lochan, R. C.; Bell, A. T.; Head-Gordon, M. Unravelling the origin of intermolecular interactions using absolutely localized molecular orbitals. *J. Phys. Chem. A.* **2007**, *111*, 8753.
- [185] Khaliullin, R. Z.; Bell, A. T.; Head-Gordon, M. Analysis of charge transfer effects in molecular complexes based on absolutely localized molecular orbitals. *J. Chem. Phys.* **2008**, *128*.
- [186] Kohn, W.; Sham, L. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev. A.* **1965**, *140*, 1133.
- [187] others,, et al. Advances in methods and algorithms in a modern quantum chemistry program package. *Phys. Chem. Chem. Phys.* **2006**, *8*, 3172.
- [188] others,, et al. General Atomic and Molecular Electronic Structure System. *J. Comput. Chem.* **1993**, *14*, 1347.
- [189] Roy, L. E.; Hay, P. J.; Martin, R. L. Revised Basis Sets for the LANL Effective Core Potentials. *J. Chem. Theory Comput.* **2008**, *4*, 1029.
- [190] Hehre, W. J.; Ditchfield, R.; Pople, J. A. Self-Consistent Molecular Orbital Methods. XII. Further Extensions of Gaussian-Type Basis Sets for Use in Molecular Orbital Studies of Organic Molecules. *J. Chem. Phys.* **1972**, *56*, 2257.
- [191] Gill, P. M. W.; Johnson, B. G.; Pople, J. A. A standard grid for density functional calculations. *Chem. Phys. Lett.* **1993**, *209*, 506.
- [192] Vyboishchikov, S. F.; Frenking, G. Structure and Bonding of Low-Valent (Fischer-Type) and High-Valent (Schrock-Type) Transition Metal Carbene Complexes. *Chem. Eur. J.* **1998**, *4*, 1428.

- [193] Becke, A. D. Density-functional thermochemistry. III. The role of exact exchange. *J. Chem. Phys.* **1993**, *98*, 5648.
- [194] Shan Li, B. d.; Peng, C.-H.; Fryd, M.; Wayland, B. B. Exchange of Organic Radicals with Organo-Cobalt Complexes Formed in the Living Radical Polymerization of Vinyl Acetate. *J. Am. Chem. Soc.* **2008**, *130*, 13374.
- [195] de Bruin, B.; Dzik, W. I.; Li, S.; Wayland, B. B. Hydrogen-Atom Transfer in Reactions of Organic Radicals with [CoII(por)](por=Porphyrinato) and in Subsequent Addition of [Co(H)(por)] to Olefins. *Chem. Eur. J.* **2009**, *15*, 4312.
- [196] Toxic Industrial Chemicals/Toxic Industrial Materials (TICs/TIMs) – Awareness and Preventative Measures. <http://deploymenthealthlibrary.fhp.osd.mil/Product/RetrieveFile?prodId=301>, Accessed: 2015-03-11.
- [197] DeCoste, J. B.; Peterson, G. W. MetalOrganic Frameworks for Air Purification of Toxic Chemicals. *Chemical Reviews* **2014**, *114*, 5695–5727, PMID: 24750116.
- [198] Maza, W. A.; Vetromile, C. M.; Kim, C.; Xu, X.; Zhang, X. P.; Larsen, R. W. Spectroscopic Investigation of the Noncovalent Association of the Nerve Agent Simulant Diisopropyl Methylphosphonate (DIMP) with Zinc(II) Porphyrins. *The Journal of Physical Chemistry A* **2013**, *117*, 11308–11315, PMID: 24093669.
- [199] Alam, R.; Lightcap, I. V.; Karwacki, C. J.; Kamat, P. V. Sense and Shoot: Simultaneous Detection and Degradation of Low-Level Contaminants Using Graphene-Based Smart Material Assembly. *ACS Nano* **2014**, *8*, 7272–7278, PMID: 24893206.
- [200] Kwon, O. S.; Park, S. J.; Lee, J. S.; Park, E.; Kim, T.; Park, H.-W.; You, S. A.; Yoon, H.; Jang, J. Multidimensional Conducting Polymer Nanotubes for Ultrasensitive Chemical Nerve Agent Sensing. *Nano Letters* **2012**, *12*, 2797–2802, PMID: 22545863.

- [201] Zhao, J.; Gong, B.; Nunn, W. T.; Lemaire, P. C.; Stevens, E. C.; Sidi, F. I.; Williams, P. S.; Oldham, C. J.; Walls, H. J.; Shepherd, S. D.; Browe, M. A.; Peterson, G. W.; Losego, M. D.; Parsons, G. N. Conformal and highly adsorptive metal-organic framework thin films via layer-by-layer growth on ALD-coated fiber mats. *J. Mater. Chem. A* **2015**, *3*, 1458–1464.
- [202] Montoro, C.; Linares, F.; Quartapelle Procopio, E.; Senkovska, I.; Kaskel, S.; Galli, S.; Masciocchi, N.; Barea, E.; Navarro, J. A. R. Capture of Nerve Agents and Mustard Gas Analogues by Hydrophobic Robust MOF-5 Type MetalOrganic Frameworks. *Journal of the American Chemical Society* **2011**, *133*, 11888–11891, PMID: 21761835.
- [203] Mondloch, J. E.; Katz, M. J.; Isley III, W. C.; Ghosh, P.; Liao, P.; Bury, W.; Wagner, G. W.; Hall, M. G.; DeCoste, J. B.; Peterson, G. W.; Snurr, R. Q.; Cramer, C. J.; Hupp, J. T.; Farha, O. K. Destruction of chemical warfare agents using metal-organic frameworks. *Nature Materials* **2015**, *Advanced online publication*, 1–5.
- [204] Gordon, W. O.; Peterson, G. W.; Durke, E. M. Reduced Chemical Warfare Agent Sorption in Polyurethane-Painted Surfaces via Plasma-Enhanced Chemical Vapor Deposition of Perfluoroalkanes. *ACS Applied Materials & Interfaces* **0**, *0*, null, PMID: 25775244.

Appendix A

solidN2.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
#
#  ©2013, Christian Cioce
#  Space Research Group
#  Department of Chemistry
#  University of South Florida
#  Email: ccioc@mail.usf.edu
#
#  This script has been used in modeling the crystalline phases of N2
#  and CO2, which has resulted in the following publications:
#
#  + C.R. Cioce, K. McLaughlin, J.L. Belof, B. Space, "A Polarizable and
#    Transferable PHAST N2 Potential for use in Materials Simulation",
#    J. Chem. Theory Comput., 2013, 9 (12), 55505557
#
#  - AND -
#
#  + A.L. Mullen, T. Pham, K.A. Forrest, C.R. Cioce, K. McLaughlin, B. Space,
#    "A Polarizable and Transferable PHAST CO2 Potential for Materials Simulation",
#    J. Chem. Theory Comput., 2013, 9 (12), 54215429
#
#  license: GNU LGPL
#
#  This library is free software; you can redistribute it and/or
#  modify it under the terms of the GNU Lesser General Public
#  License as published by the Free Software Foundation; either
#  version 2.1 of the License, or (at your option) any later version.
#
#  NOTE: The author kindly requests acknowledgement in any further use of
#        this script. Though optional, it is the right thing to do!

"""This script REQUIRES the input molecule to be diatomic. Homo/Heteronuclear independent."""
"""This version REQUIRES that the initial unit cell be cubic. For generalization, add conditionals to check min and max
coordinates for each dimension independently."""

"""

For alphaN2 (Pa3(bar)):

PRIMITIVE VECTORS:
A1 = aX      a = 5.644 (recip: 0.17717931)
A2 = aY
A3 = aZ

BASIS VECTORS:
B1 = uA1 + uA2 + uA3      = uaX + uaY + uaZ      (N) (8c)
B2 = (1/2-u)A1 - uA2 + (1/2+u)A3 = (1/2-u)aX - uaY + (1/2+u)aZ (N) (8c)
B3 = -uA1 + (1/2+u)A2 + (1/2-u)A3 = -uaX + (1/2+u)aY + (1/2-u)aZ (N) (8c)
B4 = (1/2+u)A1 + (1/2-u)A2 - uA3 = (1/2+u)aX + (1/2-u)aY - uaZ (N) (8c)
```

```

B5 = -uA1 - uA2 - uA3 = -uaX - uaY - uaZ (N) (8c)
B6 = (1/2+u)A1 + uA2 + (1/2-u)A3 = (1/2+u)aX + uaY + (1/2-u)aZ (N) (8c)
B7 = uA1 + (1/2-u)A2 + (1/2+u)A3 = uaX + (1/2-u)aY + (1/2+u)aZ (N) (8c)
B8 = (1/2-u)A1 + (1/2+u)A2 + uA3 = (1/2-u)aX + (1/2+u)aY + uaZ (N) (8c)
"""

"""

For gammaN2 (P4_2/mnm):

PRIMITIVE VECTORS:
A1 = aX a = 3.957 (recip: 0.25271670)
A2 = aY
A3 = cZ c = 5.109 (recip: 0.19573302)

BASIS VECTORS:
B1 = uA1 + uA2 = uaX + uaY (N) (4f)
B2 = -uA1 - uA2 = -uaX - uaY (N) (4f)
B3 = (1/2+u)A1 + (1/2-u)A2 + (1/2)A3 = (1/2+u)aX + (1/2-u)aY + (1/2)cZ (N) (4f)
B4 = (1/2-u)A1 + (1/2+u)A2 + (1/2)A3 = (1/2-u)aX + (1/2+u)aY + (1/2)cZ (N) (4f)
"""

import sys
import os
import time
from numpy import *
from scitools import *
from math import *
from subprocess import call
from scipy import *
from pylab import *
from shutil import move
import scitools.table as ft
import matplotlib
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec

#####
# search a dictionary for key or value using named functions or a class
# tested with Python25 by Ene Uran 01/19/2008
# http://www.daniweb.com/software-development/python/code/217019
def find_key(dic, val):
    """return the key of dictionary dic given the value"""
    return [k for k, v in symbol_dic.iteritems() if v == val][0]

def find_value(dic, key):
    """return the value of dictionary dic given the key"""
    return dic[key]

class Lookup(dict):
    """
    a dictionary which can lookup value by key, or keys by value
    """
    def __init__(self, items=[]):
        """items can be a list of pair_lists or a dictionary"""
        dict.__init__(self, items)

    def get_key(self, value):
        """find the key(s) as a list given a value"""
        return [item[0] for item in self.items() if item[1] == value]

    def get_value(self, key):
        """find the value given a key"""
        return self[key]
#####

def rmsd(crds1, crds2):
    """Returns RMSD between 2 sets of [nx3] numpy arrays"""

```

```

assert(crds1.shape[1] == 3)
assert(crds1.shape == crds2.shape)

n_vec = shape(crds1)[0]
correlation_matrix = dot(transpose(crds1), crds2)
v, s, w = linalg.svd(correlation_matrix)
is_reflection = (linalg.det(v) * linalg.det(w)) < 0.0
if is_reflection:
    s[-1] = - s[-1]
E0 = sum(sum(crds1 * crds1)) + sum(sum(crds2 * crds2))
rmsd_sq = (E0 - 2.0*sum(s)) / float(n_vec)
rmsd_sq = max([rmsd_sq, 0.0])
return sqrt(rmsd_sq)

# Alpha Loop
def alpha_loop(mixrule, pol_on, n_cells, abc_min, abc_max, abc_del, x, breal, parentdir, trajfile, input_file, job_name, pqr_input,
              pqr_output, pqr_restart, energy_output, traj_output, dipole_output, field_output, sitelist, masslist, qlist, pollist, epslist,
              siglist, site, totalelems, totalrows):
    """Generates alpha-N2 crystals"""

    for a_current in arange(abc_min, abc_max, abc_del):
        trajfile.write("REMARK Trajectory File created in Python by Chris Cioce\n")
        trajfile.write("REMARK a = %f Angstroms" % (a_current))
        print "*****"
        print "Current a=b=c = %f Angstroms" % a_current
        start_local = time.time()
        # Keep N cartesian coordinate static (specific code here...generalize)  <- Determined by generating N2E sites and looking at
        # xyz of N closest to basis origin
        N_fixed = 0.316965  # Reciprocal space coordinate
        u = N_fixed / float(a_current)

        # Generate Unit Cell from Basis Vectors (bv)
        d = zeros(x.size)
        e = zeros(x.size)
        f = zeros(x.size)

        # N(1) : B_4
        bvx = (0.5+u) * a_current
        bvy = (0.5-u) * a_current
        bvz = -u * a_current
        d.itemset(0,bvx)
        e.itemset(0,bvy)
        f.itemset(0,bvz)

        # N(2) : B_8
        bvx = (0.5-u) * a_current
        bvy = (0.5+u) * a_current
        bvz = u * a_current
        d.itemset(1,bvx)
        e.itemset(1,bvy)
        f.itemset(1,bvz)

        # N(3) : B_3
        bvx = -u * a_current
        bvy = (0.5+u) * a_current
        bvz = (0.5-u) * a_current
        d.itemset(2,bvx)
        e.itemset(2,bvy)
        f.itemset(2,bvz)

        # N(4) : B_7
        bvx = u * a_current
        bvy = (0.5-u) * a_current
        bvz = (0.5+u) * a_current
        d.itemset(3,bvx)
        e.itemset(3,bvy)
        f.itemset(3,bvz)

```

```

# N(5) : B_1
bvx =     u * a_current
bvy =     u * a_current
bvz =     u * a_current
d.itemset(4,bvx)
e.itemset(4,bvy)
f.itemset(4,bvz)

# N(6) : B_5
bvx =    -u * a_current
bvy =    -u * a_current
bvz =    -u * a_current
d.itemset(5,bvx)
e.itemset(5,bvy)
f.itemset(5,bvz)

# N(7) : B_2
bvx = (0.5-u) * a_current
bvy =    -u * a_current
bvz = (0.5+u) * a_current
d.itemset(6,bvx)
e.itemset(6,bvy)
f.itemset(6,bvz)

# N(8) : B_6
bvx = (0.5+u) * a_current
bvy =     u * a_current
bvz = (0.5-u) * a_current
d.itemset(7,bvx)
e.itemset(7,bvy)
f.itemset(7,bvz)

x = d.copy()
y = e.copy()
z = f.copy()

# ~~~~~ STEP 4: Generate N2G sites (by midpoint formula) ~~~~~ #
a = zeros(x.size*0.5)
b = zeros(x.size*0.5)
c = zeros(x.size*0.5)
j = 0

for i in range(0, x.size, 2):
    xcom = ((x.item(i) + x.item(i+1)) * 0.5)
    ycom = ((y.item(i) + y.item(i+1)) * 0.5)
    zcom = ((z.item(i) + z.item(i+1)) * 0.5)
    a.itemset(j,round(xcom,6))
    b.itemset(j,round(ycom,6))
    c.itemset(j,round(zcom,6))
    j += 1

