

**NOVEL FUNCTIONAL FORMS AND PARAMETERIZATION METHODS FOR AB
INITIO, INTERMOLECULAR FORCE FIELD DEVELOPMENT**

by

Mary J. Van Vleet

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Chemistry)

at the

UNIVERSITY OF WISCONSIN–MADISON

2017

Date of final oral examination: 08/15/17

The dissertation is approved by the following members of the Final Oral Committee:

J.R. Schmidt, Associate Professor, Chemistry

Clark R. Landis, Professor, Chemistry

Qiang Cui, Professor, Chemistry

Arun Yethiraj, Professor, Chemistry

Reid Van Lehn, Assistant Professor, Chemical and Biological Engineering

© Copyright by Mary J. Van Vleet 2017
All Rights Reserved

Soli Deo gloria.

ACKNOWLEDGMENTS

It is customary for authors of academic books to include in their prefaces statements such as this: "I am indebted to ... for their invaluable help; however, any errors which remain are my sole responsibility." Occasionally an author will go further. Rather than say that if there are any mistakes then he is responsible for them, he will say that there will inevitably be some mistakes and he is responsible for them....

Although the shouldering of all responsibility is usually a social ritual, the admission that errors exist is not — it is often a sincere avowal of belief. But this appears to present a living and everyday example of a situation which philosophers have commonly dismissed as absurd; that it is sometimes rational to hold logically incompatible beliefs.

— DAVID C. MAKINSON (1965)

Above is the famous “preface paradox,” which illustrates how to use the `wbepi` environment for epigraphs at the beginning of chapters. You probably also want to thank the Academy.

CONTENTS

Contents iii

List of Tables vii

List of Figures viii

Abstract ix

Published Work and Work in Preparation x

I Introduction 1

1 Introduction 2

1.1 *The Importance of Molecular Simulation* 2

2 Background 3

2.1 *Molecular Mechanics and the Theory of Intermolecular Forces* 3

2.1.1 The Many-Body Expansion 3

2.1.2 Energy Decomposition Schemes 3

Intramolecular Interactions 3

Electrostatics 3

Exchange 3

Induction 3

Dispersion 4

2.2 *Ab-Initio Force Field Development* 4

2.2.1 Electronic Structure Benchmarks 4

SAPT 4

Coupled-Cluster Methods 4

2.3 *ISA-based methods for force field development* 4

II Published Work 5

- 3 Isotropic Ab Initio Force Fields 6
- 4 Anisotropic Ab Initio Force Fields 7

III Unpublished Work 8

- 5 Ab Initio Force Fields using LMO-EDA 9
 - 5.1 *Preface* 9
 - 5.2 *Introduction* 10
 - 5.3 *Background and Motivation* 11
 - 5.4 *Parameterizing Coordinatively-Unsaturated (CUS)-Metal-Organic Framework (MOF) force fields with LMO-EDA* 14
 - 5.5 *Computational Methods* 18
 - 5.5.1 Partial Charge Determination 18
 - 5.5.2 Force Field Fitting 18
 - 5.6 *Results* 19
 - 5.6.1 Initial Force Field and Cluster Model Analysis 19
 - 5.6.2 Final Mg-MOF-74 CO₂ Adsorption Isotherm 23
 - 5.6.3 Transferability to Other Adsorption Isotherms 24
 - 5.6.4 Transferability to Other M-MOF-74 systems 25
 - 5.7 *Conclusions* 26
 - 5.8 *Future Work* 26
 - 5.A *Force Field Parameters for CO₂ and Mg-MOF-74* 29
 - 5.B *Simulation Parameters CO₂ Adsorption in Mg-MOF-74* 31
- 6 Benchmark Database for Ab Initio Force Field Development 33