# ~~~~~ STEP 6: Generate N2N sites ~~~~~ #
blGN = 0.738           # LB nonpolar (default)
if pol_on:
    blGN = 0.745         # Bond length from the G to the N site (for LB polar model)
if mixrule == "WH":
    blGN = 0.789
if mixrule == "WH" and pol_on:
    blGN = 0.791
if S == "+S1" and mixrule == "LB":
    blGN = 0.790983
if S == "+S1" and mixrule == "LB" and pol_on:
    blGN = 0.790811
if S == "+S2" and mixrule == "LB":
    blGN = 0.788201
if S == "+S2" and mixrule == "LB" and pol_on:

```

```

blGN = 0.789864
if S == "+S3" and mixrule == "LB":
    blGN = 0.788072
if S == "+S3" and mixrule == "LB" and pol_on:
    blGN = 0.790914
if S == "+S4" and mixrule == "LB":
    blGN = 0.788475
if S == "+S4" and mixrule == "LB" and pol_on:
    blGN = 0.785869
if S == "+S5" and mixrule == "LB":
    blGN = 0.788258
if S == "+S5" and mixrule == "LB" and pol_on:
    blGN = 0.792170
if S == "+S6" and mixrule == "LB":
    blGN = 0.787394
if S == "+S6" and mixrule == "LB" and pol_on:
    blGN = 0.789624
if S == "+S7" and mixrule == "LB":
    blGN = 0.786035
if S == "+S7" and mixrule == "LB" and pol_on:
    blGN = 0.789760
if S == "+S8" and mixrule == "LB":
    blGN = 0.788488
if S == "+S8" and mixrule == "LB" and pol_on:
    blGN = 0.792476
if S == "+S9" and mixrule == "LB":
    blGN = 0.788909
if S == "+S9" and mixrule == "LB" and pol_on:
    blGN = 0.791495
if S == "+S10" and mixrule == "LB":
    blGN = 0.788053
if S == "+S10" and mixrule == "LB" and pol_on:
    blGN = 0.794897
if S == "+S1" and mixrule == "WH":
    blGN = 0.784579
if S == "+S1" and mixrule == "WH" and pol_on:
    blGN = 0.793307
if S == "+S2" and mixrule == "WH":
    blGN = 0.787110
if S == "+S2" and mixrule == "WH" and pol_on:
    blGN = 0.790668
if S == "+S3" and mixrule == "WH":
    blGN = 0.783514
if S == "+S3" and mixrule == "WH" and pol_on:
    blGN = 0.785596
if S == "+S4" and mixrule == "WH":
    blGN = 0.788218
if S == "+S4" and mixrule == "WH" and pol_on:
    blGN = 0.792883
if S == "+S5" and mixrule == "WH":
    blGN = 0.785007
if S == "+S5" and mixrule == "WH" and pol_on:
    blGN = 0.790113
if S == "+S6" and mixrule == "WH":
    blGN = 0.785943
if S == "+S6" and mixrule == "WH" and pol_on:
    blGN = 0.785050
if S == "+S7" and mixrule == "WH":
    blGN = 0.784082
if S == "+S7" and mixrule == "WH" and pol_on:
    blGN = 0.787391
if S == "+S8" and mixrule == "WH":
    blGN = 0.784705
if S == "+S8" and mixrule == "WH" and pol_on:
    blGN = 0.788553
if S == "+S9" and mixrule == "WH":
    blGN = 0.788614

```

```

if S == "+S9" and mixrule == "WH" and pol_on:
    blGN = 0.785022
if S == "+S10" and mixrule == "WH":
    blGN = 0.786695
if S == "+S10" and mixrule == "WH" and pol_on:
    blGN = 0.787558

t = (blGN / (breal*0.5))
N2N = zeros(x.size*3)
N2N.shape = (x.size,3)
j = 0
k = 0
for i in range(a.size):
    dx = a.item(i) - x.item(j)
    dy = b.item(i) - y.item(j)
    dz = c.item(i) - z.item(j)
    xnew = a.item(i) - t*dx
    ynew = b.item(i) - t*dy
    znew = c.item(i) - t*dz

    N2N.itemset((k,0),round(xnew,6))
    N2N.itemset((k,1),round(ynew,6))
    N2N.itemset((k,2),round(znew,6))
    k += 1

    xnew = a.item(i) + t*dx
    ynew = b.item(i) + t*dy
    znew = c.item(i) + t*dz

    N2N.itemset((k,0),round(xnew,6))
    N2N.itemset((k,1),round(ynew,6))
    N2N.itemset((k,2),round(znew,6))

    k += 1
    j += 2

# Refill three_site with current x,y,z and a,b,c positions
three_site = zeros(totalelems)
three_site.shape = (totalrows,3)
k = 0      # 0 --> 12 Total counter
l = 0      # 0 --> 8 x,y,z counter
for i in range(a.size):
    three_site.itemset((k,0),a.item(i))
    three_site.itemset((k,1),b.item(i))
    three_site.itemset((k,2),c.item(i))
    k += 1
    for j in range(2):
        three_site.itemset((k,0),x.item(l))
        three_site.itemset((k,1),y.item(l))
        three_site.itemset((k,2),z.item(l))
        l += 1
        k += 1
totalrows = x.size + a.size + x.size          # 2nd x.size is b/c there are as many "N" sites as there are "E" sites
totalelems = totalrows * 3
five_site = zeros(totalelems)
three_site = three_site.ravel()

k = 0      # Total counter
l = 0      # Counter for three_site
m = 0      # Counter for N2N
for i in range(a.size):
    for j in range(9):
        five_site.itemset(k,three_site.item(l))
        l += 1
        k += 1
    for j in range(6):
        five_site.itemset(k,N2N.item(m))
        m += 1

```

```

m += 1
k += 1

five_site.shape = (totalrows,3)
#print "\nDEBUG: 5-SITE N2 MODEL ~> EXPERIMENTAL DISTANCE OF %f ANGSTROMS" % breal
#print five_site
#print

# Write unit cell to file
for i in range(len(a)):
    for j in range(len(sitelist)):
        site.append(sitelist[j])

abc_new = str(a_current)
ac = 'unit_cell-' + abc_new + '.xyz'
ucell = open(ac, 'w')
ucell.write("%d\n\n" % len(five_site))
for i in range(len(five_site)):
    ucell.write("%s\t%f\t%f\t%f\n" % (site[i],five_site.item(i,0), five_site.item(i,1), five_site.item(i,2)))
ucell.close()
site = []

# ~~~~~ STEP 7: Minor Translate ~~~~~ #
coor_min_x = float(five_site[:, 0].min())
coor_min_y = float(five_site[:, 1].min())
coor_min_z = float(five_site[:, 2].min())

if (coor_min_x != 0 or coor_min_y != 0 or coor_min_z != 0):

    if coor_min_x < 0:      # Minor Translate: Ensure that a vertex is at (0,0,0)
        five_site[:, 0] += abs(coor_min_x)
    else:
        five_site[:, 0] -= coor_min_x

    if coor_min_y < 0:
        five_site[:, 1] += abs(coor_min_y)
    else:
        five_site[:, 1] -= coor_min_y

    if coor_min_z < 0:
        five_site[:, 2] += abs(coor_min_z)
    else:
        five_site[:, 2] -= coor_min_z

# ~~~~~ STEP 7: Generate SuperCell ~~~~~ #
# Determine initial a (ASSUMES A CUBIC LATTICE, SO a=b=c) XXX
x_min = float(a.min())
x_max = float(a.max())

#print "DEBUG: AFTER MINOR TRANSLATE"
#print five_site
#print

abc_orig = abs(x_min) + abs(x_max)
delta = abc_orig * 2.0
abc_orig = delta

#if(abc_orig != float(a_current)):
#    print "Error: Calculated lattice parameter a does not match the request (%f != %f). Exiting..." % (abc_orig, a_current)
#    sys.exit(1)

scellttmp_1 = open('scellttmp.xyz', 'w')
# Write initial cell to tmp file
for i in range(len(five_site)):
    scellttmp_1.write("%f\t%f\t%f\n" % (five_site.item(i,0), five_site.item(i,1), five_site.item(i,2)))

croot = (n_cells**(1/3.0))

```

```

tstcroot = str(croot)
tstcroot = tstcroot.split('.')

if(len(tstcroot)==2 and tstcroot[1]=='0'):
    croot = int(round(croot,0))

    # Replicate unit cell in x-dimension
    for i in range(croot-1):
        five_site[:, 0] += delta
        for j in range(len(five_site)):
            scelltmp_1.write("%f\t%f\t%f\n" % (five_site.item(j,0), five_site.item(j,1), five_site.item(j,2)))

    scelltmp_1.close()

    # Replicate supercell in y-dimension
    scelltmp_2 = open('scell_tmp_2.xyz', 'w')
    scelltmp_1 = open('scell_tmp.xyz', 'r')
    scelltmp = ft.read(scelltmp_1)
    scelltmp_1.close()
    for i in range(len(scelltmp)):
        scelltmp_2.write("%f\t%f\t%f\n" % (scelltmp.item(i,0), scelltmp.item(i,1), scelltmp.item(i,2)))

    for i in range(croot-1):
        scelltmp[:, 1] += delta
        for j in range(len(scelltmp)):
            scelltmp_2.write("%f\t%f\t%f\n" % (scelltmp.item(j,0), scelltmp.item(j,1), scelltmp.item(j,2)))

    scelltmp_2.close()

    # Replicate supercell in z-dimension
    scelltmp_3 = open('scell_tmp_3.xyz', 'w')
    scelltmp_2 = open('scell_tmp_2.xyz', 'r')
    scelltmp = ft.read(scelltmp_2)
    scelltmp_2.close()
    for i in range(len(scelltmp)):
        scelltmp_3.write("%f\t%f\t%f\n" % (scelltmp.item(i,0), scelltmp.item(i,1), scelltmp.item(i,2)))

    for i in range(croot-1):
        scelltmp[:, 2] += delta
        for j in range(len(scelltmp)):
            scelltmp_3.write("%f\t%f\t%f\n" % (scelltmp.item(j,0), scelltmp.item(j,1), scelltmp.item(j,2)))

    scelltmp_3.close()

    # Read fully replicated supercell into array
    scelltmp_3 = open('scell_tmp_3.xyz', 'r')
    scelltmp = ft.read(scelltmp_3)
    scelltmp_3.close()

    os.remove('scell_tmp.xyz')
    os.remove('scell_tmp_2.xyz')
    os.remove('scell_tmp_3.xyz')

else:
    print "Not a cube root"
    sys.exit(1)

# ^^^^^^^^^^^^^^^^^ STEP 8: Center SuperCell at origin ^^^^^^^^^^^^^#
coor_min_x = float(scelltmp[:, 0].min())
coor_min_y = float(scelltmp[:, 1].min())
coor_min_z = float(scelltmp[:, 2].min())
coor_max_x = float(scelltmp[:, 0].max())
coor_max_y = float(scelltmp[:, 1].max())
coor_max_z = float(scelltmp[:, 2].max())

check_x = (coor_min_x + coor_max_x) == 0.0
check_y = (coor_min_y + coor_max_y) == 0.0
check_z = (coor_min_z + coor_max_z) == 0.0

```

```

if not (check_x or check_y or check_z):
    #print "Centering supercell at origin.\n"
    half_max_x = coor_max_x * 0.5
    half_max_y = coor_max_y * 0.5
    half_max_z = coor_max_z * 0.5

    if coor_max_x > 0:
        scelltmp[:, 0] -= half_max_x
    else:
        scelltmp[:, 0] += abs(half_max_x)

    if coor_max_y > 0:
        scelltmp[:, 1] -= half_max_y
    else:
        scelltmp[:, 1] += abs(half_max_y)

    if coor_max_z > 0:
        scelltmp[:, 2] -= half_max_z
    else:
        scelltmp[:, 2] += abs(half_max_z)

# DEBUG
#print
#for i in range(len(scelltmp)):
#    print "%f\t%f\t%f" % (scelltmp.item(i,0), scelltmp.item(i,1), scelltmp.item(i,2))

# ~~~~~ STEP 8: Scale lattice parameters ~~~~~ #
nmols = int(len(scelltmp)/5.0)          # Based on a 5-site model
for i in range(nmols):
    for j in range(len(sitelist)):
        site.append(sitelist[j])
        molnum.append(i+1)
        mass.append(masslist[j])
        charge.append(qlist[j])
        polar.append(pollist[j])
        eps.append(epslist[j])
        sig.append(siglist[j])

scell = scelltmp.copy()

abclist.append(a_current)
basis = a_current * croot           # XXX This assumes that the number of requested supercells has a cubic root
print "Basis = %f" % basis

# ~~~~~ STEP 8: Create .pqr file for MPMC ~~~~~ #
if not os.path.exists(abc_new):
    os.mkdir(abc_new)

os.chdir(abc_new)
dircurr = os.getcwd()
# Move unit cell file to pwd
move(parentdir + '/' + ac, dircurr + '/' + ac)

coord_file = open(pqr_input, 'w')
raw_coords = open('coords.dat', 'w')
# NOTE: MAX NUMBER OF DECIMALS IN COORDS ALLOWED IS 3 (%8.3f, ideally %11.6f, or %21.16f).
for i in range(len(scell)):
    raw_coords.write("%11.6f\t%11.6f\t%11.6f\n" % (scell.item(i,0),scell.item(i,1),scell.item(i,2)))
    coord_file.write("ATOM %5d %4.45s %3.3s %1.1s %4d %11.6f%11.6f%9.5f%9.5f%9.5f%9.5f%9.5f\n" %
                    (i+1,site[i],'N2','M',molnum[i],scell.item(i,0),scell.item(i,1),scell.item(i,2),mass[i],charge[i],polar[i],eps[i],sig[i],0.0))

coord_file.write("END")
raw_coords.close()
coord_file.close()

# RMSD

```

```

crds_list.append(scell)                      # Store [nx3] numpy array in list
if len(crds_list) == 2:                      # We have 2 structures to compute RMSD
    # Do RMSD
    crds1 = crds_list[0]
    crds2 = crds_list[1]
    assert(crds1.shape[1] == 3)
    assert(crds1.shape == crds2.shape)
    n_vec = shape(crds1)[0]
    correlation_matrix = dot(transpose(crds1), crds2)
    v, s, w = linalg.svd(correlation_matrix)
    is_reflection = (linalg.det(v) * linalg.det(w)) < 0.0
    if is_reflection:
        s[-1] = -s[-1]
    E0 = sum(sum(crds1 * crds1)) + sum(sum(crds2 * crds2))
    rmsd_sq = (E0 - 2.0*sum(s)) / float(n_vec)
    rmsd_sq = max([rmsd_sq, 0.0])
    rmsd = sqrt(rmsd_sq)
    rmsd_list.append(rmsd)
    print "RMSD: %f" % rmsd
    crds_list.pop(0)                         # Remove old coordinates, sets current coords to list position 0