IV Practical Matters 34

- 7 Workflow for Intermolecular Force Field Development 35

7.1	<i>Overview</i>	35
7.2	<i>Geometry Generation</i>	38
7.2.1	Guiding Principles	38
7.2.2	Theory	40
7.2.3	Practicals	41
7.3	<i>Symmetry-Adapted Perturbation Theory (SAPT) Benchmarks</i>	41
7.4	<i>CCSD(T) Calculations</i>	42
7.5	<i>Monomer-Based Parameterization</i>	43
7.5.1	Distributed Property Calculations using CamCASP	43
7.5.2	Multipoles	44
	Practicals	44
	Advanced Multipole Parameterization Options	45
7.5.3	ISA Exponents	47
7.5.4	Dispersion Coefficients	48
	Theory	48
	Iterative-Distributed Multipole Analysis (DMA)-pol	50
	Theory	50
	Practicals	50
	Iterated Stockholder Atoms (ISA)-pol	53
	Theory	53
	Practicals	54
	Comparison between iDMA-pol and ISA-pol	54
	Dispersion Coefficient Post-processing	56
7.5.5	Polarization Charges	57
	Theory	57
	Practicals	57
7.6	<i>Dimer-Based Parameterization</i>	58
7.A	<i>Input Scripts</i>	59
7.B	<i>Algorithm for Obtaining ISA Exponents</i>	62
8	POInter: A Program for Intermolecular Force Field Optimization	64

8.1	<i>Overview</i>	64
8.1.1	What is Parameter Optimizer for Inter -molecular Force Fields (POInter)?	64
8.1.2	What parameters do we get from POInter?	64
8.2	<i>Documentation</i>	66
8.3	<i>Examples</i>	66
8.A	<i>Input Files</i>	66

V Conclusions and Future Work **70**

9 Future Work 71

10 Conclusions 72

VICodes **73**

A Force Field Development Workflow 74

Bibliography 78

Acronyms 82

Glossary 83

LIST OF TABLES

7.1	Overview of ISA- and DMA-based methods for obtaining distributed monomer properties	44
7.2	Comparison between the iDMA-pol and ISA-pol methods.	55

LIST OF FIGURES

5.1	Model potential energy surface (PES) for interactions between CO ₂ and Mg-MOF-74	13
5.2	LMO-EDA vs. SAPT PES for the CO ₂ dimer	16
5.3	LMO-EDA vs. SAPT PES for the CO ₂ /Mg-MOF-74 dimer	17
5.4	Force field fitting quality for the Mg-MOF-74-small cluster	20
5.5	Model clusters for Mg-MOF-74	21
5.6	Force field fitting quality for Mg-MOF-74-Yu	24
5.7	Predicted CO ₂ Adsorption Isotherm for Mg-MOF-74	25
7.1	Generalized form of a PES showing the repulsive wall, minimum energy, and asymptotic regions.	38
7.2	Linear extrapolation algorithm for the methyl carbon in acetone. Depicted are (in legend order) Steps 1, 2, 4, and 5 in the extrapolation algorithm. Note that some portions of the 2 nd derivative extend off the graph; also note that most of logdens is located underneath the asymptotically-corrected curve.	63
A.1	The Semi-Automated Workflow for Force Field Development	77

**NOVEL FUNCTIONAL FORMS AND PARAMETERIZATION METHODS
FOR AB INITIO, INTERMOLECULAR FORCE FIELD DEVELOPMENT**

Mary J. Van Vleet

Under the supervision of Professor J.R. Schmidt

At the University of Wisconsin-Madison

FIXME: basically a placeholder; do not believe

I did some research, read a bunch of papers, published a couple myself, (pick one):

1. ran some experiments and made some graphs,
2. proved some theorems

and now I have a job. I've assembled this document in the last couple of months so you will let me leave. Thanks!

J.R. Schmidt

ABSTRACT

FIXME: basically a placeholder; do not believe

I did some research, read a bunch of papers, published a couple myself, (pick one):

1. ran some experiments and made some graphs,
2. proved some theorems

and now I have a job. I've assembled this document in the last couple of months so you will let me leave. Thanks!

PUBLISHED WORK AND WORK IN PREPARATION

- [48] Van Vleet, M. J.; Misquitta, A. J.; Stone, A. J.; Schmidt, J. R. *J. Chem. Theory Comput.* **2016**, *12*, 3851–3870.

Part I

Introduction

1 INTRODUCTION

1.1 The Importance of Molecular Simulation

This ref² is super cool!

What is molecular simulation? What types of problems can it solve? How does molecular simulation work? (Be sure to include solving Newton's EQs of motion and relevant details on the partition function and interaction energies!)

2 BACKGROUND

2.1 Molecular Mechanics and the Theory of Intermolecular Forces

What is a force field? What are the important components of a force field, and how do we model them?

2.1.1 The Many-Body Expansion

How do we break apart a force field into manageable pieces? Why does it make sense to break a force field into 2- and many-body components?

2.1.2 Energy Decomposition Schemes

Intramolecular Interactions

Brief commentary on the non-intermolecular portions of a force field

Electrostatics

Conceptual description of electrostatics: long-range multipoles and charge penetration

Exchange

Quantum-mechanically-based Pauli Exclusion. Theoretical grounds for exponential behavior

Induction

Charge transfer. Polarization. Polarization Damping.

Dispersion

Theoretical Formulation. Damping.

2.2 Ab-Initio Force Field Development

2.2.1 Electronic Structure Benchmarks

SAPT

General SAPT methodology. DFT-SAPT.

Coupled-Cluster Methods

CCSD(T). CCSD(T)-f12.

2.3 ISA-based methods for force field development

What is ISA? How can ISA be used to generate parameters for intermolecular force field development? What progress has been made from this approach?

Part II

Published Work

3 ISOTROPIC AB INITIO FORCE FIELDS

4 ANISOTROPIC AB INITIO FORCE FIELDS

Part III

Unpublished Work

5 AB INITIO FORCE FIELDS USING LMO-EDA

5.1 Preface

The preceding sections have been devoted to a development of various methodologies for ab initio intermolecular force field development, all generally assuming that Symmetry-Adapted Perturbation Theory (SAPT) can be used as a benchmark electronic structure theory. Critically, and especially given the developments discussed in Chapter 4, we can now usually expect our model force field energies to be within ~ 1 kJ/mol of the SAPT reference values! In spite of this success, this high precision between the model and SAPT energies can only lead to experimentally-accurate molecular simulation in the event that the SAPT energies themselves are accurate with respect to the exact underlying potential energy surface (PES). In practice, indeed, for systems where SAPT and CCSD(T) (a gold-standard electronic structure theory that closely matches the exact PES) disagree by several kJ/mol, there is little advantage in developing SAPT-based force fields with sub- kJ/mol precision. This limitation raises two fundamentally important questions. First, for what types of systems might we expect SAPT to be inaccurate? Second, for the systems where SAPT and the exact PES are in disagreement, how might we best modify our typical methodology for ab-initio force field development?

The purpose of this chapter is to partially address these two questions, all within the specific context of force field development for Coordinatively-Unsaturated (CUS) Metal-Organic Frameworks (MOFs). Importantly, the results presented here were gathered from 2012–2014, so some important advances (namely those presented in Chapters 3 and 4) have yet to be incorporated into the force fields presented here, probably to the detriment of the accuracy and transferability that might be possible with the LMO-EDA-based methodology. Should this project be picked up in the future, it will likely prove advantageous to refit the LMO-EDA-based force fields described herein to the functional forms and monomer-based parameters discussed in Chapter 4.

5.2 Introduction

Metal-Organic Frameworks (MOFs) are an increasingly important class of compounds, and are defined as porous materials containing inorganic nodes connected by organic linkers. Within this general motif, more than 20,000 compounds have been reported and studied,³ and this vast diversity of MOF materials shows great promise for chemical customization and optimization. Within the past two decades, a huge body of research has been devoted to the design and study of MOFs, and current applications range from gas separation and storage to catalysis and biomedical imaging.³

Somewhat recently, it has been discovered that so-called CUS MOFs can be created by activation of solvent-coordinated inorganic nodes to yield exposed (or 'open') metal sites.⁴⁻⁶ These CUS-MOFs have been shown to exhibit excellent uptakes and selectivities in a number of gas separation and storage problems,^{4,5,7} making this family of compounds an excellent target for future investigation and materials design. Owing to the vast scope of hypothetical CUS-MOF materials, however, and the number of chemically-distinct targets for gas separation/storage, it is unlikely that experiment alone can be used to screen for new and promising CUS-MOF materials.⁸ Rather, a combination of experiment and computational modeling will be required to identify (or possibly even rationally design) optimal CUS-MOFs.⁷⁻⁹