# Create input file
seed = random_integers(100000000,999999999)
infile = open(input_file, 'w')

infile.write('\n')
infile.write('job_name\t%s\n' % job_name)
infile.write('\n')
infile.write('ensemble\ttotal_energy\n')
if mixrule == "WH":
    infile.write('waldmanhagler\tton\n')
infile.write('\n')
infile.write('numsteps\tt1\n')
infile.write('corrtime\tt1\n')
infile.write('\n')
infile.write('rd_lrc\tton\n')
infile.write('rd_crystal\tton\n')
infile.write('rd_crystal_order\tt10\n')
infile.write('\n')
infile.write('basis1\tt%f\t0.0\t0.0\n' % basis)
infile.write('basis2\tt\t0.0\tt%f\t0.0\n' % basis)
infile.write('basis3\tt\t0.0\tt\t0.0\tt%f\n' % basis)
infile.write('\n')
if pol_on:
    infile.write('polarization\tton\n')
    infile.write('polar_damp_type\texponential\n')
    infile.write('polar_damp\t2.1304\n')
    infile.write('\n')
    infile.write('polar_ewald\tton\n')
    infile.write('ewald_kmax\tt10\n')
    #infile.write('polar_wolf_alpha\t0.13\n')
    infile.write('polar_palmo\tton\n')
    infile.write('polar_gs_ranked\tton\n')
    infile.write('polar_iterative\tton\n')
    infile.write('polar_max_iter\tt10\n')
    infile.write('polar_gamma\tt1.03\n')
    infile.write('\n')
    infile.write('wrapall\tton\n')
#infile.write('pqr_input\t%s\n' % pqr_input)
#infile.write('pqr_output\t%s\n' % pqr_output)
infile.write('pqr_restart\t%s\n' % pqr_restart)
#infile.write('energy_output\t%s\n' % energy_output)
infile.write('traj_output\t%s\n' % traj_output)
if pol_on:
    #infile.write('dipole_output\t%s\n' % dipole_output)
    #infile.write('field_output\t%s\n' % field_output)
    infile.write('\n')
infile.write('\n')

```

```

infile.close()

# Execute MPMC
#call("mpmc " + input_file + " >& runlog.log", shell=True)

os.chdir(parentdir)

# Print timing statistics

# Gamma Loop
def gamma_loop(mixrule, pol_on, n_cells, abc_min, abc_max, abc_del, x, breal, parentdir, trajfile, input_file, job_name, pqr_input,
               pqr_output, pqr_restart, energy_output, traj_output, dipole_output, field_output, sitelist, masslist, qlist, pollist, epslist,
               siglist, site, totalelems, totalrows):
    """Generates gamma-N2 crystals"""

    # First, statically set c lattice
    a2c_ratio = 3.957/5.109 # volume ratio between the a and c lattice constants, and is always our starting point

    # Must decide the initial parameters, depending on +-
    if inc_dec == "-":
        a_current = abc_max
        abc_max = 3.957
        c_current = a_current / a2c_ratio
    else:
        a_current = 3.957      # via experiment
        c_current = 5.109      # via experiment

    for a_current in arange(a_current, abc_max, abc_del):
        trajfile.write("REMARK Trajectory File created in Python by Chris Cioce\n")
        trajfile.write("REMARK a=b = %f A, c = %f A\n" % (a_current, c_current))
        print "*****"
        print "Current a=b = %f Ang / c = %f Ang" % (a_current, c_current)
        start_local = time.time()
        # Keep N cartesian coordinate static (specific code here...generalize)  <-- Determined by generating N2E sites and looking at xyz
        # of N closest to basis origin
        N_fixed = 0.388202 # Reciprocal space coordinate
        u = N_fixed / float(a_current)

        # Generate Unit Cell from Basis Vectors (bv)
        d = zeros(x.size)
        e = zeros(x.size)
        f = zeros(x.size)

        # N(1) : B_1
        bvx = u * a_current
        bvy = bvx
        bvz = 0.0
        d.itemset(0,bvx)
        e.itemset(0,bvy)
        f.itemset(0,bvz)

        # N(2) : B_2
        bvx = -bvx
        bvy =  bvx
        bvz =  0.0
        d.itemset(1,bvx)
        e.itemset(1,bvy)
        f.itemset(1,bvz)

        # N(3) : B_3
        bvx = (0.5+u) * a_current
        bvy = (0.5-u) * a_current
        bvz =  0.5   * c_current
        d.itemset(2,bvx)
        e.itemset(2,bvy)
        f.itemset(2,bvz)

```

```

# N(4) : B_4
bvx = (0.5-u) * a_current
bvy = (0.5+u) * a_current
bvz = 0.5 * c_current
d.itemset(3,bvx)
e.itemset(3,bvy)
f.itemset(3,bvz)

x = d.copy()
y = e.copy()
z = f.copy()

# ~~~~~ STEP 4: Generate N2G sites (by midpoint formula) ~~~~~ #
a = zeros(x.size*0.5)
b = zeros(x.size*0.5)
c = zeros(z.size*0.5)
j = 0

for i in range(0, x.size, 2):
    xcom = ((x.item(i) + x.item(i+1)) * 0.5)
    ycom = ((y.item(i) + y.item(i+1)) * 0.5)
    zcom = ((z.item(i) + z.item(i+1)) * 0.5)
    a.itemset(j,round(xcom,6))
    b.itemset(j,round(ycom,6))
    c.itemset(j,round(zcom,6))
    j += 1

# ~~~~~ STEP 6: Generate N2N sites ~~~~~ #
blGN = 0.738 # LB nonpolar (default)
if pol_on:
    blGN = 0.745 # Bond length from the G to the N site (for LB polar model)
    if mixrule == "WH":
        blGN = 0.789
    if mixrule == "WH" and pol_on:
        blGN = 0.791
    if S == "+S1" and mixrule == "LB":
        blGN = 0.790983
    if S == "+S1" and mixrule == "LB" and pol_on:
        blGN = 0.790811
    if S == "+S2" and mixrule == "LB":
        blGN = 0.788201
    if S == "+S2" and mixrule == "LB" and pol_on:
        blGN = 0.789864
    if S == "+S3" and mixrule == "LB":
        blGN = 0.788072
    if S == "+S3" and mixrule == "LB" and pol_on:
        blGN = 0.790914
    if S == "+S4" and mixrule == "LB":
        blGN = 0.788475
    if S == "+S4" and mixrule == "LB" and pol_on:
        blGN = 0.785869
    if S == "+S5" and mixrule == "LB":
        blGN = 0.788258
    if S == "+S5" and mixrule == "LB" and pol_on:
        blGN = 0.792170
    if S == "+S6" and mixrule == "LB":
        blGN = 0.787394
    if S == "+S6" and mixrule == "LB" and pol_on:
        blGN = 0.789624
    if S == "+S7" and mixrule == "LB":
        blGN = 0.786035
    if S == "+S7" and mixrule == "LB" and pol_on:
        blGN = 0.789760
    if S == "+S8" and mixrule == "LB":
        blGN = 0.788488
    if S == "+S8" and mixrule == "LB" and pol_on:

```

```

blGN = 0.792476
if S == "+S9" and mixrule == "LB":
    blGN = 0.788909
if S == "+S9" and mixrule == "LB" and pol_on:
    blGN = 0.791495
if S == "+S10" and mixrule == "LB":
    blGN = 0.788053
if S == "+S10" and mixrule == "LB" and pol_on:
    blGN = 0.794897
if S == "+S1" and mixrule == "WH":
    blGN = 0.784579
if S == "+S1" and mixrule == "WH" and pol_on:
    blGN = 0.793307
if S == "+S2" and mixrule == "WH":
    blGN = 0.787110
if S == "+S2" and mixrule == "WH" and pol_on:
    blGN = 0.790668
if S == "+S3" and mixrule == "WH":
    blGN = 0.783514
if S == "+S3" and mixrule == "WH" and pol_on:
    blGN = 0.785596
if S == "+S4" and mixrule == "WH":
    blGN = 0.788218
if S == "+S4" and mixrule == "WH" and pol_on:
    blGN = 0.792883
if S == "+S5" and mixrule == "WH":
    blGN = 0.785007
if S == "+S5" and mixrule == "WH" and pol_on:
    blGN = 0.790113
if S == "+S6" and mixrule == "WH":
    blGN = 0.785943
if S == "+S6" and mixrule == "WH" and pol_on:
    blGN = 0.785050
if S == "+S7" and mixrule == "WH":
    blGN = 0.784082
if S == "+S7" and mixrule == "WH" and pol_on:
    blGN = 0.787391
if S == "+S8" and mixrule == "WH":
    blGN = 0.784705
if S == "+S8" and mixrule == "WH" and pol_on:
    blGN = 0.788553
if S == "+S9" and mixrule == "WH":
    blGN = 0.788614
if S == "+S9" and mixrule == "WH" and pol_on:
    blGN = 0.785022
if S == "+S10" and mixrule == "WH":
    blGN = 0.786695
if S == "+S10" and mixrule == "WH" and pol_on:
    blGN = 0.787558

t = (blGN / (breal*0.5))
N2N = zeros(x.size*3)
N2N.shape = (x.size,3)
j = 0
k = 0
for i in range(a.size):
    dx = a.item(i) - x.item(j)
    dy = b.item(i) - y.item(j)
    dz = c.item(i) - z.item(j)
    xnew = a.item(i) - t*dx
    ynew = b.item(i) - t*dy
    znew = c.item(i) - t*dz

    N2N.itemset((k,0),round(xnew,6))
    N2N.itemset((k,1),round(ynew,6))
    N2N.itemset((k,2),round(znew,6))
    k += 1

```

```

xnew = a.item(i) + t*dx
ynew = b.item(i) + t*dy
znew = c.item(i) + t*dz

N2N.itemset((k,0),round(xnew,6))
N2N.itemset((k,1),round(ynew,6))
N2N.itemset((k,2),round(znew,6))

k += 1
j += 2

# Refill three_site with current x,y,z and a,b,c positions
three_site = zeros(totalelems)
three_site.shape = (totalrows,3)
k = 0 # 0 --> 12 Total counter
l = 0 # 0 --> 8 x,y,z counter
for i in range(a.size):
    three_site.itemset((k,0),a.item(i))
    three_site.itemset((k,1),b.item(i))
    three_site.itemset((k,2),c.item(i))
    k += 1
for j in range(2):
    three_site.itemset((k,0),x.item(l))
    three_site.itemset((k,1),y.item(l))
    three_site.itemset((k,2),z.item(l))
    l += 1
    k += 1

totalrows = x.size + a.size + x.size      # 2nd x.size is b/c there are as many "N" sites as there are "E" sites
totalelems = totalrows * 3
five_site = zeros(totalelems)
three_site = three_site.ravel()

k = 0 # Total counter
l = 0 # Counter for three_site
m = 0 # Counter for N2N
for i in range(a.size):
    for j in range(9):
        five_site.itemset(k,three_site.item(l))
        l += 1
        k += 1
    for j in range(6):
        five_site.itemset(k,N2N.item(m))
        m += 1
        k += 1

five_site.shape = (totalrows,3)
#print "\nDEBUG: 5-SITE N2 MODEL > EXPERIMENTAL DISTANCE OF %f ANGSTROMS" % breal
#print five_site
#print

# Write unit cell to file
for i in range(len(a)):
    for j in range(len(sitelist)):
        site.append(sitelist[j])

abc_new = str(a_current)
c_updated = str(c_current) #@CRC
ac = 'unit_cell-' + abc_new + '-' + c_updated + '.xyz'
ucell = open(ac, 'w')
ucell.write("%d\n\n" % len(five_site))
for i in range(len(five_site)):
    ucell.write("%s\t%f\t%f\t%f\n" % (site[i],five_site.item(i,0), five_site.item(i,1), five_site.item(i,2)))
ucell.close()
site = []

```

```

# ^^^^^^^^^^^^^^^^^ STEP 7: Minor Translate ^^^^^^^^^^^^^^^^^ #
coor_min_x = float(five_site[:, 0].min())
coor_min_y = float(five_site[:, 1].min())
coor_min_z = float(five_site[:, 2].min())

if (coor_min_x != 0 or coor_min_y != 0 or coor_min_z != 0):

    if coor_min_x < 0:          # Minor Translate: Ensure that a vertex is at (0,0,0)
        five_site[:, 0] += abs(coor_min_x)
    else:
        five_site[:, 0] -= coor_min_x

    if coor_min_y < 0:
        five_site[:, 1] += abs(coor_min_y)
    else:
        five_site[:, 1] -= coor_min_y

    if coor_min_z < 0:
        five_site[:, 2] += abs(coor_min_z)
    else:
        five_site[:, 2] -= coor_min_z

# ^^^^^^^^^ STEP 7: Generate SuperCell ^^^^^^ #
# Determine initial a, b and c
x_min = float(a.min())
x_max = float(a.max())
y_min = float(b.min())
y_max = float(b.max())
z_min = float(c.min())
z_max = float(c.max())

#print "DEBUG: AFTER MINOR TRANSLATE"
#print five_site
#print

a_orig = abs(x_min) + abs(x_max)
b_orig = abs(y_min) + abs(y_max)
c_orig = abs(z_min) + abs(z_max)
delta_a = a_orig * 2.0
delta_b = b_orig * 2.0
delta_c = c_orig * 2.0
a_orig = delta_a
b_orig = delta_b
c_orig = delta_c

sclltmp_1 = open('sclltmp.xyz', 'w')
# Write initial cell to tmp file
for i in range(len(five_site)):
    sclltmp_1.write("%f\t%f\t%f\n" % (five_site.item(i,0), five_site.item(i,1), five_site.item(i,2)))

croot = (n_cells**((1/3.0)))
tstcroot = str(croot)
tstcroot = tstcroot.split('.')
if(len(tstcroot)==2 and tstcroot[1]=='0'):
    croot = int(round(croot,0))

    # Replicate unit cell in x-dimension
    for i in range(croot-1):
        five_site[:, 0] += delta_a
    for j in range(len(five_site)):
        sclltmp_1.write("%f\t%f\t%f\n" % (five_site.item(j,0), five_site.item(j,1), five_site.item(j,2)))

    sclltmp_1.close()

    # Replicate supercell in y-dimension
    sclltmp_2 = open('sclltmp_2.xyz', 'w')
    sclltmp_1 = open('sclltmp.xyz', 'r')

```

```

scelltmp = ft.read(scelltmp_1)
scelltmp_1.close()
for i in range(len(scelltmp)):
    scelltmp_2.write("%f\t%f\t%f\n" % (scelltmp.item(i,0), scelltmp.item(i,1), scelltmp.item(i,2)))

    for i in range(croot-1):
        scelltmp[:, 1] += delta_b
    for j in range(len(scelltmp)):
        scelltmp_2.write("%f\t%f\t%f\n" % (scelltmp.item(j,0), scelltmp.item(j,1), scelltmp.item(j,2)))

    scelltmp_2.close()

# Replicate supercell in z-dimension
scelltmp_3 = open('scell_tmp_3.xyz', 'w')
scelltmp_2 = open('scell_tmp_2.xyz', 'r')
scelltmp = ft.read(scelltmp_2)
scelltmp_2.close()
for i in range(len(scelltmp)):
    scelltmp_3.write("%f\t%f\t%f\n" % (scelltmp.item(i,0), scelltmp.item(i,1), scelltmp.item(i,2)))

    for i in range(croot-1):
        scelltmp[:, 2] += delta_c
    for j in range(len(scelltmp)):
        scelltmp_3.write("%f\t%f\t%f\n" % (scelltmp.item(j,0), scelltmp.item(j,1), scelltmp.item(j,2)))

    scelltmp_3.close()

# Read fully replicated supercell into array
scelltmp_3 = open('scell_tmp_3.xyz', 'r')
scelltmp = ft.read(scelltmp_3)
scelltmp_3.close()

os.remove('scell_tmp.xyz')
os.remove('scell_tmp_2.xyz')
os.remove('scell_tmp_3.xyz')

else:
    print "Not a cube root"
    sys.exit(1)

# ^^^^^^^^^^^^^^^^^^ STEP 8: Center SuperCell at origin ^^^^^^^^^^^^^^ #
coor_min_x = float(scelltmp[:, 0].min())
coor_min_y = float(scelltmp[:, 1].min())
coor_min_z = float(scelltmp[:, 2].min())
coor_max_x = float(scelltmp[:, 0].max())
coor_max_y = float(scelltmp[:, 1].max())
coor_max_z = float(scelltmp[:, 2].max())

check_x = (coor_min_x + coor_max_x) == 0.0
check_y = (coor_min_y + coor_max_y) == 0.0
check_z = (coor_min_z + coor_max_z) == 0.0

if not (check_x or check_y or check_z):
    #print "Centering supercell at origin.\n"
    half_max_x = coor_max_x * 0.5
    half_max_y = coor_max_y * 0.5
    half_max_z = coor_max_z * 0.5

    if coor_max_x > 0:
        scelltmp[:, 0] -= half_max_x
    else:
        scelltmp[:, 0] += abs(half_max_x)

    if coor_max_y > 0:
        scelltmp[:, 1] -= half_max_y
    else:
        scelltmp[:, 1] += abs(half_max_y)

```

```

    if coor_max_z > 0:
        scelltmp[:, 2] -= half_max_z
    else:
        scelltmp[:, 2] += abs(half_max_z)

    # DEBUG
    #print
    #for i in range(len(scelltmp)):
    #    print "%f\t%f\t%f" % (scelltmp.item(i,0), scelltmp.item(i,1), scelltmp.item(i,2))

    # ~~~~~ STEP 8: Scale lattice parameters ~~~~~ #
    nmols = int(len(scelltmp)/5.0)      # Based on a 5-site model
    for i in range(nmols):
        for j in range(len(sitelist)):
            site.append(sitelist[j])
            molnum.append(i+1)
            mass.append(masslist[j])
            charge.append(qlist[j])
            polar.append(pollist[j])
            eps.append(epslist[j])
            sig.append(siglist[j])

    scell = scelltmp.copy()

    abclist.append(a_current)
    basis_c = c_current * croot
    basis_ab = a_current * croot      # XXX This assumes that the number of requested supercells has a cubic root
    print "Basis_ab = %f / c = %f" % (basis_ab, basis_c)

    # ~~~~~ STEP 8: Create .pqr file for MPMC ~~~~~ #
    if not os.path.exists(abc_new):
        os.mkdir(abc_new)

    os.chdir(abc_new)
    dircurr = os.getcwd()
    # Move unit cell file to pwd
    move(parentdir + '/' + ac, dircurr + '/' + ac)

    coord_file = open(pqr_input, 'w')
    raw_coords = open('coords.dat', 'w')
    # NOTE: MAX NUMBER OF DECIMALS IN COORDS ALLOWED IS 3 (%8.3f, ideally %11.6f, or %21.16f).
    for i in range(len(scell)):
        raw_coords.write("%11.6f\t%11.6f\t%11.6f\n" % (scell.item(i,0),scell.item(i,1),scell.item(i,2)))
        coord_file.write("ATOM %5d %4.4s %3.3s %1.1s %4d %11.6f%11.6f%11.6f%11.6f%11.6f%11.6f\n" %
                         (i+1,site[i],'N2','M',molnum[i],scell.item(i,0),scell.item(i,1),scell.item(i,2),mass[i],charge[i],polar[i],eps[i],sig[i],0.0))

    coord_file.write("END")
    raw_coords.close()
    coord_file.close()

    # RMSD
    crds_list.append(scell)      # Store [nx3] numpy array in list
    if len(crds_list) == 2:      # We have 2 structures to compute RMSD
        # Do RMSD
        crds1 = crds_list[0]
        crds2 = crds_list[1]
        assert(crds1.shape[1] == 3)
        assert(crds1.shape == crds2.shape)
        n_vec = shape(crds1)[0]
        correlation_matrix = dot(transpose(crds1), crds2)
        v, s, w = linalg.svd(correlation_matrix)
        is_reflection = (linalg.det(v) * linalg.det(w)) < 0.0
        if is_reflection:
            s[-1] = - s[-1]
        E0 = sum(sum(crds1 * crds1)) + sum(sum(crds2 * crds2))
        rmsd_sq = (E0 - 2.0*sum(s)) / float(n_vec)

```

```

rmsd_sq = max([rmsd_sq, 0.0])
rmsd = sqrt(rmsd_sq)
rmsd_list.append(rmsd)
print "RMSD: %f" % rmsd
crds_list.pop(0)      # Remove old coordinates, sets current coords to list position 0

# Create input file
seed = random_integers(100000000,999999999)
infile = open(input_file, 'w')
infile.write('\n')
infile.write('job_name\t%s\n' % job_name)
infile.write('\n')
infile.write('ensemble\tttotal_energy\n')
if mixrule == "WH":
    infile.write('waldmanhagler\tton\n')
infile.write('\n')
infile.write('numsteps\tt1\n')
infile.write('corrttime\tt1\n')
infile.write('\n')
infile.write('rd_lrc\tton\n')
infile.write('rd_crystal\tton\n')
infile.write('rd_crystal_order\t10\n')
infile.write('\n')
infile.write('basis1\tt%f\nt0.0\tt0.0\n' % basis_ab)
infile.write('basis2\tt\tn\tt%f\nt0.0\tt0.0\n' % basis_ab)
infile.write('basis3\tt\tn\tt0.0\tt\tn\tt%f\n' % basis_c)
infile.write('\n')
if pol_on:
    infile.write('polarization\tton\n')
    infile.write('polar_damp_type\ttexponential\n')
    infile.write('polar_damp\tt2.1304\n')
    infile.write('\n')
    infile.write('polar_ewald\tton\n')
    infile.write('ewald_kmax\tt10\n')
    #infile.write('polar_wolf_alpha\t0.13\n')
    infile.write('polar_palmo\tton\n')
    infile.write('polar_gs_ranked\tton\n')
    infile.write('polar_iterative\tton\n')
    infile.write('polar_max_iter\tt10\n')
    infile.write('polar_gamma\tt1.03\n')
    infile.write('\n')
    infile.write('wrapall\tton\n')
    #infile.write('pqr_input\t%s\n' % pqr_input)
    #infile.write('pqr_output\t%s\n' % pqr_output)
    infile.write('pqr_restart\t%s\n' % pqr_restart)
    #infile.write('energy_output\tt\tn\n' % energy_output)
    infile.write('traj_output\tt\tn\n' % traj_output)
    if pol_on:
        #infile.write('dipole_output\tt\tn\n' % dipole_output)
        #infile.write('field_output\tt\tn\n' % field_output)
        infile.write('\n')
        infile.write('\n')
        infile.close()

# Execute MPMC
#call("mpmc " + input_file + " >& runlog.log", shell=True)

os.chdir(parentdir)

# Print timing statistics

c_current = (a_current + abc_del) / a2c_ratio

if __name__ == '__main__':
    print
    # ~~~~~ STEP 1: Load in original coordinates ~~~~~ #

```

```

if (len(sys.argv) > 1):
    if (len(sys.argv) < 9):
        print "ERROR: Insufficient number of arguments. Usage: $ python solidN2.py [alpha | gamma] [np | pol] [LB | WH] [# UCs]
            [min_dist] [max_dist] [increment] [+ | -] [+SX | -SX]"
        sys.exit(1)

    phase    = sys.argv[1]
    pol_on   = sys.argv[2]
    mixrule  = sys.argv[3]
    n_cells  = int(sys.argv[4])
    abc_min  = float(sys.argv[5])
    abc_max  = float(sys.argv[6])
    abc_del  = float(sys.argv[7])
    inc_dec  = sys.argv[8]
    S        = sys.argv[9]

    if phase == "alpha":
        print "Alpha-N2 crystal will be generated."
    elif phase == "gamma":
        print "Gamma-N2 crystal will be generated."
    else:
        print "Specified phase is not compatible. Sorry, try again."
        sys.exit(1)

    if mixrule == "LB":
        print "Lorentz-Berthelot mixing rules will be used."
    elif mixrule == "WH":
        print "Waldman-Hagler mixing rules will be used."
    else:
        print "Specified mixing rule is not compatible. Sorry, try again."
        sys.exit(1)

    if pol_on == "pol":
        pol_on = 1      # Turn on polarization
    print "Polar job requested."
    else:
        pol_on = None # Turn off polarization
    print "Nonpolar job requested."
else:
    print "ERROR: Insufficient number of arguments. Usage: $ python solidN2.py [alpha | gamma] [np | pol] [LB | WH] [# UCs]
        [min_dist] [max_dist] [increment] [+ | -] [+SX | -SX]"
    sys.exit(1)

ncycles = (round(abc_max,0) - abc_min) / float(abc_del) # Number of cycles: not perfect, but won't mess up important things, just timing stats

if phase == "alpha":
    ifile = open('/home/ccioce/N2/Solid/script/alphaN2.UC', 'r')
elif phase == "gamma":
    ifile = open('/home/ccioce/N2/Solid/script/gammaN2.UC', 'r')

x, y, z = ft.read_columns(ifile)
ifile.close()

# DEBUG: Print initial unit cell to stdout
#print "\nINITIAL UNIT CELL:"
#for i in xrange(len(x)):
#    print x.item(i), y.item(i), z.item(i)
#print

# ~~~~~ STEP 2: Form distance list from combinatorics (nCr), as to correctly pair neighbors
# (obtained XYZ of unit cell may not have neighbors in correct order) ~~~~~ #
distlist = []
numdists = int((factorial(x.size) / (2*factorial(x.size-2))))
distarr = zeros(numdists)
k = 1
for i in range(x.size-1):
    for j in range(i+1, x.size):

```

```

dx = x.item(i) - x.item(j)
dy = y.item(i) - y.item(j)
dz = z.item(i) - z.item(j)
r2 = dx*dx + dy*dy + dz*dz
d = sqrt(r2)
d = round(d,6)           # Limit distance to 6 decimal places
distarr.itemset(k-1,d)
distlist.append([(i+1),(j+1),d])
#print "%d : %d->%d = %f" % (k, i+1, j+1, d)
k += 1

distarr.sort()
bondlen = round(distarr.item(0),6)      # Limit bondlength match to 6 decimals (not as strict as full 16 decimals matching)
look2 = Lookup(distlist)
pairslist = look2.get_key(bondlen)
numatoms = len(pairslist)*2

if numatoms != x.size:                 # Error check
    print "\nERROR in determining the correct pairs from bond lengths. Exiting..."
    sys.exit(1)

pairs = zeros(numatoms, int)
k = 0
for i in range(len(pairslist)):
    for j in range(2):
        pairs.itemset(k,pairslist[i][j])
    k += 1

pairs.shape = (len(pairslist),2)

# Armed with the list of atom pairs, re-arrange input file so as to list pairs on neighboring lines
d = zeros(x.size)
e = zeros(x.size)
f = zeros(x.size)
k = 0
for i in range(len(pairs)):
    for j in range(2):
        atnum = pairs[i][j]
        d.itemset(k,x.item(atnum-1))
        e.itemset(k,y.item(atnum-1))
        f.itemset(k,z.item(atnum-1))
    k += 1

x = d.copy()
y = e.copy()
z = f.copy()

# DEBUG
#print "Rearranged Unit Cell:"
#for i in range(len(x)):
#    print "%f\t%f\t%f" % (x.item(i), y.item(i), z.item(i))
#print

# ~~~~~ STEP 4: Generate N2G sites (by midpoint formula) ~~~~~ #
a = zeros(x.size*0.5)
b = zeros(x.size*0.5)
if phase == "alpha":
    c = zeros(x.size*0.5)
elif phase == "gamma":
    c = zeros(z.size*0.5)
j = 0

for i in range(0, x.size, 2):
    xcom = ((x.item(i) + x.item(i+1)) * 0.5)
    ycom = ((y.item(i) + y.item(i+1)) * 0.5)
    zcom = ((z.item(i) + z.item(i+1)) * 0.5)
    a.itemset(j,round(xcom,6))

```

```

b.itemset(j,round(ycom,6))
c.itemset(j,round(zcom,6))
j += 1

# Print COM coordinates to stdout
#for i in xrange(a.shape[0]):
#    print a.item(i), b.item(i), z.item(i)

# Print N2G + N2E coordinates (xyz) to stdout
#print "\nDEBUG: REARRANGED UNIT CELL + COM SITES:"
#k = 0
#for i in range(a.size):
#    print a.item(i), b.item(i), c.item(i)
#    for j in range(2):
#        print x.item(k), y.item(k), z.item(k)
#    k += 1
#print

# ~~~~~ STEP 5: Generate N2E sites (crystallographic data yields bond lengths less than 0.549 A) ~~~~~ #
# XXX Should have a check here that if the provided unit cell has the correct bond length, don't need to execute this step.
breal = 1.098
bhalf = breal * 0.5
t = (bhalf / (bondlen*0.5))
print "Bond length (via input) = %f Angstroms" % bondlen
print "Bond length (experiment) = %f Angstroms" % breal
print "Optimal t for G-->E: %f" % t
print

N2E = zeros(x.size*3)
j = 0
k = 0
for i in range(a.size):
    dx = a.item(i) - x.item(j)
    dy = b.item(i) - y.item(j)
    dz = c.item(i) - z.item(j)
    xnew = a.item(i) - t*dx
    ynew = b.item(i) - t*dy
    znew = c.item(i) - t*dz

    N2E.itemset(k,round(xnew,6))
    k += 1
    N2E.itemset(k,round(ynew,6))
    k += 1
    N2E.itemset(k,round(znew,6))
    k += 1

    xnew = a.item(i) + t*dx
    ynew = b.item(i) + t*dy
    znew = c.item(i) + t*dz

    N2E.itemset(k,round(xnew,6))
    k += 1
    N2E.itemset(k,round(ynew,6))
    k += 1
    N2E.itemset(k,round(znew,6))
    k += 1
    j += 2

N2E.shape = (x.size,3)

totalrows = x.size + a.size
totalelems = totalrows * 3
three_site = zeros(totalelems)

l = 0
m = 0
for i in range(a.size):

```

```

three_site.itemset(m,a.item(i))
m += 1
three_site.itemset(m,b.item(i))
m += 1
three_site.itemset(m,c.item(i))
m += 1
for j in range(2):
    for k in range(3):
        three_site.itemset(m,N2E.item(l,k))
        m += 1
    l += 1

three_site.shape = (totalrows,3)
# DEBUG
#print
#print "3-Site (Scaled Nitrogen for appropriate bond length):"
#for i in range(len(three_site)):
#    print "%d: %f\t%f\t%f" % (i+1, three_site.item(i,0), three_site.item(i,1), three_site.item(i,2))
#print

# Job Preparation (don't need to loop this)
parentdir = os.getcwd()
trajfile = open(parentdir + '/traj.pqr', 'w')

input_file    = 'n2.input'
job_name      = 'n2'
pqr_input     = 'n2.initial.pqr'      # This should ALWAYS be set. There are dependencies on this keyword within this script!
pqr_output    = 'n2.final.pqr'
pqr_restart   = 'n2.restart.pqr'
energy_output = 'n2.energy.dat'
traj_output   = 'n2.traj.pqr'
dipole_output = 'dipole.dat'
field_output  = 'field.dat'

sitelist      = [    'N2G',    'N2E',    'N2E',    'N2N',    'N2N' ]
masslist      = [ 0.00000, 14.00670, 14.00670, 0.00000, 0.00000 ]
qlist         = [ 1.04742, -0.52371, -0.52371, 0.00000, 0.00000 ]
pollist       = [     0.0,     0.0,     0.0,     0.0,     0.0 ]

# Set LB LJ parameters (default)
epslist       = [ 17.60293, 0.00000, 0.00000, 18.12772, 18.12772 ]
siglist       = [ 3.44522, 0.00000, 0.00000, 3.15125, 3.15125 ]

if pol_on: # Replace LB alpha and LJ parameters with polar ones
    pollist   = [ 1.49645, 0.45510, 0.45510, 0.0, 0.0 ]
    epslist   = [ 20.63650, 0.00000, 0.00000, 16.14200, 16.14200 ]
    siglist   = [ 3.42344, 0.00000, 0.00000, 3.16141, 3.16141 ]

# ...but if WH mixing rules are requested, then use these LJ parameters
if mixrule == "WH":
    epslist   = [ 32.01121, 0.00000, 0.00000, 13.09958, 13.09958 ]
    siglist   = [ 3.39190, 0.00000, 0.00000, 3.09821, 3.09821 ]

# ...if both WH and polarization are requested, use these parameters
if mixrule == "WH" and pol_on:
    print "Both WH and pol_on are set..."
    pollist   = [ 1.49645, 0.45510, 0.45510, 0.0, 0.0 ]
    epslist   = [ 32.10001, 0.00000, 0.00000, 12.72315, 12.72315 ]
    siglist   = [ 3.39450, 0.00000, 0.00000, 3.10175, 3.10175 ]

# LB Parameters for the S configuration
if S == "+S1" and mixrule == "LB":
    print "Using S1 / NP / LB parameters..."
    epslist   = [ 25.03351, 0.00000, 0.00000, 15.93472, 15.93472 ]
    siglist   = [ 3.45163, 0.00000, 0.00000, 3.06412, 3.06412 ]
if S == "+S1" and mixrule == "LB" and pol_on:
    print "Using S1 / Pol / LB parameters..."

```

```

epslist  = [ 27.02224,  0.00000,  0.00000, 14.55166, 14.55166 ]
siglist  = [ 3.43565,  0.00000,  0.00000,  3.08409,  3.08409 ]

if S == "+S2" and mixrule == "LB":
    print "Using S2 / NP / LB parameters..."
    epslist      = [ 26.01446,  0.00000,  0.00000, 15.44499, 15.44499 ]
    siglist      = [ 3.44119,  0.00000,  0.00000,  3.07386,  3.07386 ]
if S == "+S2" and mixrule == "LB" and pol_on:
    print "Using S2 / Pol / LB parameters..."
    epslist      = [ 28.19592,  0.00000,  0.00000, 14.07209, 14.07209 ]
    siglist      = [ 3.42602,  0.00000,  0.00000,  3.09286,  3.09286 ]

if S == "+S3" and mixrule == "LB":
    print "Using S3 / NP / LB parameters..."
    epslist      = [ 26.67119,  0.00000,  0.00000, 15.16878, 15.16878 ]
    siglist      = [ 3.43807,  0.00000,  0.00000,  3.07685,  3.07685 ]
if S == "+S3" and mixrule == "LB" and pol_on:
    print "Using S3 / Pol / LB parameters..."
    epslist      = [ 27.26533,  0.00000,  0.00000, 14.47356, 14.47356 ]
    siglist      = [ 3.43441,  0.00000,  0.00000,  3.08518,  3.08518 ]

if S == "+S4" and mixrule == "LB":
    print "Using S4 / NP / LB parameters..."
    epslist      = [ 27.00444,  0.00000,  0.00000, 14.94928, 14.94928 ]
    siglist      = [ 3.43335,  0.00000,  0.00000,  3.08084,  3.08084 ]
if S == "+S4" and mixrule == "LB" and pol_on:
    print "Using S4 / Pol / LB parameters..."
    epslist      = [ 25.84594,  0.00000,  0.00000, 15.07049, 15.07049 ]
    siglist      = [ 3.44659,  0.00000,  0.00000,  3.08442,  3.08442 ]

if S == "+S5" and mixrule == "LB":
    print "Using S5 / NP / LB parameters..."
    epslist      = [ 25.64425,  0.00000,  0.00000, 15.53200, 15.53200 ]
    siglist      = [ 3.44416,  0.00000,  0.00000,  3.07293,  3.07293 ]
if S == "+S5" and mixrule == "LB" and pol_on:
    print "Using S5 / Pol / LB parameters..."
    epslist      = [ 24.09181,  0.00000,  0.00000, 16.06115, 16.06115 ]
    siglist      = [ 3.46325,  0.00000,  0.00000,  3.06249,  3.06249 ]

if S == "+S6" and mixrule == "LB":
    print "Using S6 / NP / LB parameters..."
    epslist      = [ 24.42768,  0.00000,  0.00000, 16.19522, 16.19522 ]
    siglist      = [ 3.45331,  0.00000,  0.00000,  3.06721,  3.06721 ]
if S == "+S6" and mixrule == "LB" and pol_on:
    print "Using S6 / Pol / LB parameters..."
    epslist      = [ 28.10306,  0.00000,  0.00000, 14.07191, 14.07191 ]
    siglist      = [ 3.42862,  0.00000,  0.00000,  3.09229,  3.09229 ]

if S == "+S7" and mixrule == "LB":
    print "Using S7 / NP / LB parameters..."
    epslist      = [ 25.92003,  0.00000,  0.00000, 15.45715, 15.45715 ]
    siglist      = [ 3.44168,  0.00000,  0.00000,  3.07802,  3.07802 ]
if S == "+S7" and mixrule == "LB" and pol_on:
    print "Using S7 / Pol / LB parameters..."
    epslist      = [ 27.34094,  0.00000,  0.00000, 14.44778, 14.44778 ]
    siglist      = [ 3.43299,  0.00000,  0.00000,  3.08472,  3.08472 ]

if S == "+S8" and mixrule == "LB":
    print "Using S8 / NP / LB parameters..."
    epslist      = [ 26.17654,  0.00000,  0.00000, 15.31917, 15.31917 ]
    siglist      = [ 3.44049,  0.00000,  0.00000,  3.07569,  3.07569 ]
if S == "+S8" and mixrule == "LB" and pol_on:
    print "Using S8 / Pol / LB parameters..."
    epslist      = [ 27.26834,  0.00000,  0.00000, 14.49144, 14.49144 ]
    siglist      = [ 3.43809,  0.00000,  0.00000,  3.08059,  3.08059 ]

if S == "+S9" and mixrule == "LB":

```

```

print "Using S9 / NP / LB parameters..."
epslist      = [ 24.75185,  0.00000,  0.00000, 15.98923, 15.98923 ]
siglist      = [ 3.45095,  0.00000,  0.00000,  3.06702,  3.06702 ]
if S == "+S9" and mixrule == "LB" and pol_on:
    print "Using S9 / Pol / LB parameters..."
    epslist      = [ 26.96095,  0.00000,  0.00000, 14.74373, 14.74373 ]
    siglist      = [ 3.43808,  0.00000,  0.00000,  3.08184,  3.08184 ]

if S == "+S10" and mixrule == "LB":
    print "Using S10 / NP / LB parameters..."
    epslist      = [ 26.65155,  0.00000,  0.00000, 15.07068, 15.07068 ]
    siglist      = [ 3.43539,  0.00000,  0.00000,  3.07904,  3.07904 ]
if S == "+S10" and mixrule == "LB" and pol_on:
    print "Using S10 / Pol / LB parameters..."
    epslist      = [ 27.05609,  0.00000,  0.00000, 14.68874, 14.68874 ]
    siglist      = [ 3.44079,  0.00000,  0.00000,  3.07444,  3.07444 ]

# WH Parameters for the S configuration
if S == "+S1" and mixrule == "WH":
    print "Using S1 / NP / WH parameters..."
    epslist      = [ 27.61722,  0.00000,  0.00000, 15.19242, 15.19242 ]
    siglist      = [ 3.41827,  0.00000,  0.00000,  3.07809,  3.07809 ]
if S == "+S1" and mixrule == "WH" and pol_on:
    print "Using S1 / Pol / WH parameters..."
    epslist      = [ 31.03292,  0.00000,  0.00000, 13.47803, 13.47803 ]
    siglist      = [ 3.40469,  0.00000,  0.00000,  3.08985,  3.08985 ]

if S == "+S2" and mixrule == "WH":
    print "Using S2 / NP / WH parameters..."
    epslist      = [ 28.10847,  0.00000,  0.00000, 15.10849, 15.10849 ]
    siglist      = [ 3.41665,  0.00000,  0.00000,  3.07546,  3.07546 ]
if S == "+S2" and mixrule == "WH" and pol_on:
    print "Using S2 / Pol / WH parameters..."
    epslist      = [ 29.21930,  0.00000,  0.00000, 14.16880, 14.16880 ]
    siglist      = [ 3.41148,  0.00000,  0.00000,  3.08606,  3.08606 ]

if S == "+S3" and mixrule == "WH":
    print "Using S3 / NP / WH parameters..."
    epslist      = [ 27.26005,  0.00000,  0.00000, 15.30942, 15.30942 ]
    siglist      = [ 3.42032,  0.00000,  0.00000,  3.07774,  3.07774 ]
if S == "+S3" and mixrule == "WH" and pol_on:
    print "Using S3 / Pol / WH parameters..."
    epslist      = [ 28.19577,  0.00000,  0.00000, 14.51448, 14.51448 ]
    siglist      = [ 3.41610,  0.00000,  0.00000,  3.08761,  3.08761 ]

if S == "+S4" and mixrule == "WH":
    print "Using S4 / NP / WH parameters..."
    epslist      = [ 29.00184,  0.00000,  0.00000, 14.66607, 14.66607 ]
    siglist      = [ 3.41348,  0.00000,  0.00000,  3.07822,  3.07822 ]
if S == "+S4" and mixrule == "WH" and pol_on:
    print "Using S4 / Pol / WH parameters..."
    epslist      = [ 29.53328,  0.00000,  0.00000, 14.14841, 14.14841 ]
    siglist      = [ 3.41320,  0.00000,  0.00000,  3.08045,  3.08045 ]

if S == "+S5" and mixrule == "WH":
    print "Using S5 / NP / WH parameters..."
    epslist      = [ 27.28212,  0.00000,  0.00000, 15.35816, 15.35816 ]
    siglist      = [ 3.42104,  0.00000,  0.00000,  3.07580,  3.07580 ]
if S == "+S5" and mixrule == "WH" and pol_on:
    print "Using S5 / Pol / WH parameters..."
    epslist      = [ 29.74861,  0.00000,  0.00000, 13.93183, 13.93183 ]
    siglist      = [ 3.41115,  0.00000,  0.00000,  3.08681,  3.08681 ]

if S == "+S6" and mixrule == "WH":
    print "Using S6 / NP / WH parameters..."
    epslist      = [ 27.66681,  0.00000,  0.00000, 15.24137, 15.24137 ]
    siglist      = [ 3.41800,  0.00000,  0.00000,  3.07565,  3.07565 ]

```

```

if S == "+S6" and mixrule == "WH" and pol_on:
    print "Using S6 / Pol / WH parameters..."
    epslist      = [ 29.83458,  0.00000,  0.00000, 13.73245, 13.73245 ]
    siglist      = [  3.40462,  0.00000,  0.00000,  3.09956,  3.09956 ]

if S == "+S7" and mixrule == "WH":
    print "Using S7 / NP / WH parameters..."
    epslist      = [ 27.59946,  0.00000,  0.00000, 15.24748, 15.24748 ]
    siglist      = [  3.41814,  0.00000,  0.00000,  3.07852,  3.07852 ]
if S == "+S7" and mixrule == "WH" and pol_on:
    print "Using S7 / Pol / WH parameters..."
    epslist      = [ 29.66824,  0.00000,  0.00000, 13.95069, 13.95069 ]
    siglist      = [  3.40969,  0.00000,  0.00000,  3.09263,  3.09263 ]

if S == "+S8" and mixrule == "WH":
    print "Using S8 / NP / WH parameters..."
    epslist      = [ 27.63584,  0.00000,  0.00000, 15.19941, 15.19941 ]
    siglist      = [  3.41651,  0.00000,  0.00000,  3.07763,  3.07763 ]
if S == "+S8" and mixrule == "WH" and pol_on:
    print "Using S8 / Pol / WH parameters..."
    epslist      = [ 30.22857,  0.00000,  0.00000, 13.60798, 13.60798 ]
    siglist      = [  3.40488,  0.00000,  0.00000,  3.09790,  3.09790 ]

if S == "+S9" and mixrule == "WH":
    print "Using S9 / NP / WH parameters..."
    epslist      = [ 29.28207,  0.00000,  0.00000, 14.54166, 14.54166 ]
    siglist      = [  3.41080,  0.00000,  0.00000,  3.08026,  3.08026 ]
if S == "+S9" and mixrule == "WH" and pol_on:
    print "Using S9 / Pol / WH parameters..."
    epslist      = [ 28.76589,  0.00000,  0.00000, 14.14972, 14.14972 ]
    siglist      = [  3.41006,  0.00000,  0.00000,  3.09524,  3.09524 ]

if S == "+S10" and mixrule == "WH":
    print "Using S10 / NP / WH parameters..."
    epslist      = [ 29.14182,  0.00000,  0.00000, 14.51428, 14.51428 ]
    siglist      = [  3.41029,  0.00000,  0.00000,  3.08428,  3.08428 ]
if S == "+S10" and mixrule == "WH" and pol_on:
    print "Using S10 / Pol / WH parameters..."
    epslist      = [ 29.13527,  0.00000,  0.00000, 14.15119, 14.15119 ]
    siglist      = [  3.41287,  0.00000,  0.00000,  3.08987,  3.08987 ]

site      = []
molnum    = []
mass      = []
charge    = []
polar     = []
eps       = []
sig       = []
total_e   = []
total_e_norm = []
coul_e   = []
coul_e_norm = []
rd_e     = []
rd_e_norm = []
pol_e    = []
pol_e_norm = []
abclist   = []
rmsd_list = []
crds_list = []
cell_volume = []

# Main Loop over all a's
cc = 0      # Set coordinates counter (for RMSD)

# Time MPMC
times = zeros(0)      # Declare numpy array which will store each local time value for averaging
start_global = time.time()

```

```

# Call the appropriate loop
if phase == "alpha":
    al = alpha_loop(mixrule, pol_on, n_cells, abc_min, abc_max, abc_del, x, breal, parentdir, trajfile, input_file, job_name,
                    pqr_input, pqr_output, pqr_restart, energy_output, traj_output, dipole_output, field_output, sitelist, masslist, qlist,
                    pollist, epslist, siglist, site, totalelems, totalrows)
    print al
elif phase == "gamma":
    gl = gamma_loop(mixrule, pol_on, n_cells, abc_min, abc_max, abc_del, x, breal, parentdir, trajfile, input_file, job_name,
                    pqr_input, pqr_output, pqr_restart, energy_output, traj_output, dipole_output, field_output, sitelist, masslist, qlist,
                    pollist, epslist, siglist, site, totalelems, totalrows)
    print gl
else:
    print "Unsure as to how you managed to get to this point without prior erroring, but whatev...I have no loop for you,
          as your phase is not 'alpha' or 'gamma', so try again!"
    sys.exit(1)

# Execute MPMC
call("mpmc " + input_file + " >& runlog.log", shell=True)

# Append final coordinates to trajectory file
infinal = open(pqr_output, 'r')
finalpqr = finalpqr.read()
infinal.close()
stop = finalpqr.find('CONECT')
trajpqr = finalpqr[:stop]
trajfile.write('%' % trajpqr)
trajfile.write('ENDMDL\n')

# Process Energies
norm_fac = n_cells * 2
enfile = open(energy_output, 'r')
enfile.readline()
step, tote, coul, rd, pol, vdw, kin, tempe, partnum, sr, vol = ft.read_columns(enfile)
enfile.close()
total_e.append(tote.item(0))
total_e_norm.append(tote.item(0)/norm_fac)
coul_e.append(coul.item(0))
coul_e_norm.append(coul.item(0)/norm_fac)
rd_e.append(rd.item(0))
rd_e_norm.append(rd.item(0)/norm_fac)
pol_e.append(pol.item(0))
pol_e_norm.append(pol.item(0)/norm_fac)
cell_volume.append(vol.item(0))

os.chdir(parentdir)

# Print timing statistics
end = time.time()
elapsed_global = end - start_global
elapsed_local = end - start_local
times = append(times, elapsed_local) # Running list of local times
avg = float(average(times)) # Average local time (time/MPMC job)
pred_time = avg * ncycles # Predicted total time
eta = pred_time - elapsed_global # Predicted ETA to job completion
print "Global time = %08.2f s / %07.2f m / %5.2f h" % (elapsed_global, elapsed_global/60.0, elapsed_global/3600.0)
print "Local time = %08.2f s / %07.2f m / %5.2f h" % (elapsed_local, elapsed_local/60.0, elapsed_local/3600.0)
print "    ETA      = %08.2f s / %07.2f m / %5.2f h" % (eta, eta/60.0, eta/3600.0)

#####
# UPDATE c_current!!! #@CRC
c_current = (a_current + abc_del) / a2c_ratio #@CRC
#### #@CRC

# Print Energy & Stats
print

```

```

esummary      = open('energy_total_summary.dat', 'w')
esummary_norm = open('energy_total_summary_NORM.dat', 'w')
csummary      = open('energy_coul_summary.dat', 'w')
csummary_norm = open('energy_coul_summary_NORM.dat', 'w')
rsummary      = open('energy_rd_summary.dat', 'w')
rsummary_norm = open('energy_rd_summary_NORM.dat', 'w')
psummary      = open('energy_polar_summary.dat', 'w')
psummary_norm = open('energy_polar_summary_NORM.dat', 'w')
rmsdsum       = open('rmsd_summary.dat', 'w')
volum         = open('volumes.dat', 'w')

for i in range(len(abclist)):
    print abclist[i], total_e[i]
    esummary.write("%f\t%f\n" % (abclist[i], total_e[i]))
    esummary_norm.write("%f\t%f\n" % (abclist[i], total_e_norm[i]))
    csummary.write("%f\t%f\n" % (abclist[i], coul_e[i]))
    csummary_norm.write("%f\t%f\n" % (abclist[i], coul_e_norm[i]))
    rsummary.write("%f\t%f\n" % (abclist[i], rd_e[i]))
    rsummary_norm.write("%f\t%f\n" % (abclist[i], rd_e_norm[i]))
    psummary.write("%f\t%f\n" % (abclist[i], pol_e[i]))
    psummary_norm.write("%f\t%f\n" % (abclist[i], pol_e_norm[i]))
    volum.write( "%f\t%f\n" % (abclist[i], cell_volume[i]))

trajfile.close()
esummary.close()
csummary.close()
rsummary.close()
psummary.close()
esummary_norm.close()
csummary_norm.close()
rsummary_norm.close()
psummary_norm.close()

for i in range(len(rmsd_list)):
    rmsdsum.write("%f\n" % rmsd_list[i])

rmsdsum.close()

```

Appendix B

genSorptionTraj.py

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
#
#  @2014, Christian Cioce
#  Space Research Group
#  Department of Chemistry
#  University of South Florida
#
#  license: GNU LGPL
#
#  This library is free software; you can redistribute it and/or
#  modify it under the terms of the GNU Lesser General Public
#  License as published by the Free Software Foundation; either
#  version 2.1 of the License, or (at your option) any later version.

import os
import sys
import numpy as np
import math
import time

def initSys(fitGeom, C6, nSites):
    """Initializes the system coordinates and variables -- needed for the 'all' loop"""
    with open(fitGeom) as initPQR:
        if nSites == 1:
            head = [initPQR.next() for x in xrange(1)] # same as bash: $ head -1
        elif nSites == 3:
            head = [initPQR.next() for x in xrange(3)] # same as bash: $ head -3
        elif nSites == 5:
            head = [initPQR.next() for x in xrange(5)] # same as bash: $ head -5

        fitNum = fitGeom.split('-')[0]
        Damp = None
        Extrapolate = None
        if fitNum[-1] == 'D':
            Damp = 1
        elif fitNum[-1] == 'E':
            Extrapolate = 1

        site      = []
        coordsOrig = []
        mass      = []
        charge    = []
        alpha     = []
        epsilon   = []
        sigma     = []
        c6       = []
        c8       = []
        c10      = []

        for line in head:
```

```

for line in head:
    site.append(line.split()[2])
    coordsOrig.append(float(line.split()[6]))
    coordsOrig.append(float(line.split()[7]))
    coordsOrig.append(float(line.split()[8]))
    mass.append(float(line.split()[9]))
    charge.append(float(line.split()[10]))
    alpha.append(float(line.split()[11]))
    epsilon.append(float(line.split()[12]))
    sigma.append(float(line.split()[13]))

if C6:
    c6.append(float(line.split()[16]))
    c8.append(float(line.split()[17]))
    c10.append(float(line.split()[18]))


if nSites == 1:
    assert len(coordsOrig) == 3
    coordsOrig = np.array(coordsOrig)
    coordsOrig = coordsOrig.reshape(1,3)
elif nSites == 3:
    assert len(coordsOrig) == 9
    coordsOrig = np.array(coordsOrig)
    coordsOrig = coordsOrig.reshape(3,3)
elif nSites == 5:
    assert len(coordsOrig) == 15
    coordsOrig = np.array(coordsOrig)
    coordsOrig = coordsOrig.reshape(5,3)

for item in coordsOrig:
    assert item.item(1) == 0. and item.item(2) == 0.      # Check to ensure initial Y & Z components are 0

return fitNum, site, coordsOrig, mass, charge, alpha, epsilon, sigma, c6, c8, c10, Damp, Extrapol


def rotate(R, theta, r1):
    """Returns a rotated matrix about an axis (x, y, or z) and angle"""

    cosa = math.cos(math.radians(theta))
    chk = round(cosa, 6)
    if (chk == 0.000000):
        cosa = 0.0
    sina = math.sin(math.radians(theta))
    chk = round(sina, 6)
    if (chk == 0.000000):
        sina = 0.0

    if (R == 'X'):
        R = np.array([ [1.0, 0.0, 0.0], [0.0, cosa, -sina], [0.0, sina, cosa] ])
    elif (R == 'Y'):
        R = np.array([ [cosa, 0.0, sina], [0.0, 1.0, 0.0], [-sina, 0.0, cosa] ])
    elif (R == 'Z'):
        R = np.array([ [cosa, -sina, 0.0], [sina, cosa, 0.0], [0.0, 0.0, 1.0] ])
    else:
        print "Invalid rotation type %s...types are Rx, Ry, Rz. Exiting..." % R
        sys.exit(1)

    crdsNew = np.dot(R, r1)
    return crdsNew


def genInputFile(fitNum, config, heConfig, Po, WH, C6, Damp, Extrapol, mixing):
    """Creates MPMC Input file"""
    with open(fitNum + "-" + config + "-He" + str(heConfig) + ".input", "w") as mpmcInFile:
        mpmcInFile.write("job_name\t%s-He%d\n" % (fitNum, config, heConfig))
        mpmcInFile.write("ensemble\ttreplay\n")
        mpmcInFile.write("long_output\tton\n\n")

```

```

mpmcInFile.write("rd_lrc\t\t\toff\n\n")
if C6:
    mpmcInFile.write("disp_expansion\t\tton\n\n")
if Damp:
    mpmcInFile.write("damp_dispersion\t\tton\n\n")
if Extrap:
    mpmcInFile.write("damp_dispersion\t\tton\n\n")
    mpmcInFile.write("extrapolate_disp_coeffs\t\tton\n\n")
if WH:
    mpmcInFile.write("waldmanhagler\t\tton\n\n")
if mixing:
    mpmcInFile.write("schmidt_ff\t\tton\n\n")
if Po:
    mpmcInFile.write("polarization\t\tton\n\n")
    mpmcInFile.write("polar_damp_type\t\texponential\n\n")
    mpmcInFile.write("polar_damp\t\tt2.1304\n\n")
    mpmcInFile.write("polar_iterative\t\tton\n\n")
    mpmcInFile.write("polar_max_iter\t\tt5\n\n")
    mpmcInFile.write("polar_gs_ranked\t\tton\n\n")
    mpmcInFile.write("polar_palmo\t\tton\n\n")
mpmcInFile.write("pqr_input\t\tdev/null\n\n")
mpmcInFile.write("pqr_output\t\tdev/null\n\n")
mpmcInFile.write("pqr_restart\t\tdev/null\n\n")
mpmcInFile.write("traj_output\t\tdev/null\n\n")
mpmcInFile.write("traj_input\t\t%s-He%d.traj.pqr\n" % (fitNum, config, heConfig))
mpmcInFile.write("energy_output\t\t%s-He%d.energy.dat\n\n" % (fitNum, config, heConfig))
mpmcInFile.write("basis1\t1e9\t0.0\t0.0\n")
mpmcInFile.write("basis2\t0.0\t1e9\t0.0\n")
mpmcInFile.write("basis3\t0.0\t0.0\t1e9\n")

def genHe(minDist, heConfig, j):
    """Generates helium PQR"""
    rootDir = '/home/ccioce/Ne/Surf/heliums/'
    heCoords = []
    with open(rootDir + 'helium' + str(heConfig) + '.dat') as initHe:
        for line in initHe:
            heCoords.append(float(line.split()[6]))
            heCoords.append(float(line.split()[7]))
            heCoords.append(float(line.split()[8]))

    assert len(heCoords) % 3 == 0 # Check to ensure we can reshape array
    nAtoms = len(heCoords) / 3
    heCoords = np.array(heCoords)
    heCoords = heCoords.reshape(nAtoms,3)

    if heConfig == 1:
        assert heCoords.shape == (3, 3) # Config 1 must contain 3 He atoms
        preserveDist = heCoords.item(2,0) - minDist
        heCoords.itemset((1,0), j*0.5)
        heCoords.itemset((2,0), j+preserveDist)
    elif heConfig == 2:
        assert heCoords.shape == (4, 3) # Config 2 must contain 4 He atoms
        heCoords.itemset((2,0), j)
        heCoords.itemset((3,0), j)
    elif heConfig == 3 or heConfig == 4:
        assert heCoords.shape == (1, 3) # Config 3 & 4 must contain 1 He atom
        heCoords.itemset((0,0), j*0.5)
    elif heConfig == 5:
        assert heCoords.shape == (2, 3) # Config 5 must contain 2 He atoms
        heCoords.itemset((0,0), j*0.5)
        heCoords.itemset((1,0), j*0.5)
    elif heConfig == 6:
        assert heCoords.shape == (7, 3) # Config 6 must contain 7 He atoms. Config 6 is a mix of C1 & C2,
                                       # however in this case we want the C2 atoms to be static, so we only scale the C1's.
        preserveDist = heCoords.item(2,0) - minDist
        heCoords.itemset((1,0), j*0.5)

```

```

heCoords.itemset((2,0), j+preserveDist)
heCoords.itemset((5,0), j)
heCoords.itemset((6,0), j)
elif heConfig == 7:
    assert heCoords.shape == (5, 3) # Config 7 must contain 5 He atoms -- Planar Sorption
heCoords.itemset((0,0), j*0.5)
heCoords.itemset((1,0), j*0.5)
heCoords.itemset((2,0), j*0.5)
heCoords.itemset((3,0), j*0.5)
heCoords.itemset((4,0), j*0.5)
elif heConfig == 8:
    assert heCoords.shape == (5, 3) # Config 8 must contain 5 He atoms -- Pocket Sorption
heCoords.itemset((0,0), (j*0.5+3.0)-3.0)
heCoords.itemset((1,0), (j*0.5)-3.0)
heCoords.itemset((2,0), (j*0.5)-3.0)
heCoords.itemset((3,0), (j*0.5)-3.0)
heCoords.itemset((4,0), (j*0.5)-3.0)
elif heConfig == 9:
    assert heCoords.shape == (6, 3) # Config 9 must contain 6 He atoms -- Cage Sorption
heCoords.itemset((0,0), (j*0.5+3.0)-3.0)
heCoords.itemset((1,0), (j*0.5)-6.0)
heCoords.itemset((2,0), (j*0.5)-3.0)
heCoords.itemset((3,0), (j*0.5)-3.0)
heCoords.itemset((4,0), (j*0.5)-3.0)
heCoords.itemset((5,0), (j*0.5)-3.0)
elif heConfig == 10:
    assert heCoords.shape == (6, 3) # Config 10 must contain 6 He atoms -- Hexagon
heCoords.itemset((0,0), (j*0.5)-4.0)
heCoords.itemset((1,0), (j*0.5)-4.0)
heCoords.itemset((2,0), (j*0.5)-4.0)
heCoords.itemset((3,0), (j*0.5)-4.0)
heCoords.itemset((4,0), (j*0.5)-4.0)
heCoords.itemset((5,0), (j*0.5)-4.0)
elif heConfig == 11:
    assert heCoords.shape == (10, 3) # Config 11 must contain 10 He atoms
preserveDist = heCoords.item(1,0) - minDist
heCoords.itemset((1,0), j+preserveDist)
heCoords.itemset((2,0), (j*0.5)-1.5)
heCoords.itemset((3,0), (j*0.5)-1.5)
heCoords.itemset((4,0), (j*0.5)-1.5)
heCoords.itemset((5,0), (j*0.5)-1.5)
heCoords.itemset((6,0), (j*0.5)+1.5)
heCoords.itemset((7,0), (j*0.5)+1.5)
heCoords.itemset((8,0), (j*0.5)+1.5)
heCoords.itemset((9,0), (j*0.5)+1.5)
else:
    print "Invalid Helium Configuration: %d. Exiting..." % heConfig
    sys.exit(1)

return heCoords

```

```

# Define a lookup table for number of monomers in a given helium configuration, i.e. Helium 1 config contains 5 monomers in total.
dict = {'1': 5, '2': 6, '3': 3, '4': 3, '5': 4, '6': 9, '7': 7, '8': 6, '9': 7, '10': 7, '11': 12};

```

```

def CCSDT(mol1RotArray, mol2RotArray, helium, config, j, heConfig, nSites, site, cbs):
    if not os.path.exists("./CCSDT/" + config + "/" + str(j)):
        os.mkdir("./CCSDT/" + config + "/" + str(j))

    nmer = dict[str(heConfig)]
    nSitesT2P1 = (nSites * 2) + 1

    if cbs == 'QZTZ':
        function = ['q', 't']
    elif cbs == 'TZDZ':
        function = ['t', 'd']

```

```

else:
print "Invalid CBS Extrapolation Method: %s. Options are 'QZTZ' or 'TZDZ'. Exiting..." % cbs
sys.exit(1)

for basis in function:
    # Generate full nZ input file
    with open("./CCSDT/" + config + "/" + str(j) + "/" + basis + "z_" + str(nmer) + "mer.inp", "w") as nz:
        if cbs == 'QZTZ':
            nz.write("%MaxCore 8000\n! RHF aug-cc-pV" + str.capitalize(basis) + "Z CCSD(T) TightSCF\n* xyz 0 1\n")
        else:
            nz.write("%MaxCore 8000\n! RHF aug-cc-pV" + str.capitalize(basis) + "Z CCSD(T) VeryTightSCF\n* xyz 0 1\n")
    for i in range(nSites):
        if site[i] == 'H2E':
            nz.write("H\t%11.6f %11.6f %11.6f\n" % (mol1RotArray[i][0],mol1RotArray[i][1],mol1RotArray[i][2]))
    # for i in range(nSites,nSites*2):
    #     if site[i-nSites] == 'H2E':
    #         nz.write("H\t%11.6f %11.6f %11.6f\n" % (mol2RotArray[i-nSites][0],mol2RotArray[i-nSites][1],mol2RotArray[i-nSites][2]))
    if heConfig != 0:
        for i in range(nSitesT2P1,nSitesT2P1+helium.shape[0]):
            nz.write("He\t%11.6f %11.6f %11.6f\n" % (helium.item(i-nSitesT2P1,0),helium.item(i-nSitesT2P1,1),helium.item(i-nSitesT2P1,2)))
    nz.write("*\n%pal nprocs 8\nnend\n")

# Generate full nZ submit script -- Supported on SDSC's Trestles
    with open("./CCSDT/" + config + "/" + str(j) + "/" + basis + "z_" + str(nmer) + "mer.pbs", "w") as pbs:
        pbs.write("#!/bin/bash\n#PBS -q shared\n#PBS -l nodes=1:ppn=32\n")
        if basis == 'q':
            pbs.write("#PBS -l walltime=05:00:00\n")
        else:
            pbs.write("#PBS -l walltime=01:00:00\n")
        pbs.write("#PBS -N H2" + basis + "z" + str(nmer) + "M\n")
        pbs.write("#PBS -V\n#PBS -M ccioce@mail.usf.edu\n#PBS -m e\n\n")
        pbs.write("fname=" + basis + "z_" + str(nmer) + "mer\n\nexport OMP_NUM_THREADS=8\nncd\n"
                 "/oasis/scratch/trestles/cioce/$PBS_JOBID/ncp $PBS_O_WORKDIR/$fname.inp .\n\n"
                 "/home/cioce/orca_3_0_2_linux_x86-64/orca $fname.inp > $fname.log\n\n# Clean up\n"
                 "and copy back files\nrsync -auvh $fname.log $PBS_O_WORKDIR\n\n")

# Generate monomer nZ input files
for k in range(nmer):
    with open("./CCSDT/" + config + "/" + str(j) + "/" + basis + "z_monomer" + str(k+1) + ".inp", "w") as nz:
        if cbs == 'QZTZ':
            nz.write("%MaxCore 64000\n! RHF aug-cc-pV" + str.capitalize(basis) + "Z CCSD(T) TightSCF\n* xyz 0 1\n")
        else:
            nz.write("%MaxCore 64000\n! RHF aug-cc-pV" + str.capitalize(basis) + "Z CCSD(T) VeryTightSCF\n* xyz 0 1\n")
    for i in range(nSites):
        if site[i] == 'H2E':
            if k == 0:
                nz.write("H\t%11.6f %11.6f %11.6f\n" % (mol1RotArray[i][0],mol1RotArray[i][1],mol1RotArray[i][2]))
            else:
                nz.write("H :t%11.6f %11.6f %11.6f\n" % (mol1RotArray[i][0],mol1RotArray[i][1],mol1RotArray[i][2]))
    # for i in range(nSites,nSites*2):
    #     if site[i-nSites] == 'H2E':
    #         if k == 1:
    #             nz.write("H\t%11.6f %11.6f %11.6f\n" % (mol2RotArray[i-nSites][0],mol2RotArray[i-nSites][1],mol2RotArray[i-nSites][2]))
    #         else:
    #             nz.write("H :t%11.6f %11.6f %11.6f\n" % (mol2RotArray[i-nSites][0],mol2RotArray[i-nSites][1],mol2RotArray[i-nSites][2]))
    if heConfig != 0:
        for i in range(nSitesT2P1,nSitesT2P1+helium.shape[0]):
            if (k-1) == (i-nSitesT2P1):
                nz.write("He\t%11.6f %11.6f %11.6f\n" % (helium.item(i-nSitesT2P1,0),helium.item(i-nSitesT2P1,1),helium.item(i-nSitesT2P1,2)))
            else:
                nz.write("He :t%11.6f %11.6f %11.6f\n" % (helium.item(i-nSitesT2P1,0),helium.item(i-nSitesT2P1,1),helium.item(i-nSitesT2P1,2)))
    nz.write("\n")

# Generate monomer nZ submit scripts -- Supported on SDSC's Trestles
    with open("./CCSDT/" + config + "/" + str(j) + "/" + basis + "z_monomer" + str(k+1) + ".pbs", "w") as pbs:
        pbs.write("#!/bin/bash\n#PBS -q shared\n#PBS -l nodes=1:ppn=32\n")
        if basis == 'q':

```

```

        pbs.write("#PBS -l walltime=05:00:00\n")
    else:
        pbs.write("#PBS -l walltime=00:30:00\n")
    pbs.write("#PBS -N H2" + basis + "zM" + str(k+1) + "\n")
    pbs.write("#PBS -V\n#PBS -M ccioce@mail.usf.edu\n#PBS -m e\n\n")
    pbs.write("fnamee'" + basis + "z_monomer" + str(k+1) + "'\n\n"
              cd /oasis/scratch/trestles/cioce/$PBS_JOBID/ncp $PBS_O_WORKDIR/$fname.inp .\n\n
              /home/cioce/orca_3_0_2_linux_x86-64/orca $fname.inp > $fname.log\n\n# Clean up
              and copy back files\nrsync -auvh $fname.log $PBS_O_WORKDIR\n\n")

def genTraj(mol1RotArray, mol2RotArray, site, mass, charge, alpha, epsilon, sigma, config, heConfig, helium, fitNum, j,
           Po, WH, C6, c6, c8, c10, nSites, Damp, Extrapol, mixing, cbs):
    """Writes molecular configurations to file"""
    nSitesT2P1 = (nSites * 2) + 1
    if nSites == 1:
        sub = 0
    elif nSites == 3:
        sub = 4
    elif nSites == 5:
        sub = 8
    else:
        sub = 0

    with open(fitNum + "-" + config + "-He" + str(heConfig) + ".traj.pqr", "a") as trajFile:
        trajFile.write("REMARK File created via genSorptionTraj.py\nREMARK COM Separation = %.2f \n"
                      CRYST1 1000000000.000100000000.000100000000.000 90.00 90.00 90.00\n" % j)

    if C6:
        for i in range(nSites):
            trajFile.write("ATOM %5d %-4.45s %-3.3s %-1.1s %4d %11.6f %11.6f %11.6f
                           %8.5f %8.5f %8.5f %8.5f %8.5f %13.6f %13.6f %13.6f\n" %
                           (i+1,site[i],"H2","M",1,mol1RotArray[i][0],mol1RotArray[i][1],mol1RotArray[i][2],
                           mass[i],charge[i],alpha[i],epsilon[i],sigma[i],0.,0.,c6[i],c8[i],c10[i]))
            if heConfig != 0:
                if Extrapol:
                    for i in range(nSitesT2P1,nSitesT2P1+helium.shape[0]):
                        trajFile.write("ATOM %5d %-4.45s %-3.3s %-1.1s %4d %11.6f %11.6f %11.6f
                           %8.5f %8.5f %8.5f %8.5f %8.5f %13.6f %13.6f %13.6f\n" %
                           (i-1,"He","He","M",i-sub-1,helium.item(i-nSitesT2P1,0),helium.item(i-nSitesT2P1,1),
                           helium.item(i-nSitesT2P1,2),4.00260,0.,0.20494,4.498800,2.184500,0.,0.,1.407164,11.13635,107.964))
                else:
                    for i in range(nSitesT2P1,nSitesT2P1+helium.shape[0]):
                        trajFile.write("ATOM %5d %-4.45s %-3.3s %-1.1s %4d %11.6f %11.6f %11.6f
                           %8.5f %8.5f %8.5f %8.5f %8.5f %13.6f %13.6f %13.6f\n" %
                           (i-1,"He","He","M",i-sub-1,helium.item(i-nSitesT2P1,0),helium.item(i-nSitesT2P1,1),
                           helium.item(i-nSitesT2P1,2),4.00260,0.,0.20494,4.58224,2.21199,0.,0.,1.407164,11.13635,107.964))
            else:
                for i in range(nSites):
                    trajFile.write("ATOM %5d %-4.45s %-3.3s %-1.1s %4d %11.6f %11.6f %11.6f
                           %8.5f %8.5f %8.5f %8.5f %8.5f %13.6f %13.6f %13.6f\n" %
                           (i+1,site[i],"H2","M",1,mol1RotArray[i][0],
                           mol1RotArray[i][1],mol1RotArray[i][2],mass[i],charge[i],alpha[i],epsilon[i],sigma[i],0.,0.))

        for i in range(nSites,nSites*2):
            trajFile.write("ATOM %5d %-4.45s %-3.3s %-1.1s %4d %11.6f %11.6f %11.6f
                           %8.5f %8.5f %8.5f %8.5f %8.5f %13.6f %13.6f %13.6f\n" %
                           (i+1,site[i-nSites],"H2","M",2,mol2RotArray[i-nSites][0],
                           mol2RotArray[i-nSites][1],mol2RotArray[i-nSites][2],mass[i-nSites],charge[i-nSites],alpha[i-nSites],
                           epsilon[i-nSites],sigma[i-nSites],0.,0.))

        if heConfig != 0:
            for i in range(nSitesT2P1,nSitesT2P1+helium.shape[0]):
                trajFile.write("ATOM %5d %-4.45s %-3.3s %-1.1s %4d %11.6f %11.6f %11.6f
                               %8.5f %8.5f %8.5f %8.5f %8.5f %13.6f %13.6f %13.6f\n" %
                               (i,"He","He","M",i-sub,helium.item(i-nSitesT2P1,0),
                               helium.item(i-nSitesT2P1,1),helium.item(i-nSitesT2P1,2),4.00260,0.,0.094,10.22100,2.55651,0.,0.))

        trajFile.write("REMARK BOX BASIS[0] = 1000000000.0000000000000000 0.0000000000000000 0.0000000000000000\n"
                      REMARK BOX BASIS[1] = 0.0000000000000000 1000000000.0000000000000000 0.0000000000000000\n"
                      REMARK BOX BASIS[2] = 0.0000000000000000 0.0000000000000000 1000000000.0000000000000000\nENDMDL\n")

    # Also write out JUST helium coords to file
    if heConfig != 0:

```

```

        with open(fitNum + "-Heliums.traj.pqr", "a") as trajFile:
            trajFile.write("REMARK File created via genSorptionTraj.py\n")
            CRYST1 1000000000.0001000000000.0001000000000.000 90.00 90.00 90.00\n")
            for i in range(helium.shape[0]):
                if C6:
                    if Extrap:
                        trajFile.write("ATOM %5d %-4.45s %-3.3s %-1.1s %4d   %11.6f %11.6f %11.6f
                           %8.5f %8.5f %8.5f %8.5f %8.5f %13.6f %13.6f %13.6f\n" % (i+1,"He","He","M",i+1,
                           helium.item(i,0),helium.item(i,1),helium.item(i,2),4.00260,0.,0.20494,4.498800,2.184500,0.,0.,
                           1.407164,11.13635,107.964))
                    else:
                        trajFile.write("ATOM %5d %-4.45s %-3.3s %-1.1s %4d   %11.6f %11.6f %11.6f
                           %8.5f %8.5f %8.5f %8.5f %8.5f %13.6f %13.6f %13.6f\n" % (i+1,"He","He","M",i+1,
                           helium.item(i,0),helium.item(i,1),helium.item(i,2),4.00260,0.,0.20494,4.58224,2.21199,0.,0.,
                           1.407164,11.13635,107.964))
                else:
                    trajFile.write("ATOM %5d %-4.45s %-3.3s %-1.1s %4d   %11.6f %11.6f %11.6f
                           %8.5f %8.5f %8.5f %8.5f %8.5f\n" % (i+1,"He","He","M",i+1,helium.item(i,0),
                           helium.item(i,1),helium.item(i,2),4.00260,0.,0.094,10.22100,2.55651,0.,0.))
                    trajFile.write("REMARK BOX BASIS[0] = 1000000000.0000000000000000 0.0000000000000000 0.0000000000000000\n"
                                  "REMARK BOX BASIS[1] =      0.0000000000000000 10000000000.0000000000000000 0.0000000000000000\n"
                                  "REMARK BOX BASIS[2] =      0.0000000000000000 0.0000000000000000 10000000000.0000000000000000\nENDMDL\n")

# Also create corresponding MPMC Input File
genInputFile(fitNum, config, heConfig, Po, WH, C6, Damp, Extrap, mixing)

# Create CCSD(T) files
CCSDT(mol1RotArray, mol2RotArray, helium, config, j, heConfig, nSites, site, cbs)

def E(mol1, mol2):
    """Generates END-ON-END Geometry"""
    mol1RotArray = np.array(mol1)
    mol2RotArray = np.array(mol2)

    return mol1RotArray, mol2RotArray

def P(mol1, mol2):
    """Generates PARALLEL Geometry"""
    for rotAtom in mol1:
        mol1Rot = rotate('Z', 90, rotAtom)
        mol1RotList.append(mol1Rot)

    for rotAtom in mol2:
        mol2Rot = rotate('Z', 90, rotAtom)
        mol2RotList.append(mol2Rot)

    mol1RotArray = np.array(mol1RotList)
    mol2RotArray = np.array(mol2RotList)

    return mol1RotArray, mol2RotArray

def S(mol1, mol2):
    """Generates SLIP-PARALLEL Geometry"""
    for rotAtom in mol1:
        mol1Rot = rotate('Z', -45, rotAtom)
        mol1RotList.append(mol1Rot)

    for rotAtom in mol2:
        mol2Rot = rotate('Z', -45, rotAtom)
        mol2RotList.append(mol2Rot)

    mol1RotArray = np.array(mol1RotList)
    mol2RotArray = np.array(mol2RotList)

```

```

    return mol1RotArray, mol2RotArray

def T(mol1, mol2):
    """Generates T Geometry"""
    for rotAtom in mol2:
        mol2Rot = rotate('Z', 90, rotAtom)
        mol2RotList.append(mol2Rot)

    mol1RotArray = np.array(mol1)
    mol2RotArray = np.array(mol2RotList)

    return mol1RotArray, mol2RotArray

def X(mol1, mol2):
    """Generates X Geometry"""
    for rotAtom in mol1:
        mol1Rot = rotate('Z', 90, rotAtom)
        mol1RotList.append(mol1Rot)

    for rotAtom in mol2:
        mol2Rot = rotate('Y', 90, rotAtom)
        mol2RotList.append(mol2Rot)

    mol1RotArray = np.array(mol1RotList)
    mol2RotArray = np.array(mol2RotList)

    return mol1RotArray, mol2RotArray

if __name__ == '__main__':
    if len(sys.argv) == 10:
        fitGeom = sys.argv[1]
        config = sys.argv[2]
        minDist = float(sys.argv[3])
        maxDist = float(sys.argv[4])
        delDist = float(sys.argv[5])
        heConfig = int(sys.argv[6])
        extras = sys.argv[7]
        mixing = sys.argv[8]
        cbs = sys.argv[9]
    else:
        print "ERROR: Insufficient number of args. Usage: $python massOutToJobs.py [fit_geometry.pqr] [E|P|S|T|X|all|none]"
        print "[minDist] [maxDist] [increment] [He Configuration [INT]] [NPLB|NPWH|PoLB|PoWH|NPC6|PoC6] [default|schmidt]"
        print "[QZTZ|TZDZ]"
        sys.exit(1)

    Po = None
    WH = None
    C6 = None
    if extras == 'NPWH':
        WH = 1
    elif extras == 'PoLB':
        Po = 1
    elif extras == 'PoWH':
        Po = 1
        WH = 1
    elif extras == 'PoC6':
        Po = 1
        C6 = 1
    elif extras == 'NPC6':
        C6 = 1

    if mixing == 'schmidt':
        mixing = 1

```

```

else:
mixing = None

# Make CCSD(T) directory
if not os.path.exists("./CCSDT"):
os.mkdir("./CCSDT")

if config == 'all':
if not os.path.exists("./CCSDT/E"):
os.mkdir("./CCSDT/E")
if not os.path.exists("./CCSDT/P"):
os.mkdir("./CCSDT/P")
if not os.path.exists("./CCSDT/S"):
os.mkdir("./CCSDT/S")
if not os.path.exists("./CCSDT/T"):
os.mkdir("./CCSDT/T")
if not os.path.exists("./CCSDT/X"):
os.mkdir("./CCSDT/X")
elif config == 'none':
pass
else:
os.mkdir("./CCSDT/" + config)

# Determine the number of sites for this model
with open(fitGeom) as initPQR:
data = initPQR.read()
nSites = int(data.count("ATOM") / 2.)

fitNum, site, coordsOrig, mass, charge, alpha, epsilon, sigma, c6, c8, c10, Damp, Extrapolation = initSys(fitGeom, C6, nSites)

mol1 = coordsOrig
mol2 = np.copy(coordsOrig)
mol1RotList = []
mol2RotList = []
helium = None

if config == 'all':
config = 'E'
for j in np.arange(minDist, maxDist, delDist):
mol1RotArray, mol2RotArray = E(mol1, mol2)
mol2RotArray[:,0] += j
if heConfig != 0:
helium = genHe(minDist, heConfig, j)
genTraj(mol1RotArray, mol2RotArray, site, mass, charge, alpha, epsilon, sigma, config, heConfig, helium, fitNum,
j, Po, WH, C6, c6, c8, c10, nSites, Damp, Extrapolation, mixing, cbs)

config = 'P'
fitNum, site, coordsOrig, mass, charge, alpha, epsilon, sigma, c6, c8, c10, Damp, Extrapolation = initSys(fitGeom, C6, nSites)
mol1 = coordsOrig
mol2 = np.copy(coordsOrig)
mol1RotList = []
mol2RotList = []
for j in np.arange(minDist, maxDist, delDist):
mol1RotArray, mol2RotArray = P(mol1, mol2)
mol2RotArray[:,0] += j
if heConfig != 0:
helium = genHe(minDist, heConfig, j)
genTraj(mol1RotArray, mol2RotArray, site, mass, charge, alpha, epsilon, sigma, config, heConfig, helium, fitNum,
j, Po, WH, C6, c6, c8, c10, nSites, Damp, Extrapolation, mixing, cbs)

config = 'S'
fitNum, site, coordsOrig, mass, charge, alpha, epsilon, sigma, c6, c8, c10, Damp, Extrapolation = initSys(fitGeom, C6, nSites)
mol1 = coordsOrig
mol2 = np.copy(coordsOrig)
mol1RotList = []
mol2RotList = []
for j in np.arange(minDist, maxDist, delDist):

```

```

mol1RotArray, mol2RotArray = S(mol1, mol2)
mol2RotArray[:,0] += j
if heConfig != 0:
    helium = genHe(minDist, heConfig, j)
    genTraj(mol1RotArray, mol2RotArray, site, mass, charge, alpha, epsilon, sigma, config, heConfig, helium, fitNum,
             j, Po, WH, C6, c6, c8, c10, nSites, Damp, Extrap, mixing, cbs)

config = 'T'
fitNum, site, coordsOrig, mass, charge, alpha, epsilon, sigma, c6, c8, c10, Damp, Extrap = initSys(fitGeom, C6, nSites)
mol1 = coordsOrig
mol2 = np.copy(coordsOrig)
mol1RotList = []
mol2RotList = []
for j in np.arange(minDist, maxDist, delDist):
    mol1RotArray, mol2RotArray = T(mol1, mol2)
    mol2RotArray[:,0] += j
    if heConfig != 0:
        helium = genHe(minDist, heConfig, j)
        genTraj(mol1RotArray, mol2RotArray, site, mass, charge, alpha, epsilon, sigma, config, heConfig, helium, fitNum,
                 j, Po, WH, C6, c6, c8, c10, nSites, Damp, Extrap, mixing, cbs)

config = 'X'
fitNum, site, coordsOrig, mass, charge, alpha, epsilon, sigma, c6, c8, c10, Damp, Extrap = initSys(fitGeom, C6, nSites)
mol1 = coordsOrig
mol2 = np.copy(coordsOrig)
mol1RotList = []
mol2RotList = []
for j in np.arange(minDist, maxDist, delDist):
    mol1RotArray, mol2RotArray = X(mol1, mol2)
    mol2RotArray[:,0] += j
    if heConfig != 0:
        helium = genHe(minDist, heConfig, j)
        genTraj(mol1RotArray, mol2RotArray, site, mass, charge, alpha, epsilon, sigma, config, heConfig, helium, fitNum,
                 j, Po, WH, C6, c6, c8, c10, nSites, Damp, Extrap, mixing, cbs)

elif config == 'E':
    for j in np.arange(minDist, maxDist, delDist):
        mol1RotArray, mol2RotArray = E(mol1, mol2)
        mol2RotArray[:,0] += j
        if heConfig != 0:
            helium = genHe(minDist, heConfig, j)
            genTraj(mol1RotArray, mol2RotArray, site, mass, charge, alpha, epsilon, sigma, config, heConfig, helium, fitNum,
                     j, Po, WH, C6, c6, c8, c10, nSites, Damp, Extrap, mixing, cbs)

elif config == 'P':
    for j in np.arange(minDist, maxDist, delDist):
        mol1RotArray, mol2RotArray = P(mol1, mol2)
        mol2RotArray[:,0] += j
        if heConfig != 0:
            helium = genHe(minDist, heConfig, j)
            genTraj(mol1RotArray, mol2RotArray, site, mass, charge, alpha, epsilon, sigma, config, heConfig, helium, fitNum,
                     j, Po, WH, C6, c6, c8, c10, nSites, Damp, Extrap, mixing, cbs)

elif config == 'S':
    for j in np.arange(minDist, maxDist, delDist):
        mol1RotArray, mol2RotArray = S(mol1, mol2)
        mol2RotArray[:,0] += j
        if heConfig != 0:
            helium = genHe(minDist, heConfig, j)
            genTraj(mol1RotArray, mol2RotArray, site, mass, charge, alpha, epsilon, sigma, config, heConfig, helium, fitNum,
                     j, Po, WH, C6, c6, c8, c10, nSites, Damp, Extrap, mixing, cbs)

elif config == 'T':
    for j in np.arange(minDist, maxDist, delDist):
        mol1RotArray, mol2RotArray = T(mol1, mol2)
        mol2RotArray[:,0] += j
        if heConfig != 0:
            helium = genHe(minDist, heConfig, j)
            genTraj(mol1RotArray, mol2RotArray, site, mass, charge, alpha, epsilon, sigma, config, heConfig, helium, fitNum,
                     j, Po, WH, C6, c6, c8, c10, nSites, Damp, Extrap, mixing, cbs)

elif config == 'X':

```

```
for j in np.arange(minDist, maxDist, delDist):
    mol1RotArray, mol2RotArray = X(mol1, mol2)
    mol2RotArray[:,0] += j
    if heConfig != 0:
        helium = genHe(minDist, heConfig, j)
        genTraj(mol1RotArray, mol2RotArray, site, mass, charge, alpha, epsilon, sigma, config, heConfig, helium, fitNum,
                j, Po, WH, C6, c6, c8, c10, nSites, Damp, Extrapolation, mixing, cbs)
    else:
        print "Unrecognized configuration %s. Options are [E|P|S|T|X|all|none]. Exiting..." % config
        sys.exit(1)
```

Appendix C

Copyright Permissions

1/12/2015

Rightslink® by Copyright Clearance Center



RightsLink®

[Home](#) [Account Info](#) [Help](#)



ACS Publications
Most Trusted. Most Cited. Most Read.

RightsLink®

Title: A Polarizable and Transferable PHAST
N2 Potential for Use in Materials
Simulation

Logged in as:
Christian Cioce

[LOGOUT](#)

Author: Christian R. Cioce, Keith McLaughlin,
Jonathan L. Belof, et al

Publication: Journal of Chemical Theory and
Computation

Publisher: American Chemical Society

Date: Dec 1, 2013

Copyright © 2013, American Chemical Society

PERMISSION/LICENSE IS GRANTED FOR YOUR ORDER AT NO CHARGE

This type of permission/license, instead of the standard Terms & Conditions, is sent to you because no fee is being charged for your order. Please note the following:

- Permission is granted for your request in both print and electronic formats, and translations.
- If figures and/or tables were requested, they may be adapted or used in part.
- Please print this page for your records and send a copy of it to your publisher/graduate school.
- Appropriate credit for the requested material should be given as follows: "Reprinted (adapted) with permission from (COMPLETE REFERENCE CITATION). Copyright (YEAR) American Chemical Society." Insert appropriate information in place of the capitalized words.
- One-time permission is granted only for the use specified in your request. No additional uses are granted (such as derivative works or other editions). For any other uses, please submit a new request.

[BACK](#)

[CLOSE WINDOW](#)

Copyright © 2015 [Copyright Clearance Center, Inc.](#). All Rights Reserved. [Privacy statement](#).
Comments? We would like to hear from you. E-mail us at customerservice@copyright.com

<https://s100.copyright.com/AppDispatchServlet>

1/1

Figure C.1: Permission from the American Chemical Society to reuse the full article previously published in *J. Chem. Theory Comput.*, **2013**, 9 (12), 5550–5557.

1/12/2015

Rightslink® by Copyright Clearance Center



RightsLink®

Home

Account Info

Help



ACS Publications
Most Trusted. Most Cited. Most Read.

RightsLink®

Title: Characterization of Tunable Radical Metal–Carbenes: Key Intermediates in Catalytic Cyclopropanation

Logged in as:
Christian Ciocca

Author: Jonathan L. Belof, Christian R. Ciocca,
Xue Xu, et al

LOGOUT

Publication: Organometallics

Publisher: American Chemical Society

Date: May 1, 2011

Copyright © 2011, American Chemical Society

PERMISSION/LICENSE IS GRANTED FOR YOUR ORDER AT NO CHARGE

This type of permission/license, instead of the standard Terms & Conditions, is sent to you because no fee is being charged for your order. Please note the following:

- Permission is granted for your request in both print and electronic formats, and translations.
- If figures and/or tables were requested, they may be adapted or used in part.
- Please print this page for your records and send a copy of it to your publisher/graduate school.
- Appropriate credit for the requested material should be given as follows: "Reprinted (adapted) with permission from (COMPLETE REFERENCE CITATION). Copyright (YEAR) American Chemical Society." Insert appropriate information in place of the capitalized words.
- One-time permission is granted only for the use specified in your request. No additional uses are granted (such as derivative works or other editions). For any other uses, please submit a new request.

BACK

CLOSE WINDOW

Copyright © 2015 Copyright Clearance Center, Inc. All Rights Reserved. [Privacy statement](#).
Comments? We would like to hear from you. E-mail us at customerservice@copyright.com

<https://s100.copyright.com/AppDispatchServlet>

1/1

Figure C.2: Permission from the American Chemical Society to reuse the full article previously published in *Organometallics*, 2011, 30 (10), 2739–2746.

About the Author

Christian Cioce was born and raised in New Jersey, USA, and began post-secondary education at John Jay College of Criminal Justice in New York, NY. He spent two and a half years studying international criminal justice and forensic science before transferring to the University of South Florida in early 2007, where he ultimately received his B.S. in Chemistry, along with a minor in Biomedical Physics, in 2009. He began working with Professor Brian Space as an undergraduate, in 2008. Throughout graduate school, he discovered a new technique for detecting chemical weapons through nanomaterials via DTRA / C.S. Draper Laboratory sponsored research, he characterized a new class of radical metal-carbene complex, and contributed to several sponsored research projects. His research interests are wide, and encompass CBRN (Chemical, Biological, Radiological, and Nuclear) threat detection and decontamination, defense and national security threat mitigation, nanoporous and solid state materials, Metal-Organic Materials, sensing, sustainable energy, artificial intelligence and machine learning, predictive analytics, high performance computing, big data, data visualization, satellite imaging and cyber/information security, electronic structure methods, statistical mechanics, metal-carbenes and metallocporphyrins.