Despite the utility of computational studies, it remains challenging to develop molecular models for CUS-MOFs.^{6,8,9} Because the strong binding between metal and adsorbate leads to chemical environments substantially different from typical coordinatively-saturated MOFs, many standard force fields (such as UFF and DREIDING) that yield good predictions for these CS-MOFs can frequently (and substantially!) underpredict adsorption in CUS-MOFs.⁸⁻¹⁰ Importantly, these underpredictions are especially prominent at low pressures, where metal-adsorbate interactions dominate.⁸⁻¹⁰ While CUS-MOFs can sometimes be studied using quantum mechanical means,^{9,11,12} clearly new and improved force fields will be required to perform in-depth simulations and large-scale screenings of these materials, and

such studies are already being undertaken.^{13–16}

The goal of the present chapter is to present a general methodology for developing accurate and transferable force fields for CUS-MOFs. The current study is limited to a discussion of the MOF-74 series (a prototypical and well-studied CUS-MOF), however it is expected that the methods presented herein might also be applicable to other systems. After outlining this methodology (Sections 5.3 and 5.4), we next show how our force fields can be applied to accurately predict CO₂ adsorption isotherms in Mg-MOF-74. At the present time, we do not have results for other compounds in the M-MOF-74 series (M = Co, Cr, Cu, Fe, Mn, Ni, Ti, V, and Zn), largely as a result of technical challenges in the force field parameterization itself. We discuss these technical challenges in some detail, and conclude with our perspective on the challenges and opportunities associated with developing transferable force fields for the M-MOF-74 series and other similar CUS-MOF systems.

5.3 Background and Motivation

Prior work in our group has shown how, at least for coordinatively-saturated MOFs, accurate and transferable force fields can be generated for a wide variety of systems by fitting force field parameters on a component-by-component basis to reproduce an ab initio SAPT energy decomposition.^{17,18} While full details for this force field development methodology can be found in Refs. 18, 19, a short workflow is given here for Mg-MOF-74:

1. Generate a representative cluster model from which interaction parameters can be determined for each pairwise interaction. An example cluster, used to parameterize Mg–CO₂ interactions in Mg-MOF-74, is shown in Fig. 5.1.
2. Using DFT-SAPT (a variant of SAPT with monomer densities given by Density Functional Theory (DFT)), compute a series of representative dimer interaction energies for the model cluster. For the cluster model in Fig. 5.1, representative dimers were generated by varying the position of CO₂ with respect to the

MOF cluster, and the corresponding DFT-SAPT total interaction energies are shown for a subset of representative points.

3. To determine partial charges for the system, generate representative clusters (as described in Section 5.5) for each the organic ligand and the inorganic node, and perform a Distributed Multipole Analysis (DMA) analysis on each cluster to determine partial charges for the overall system.
4. For each component of the DFT-SAPT interaction energy, parameterize the relevant functional forms (as detailed in Ref. 18 and Section 5.4) to reproduce the DFT-SAPT component energy.

Once parameterized, these SAPT-based MOF force fields can be used for calculating individual adsorption isotherms or even for high-throughput screening.¹⁹

In the generation of force fields for CUS-MOFs, we expect that many of the advantages of the development methodology for coordinatively-saturated MOFs (such as the component-by-component based parameterization and protocol for partial charge determination) should also translate well to CUS-MOF materials. Nevertheless, there are two reasons why a SAPT-based methodology is non-ideal for generating CUS-MOF force fields. First, and as shown in Fig. 5.1 for a representative Mg-MOF-74 cluster model, by comparing to benchmark CCSD(T)-f12 calculations we have discovered SAPT to be in error for CUS-MOF-like systems. DFT-SAPT is known to struggle with highly ionic systems (relative to CCSD(T) or DFT methods),^{20,21} and so this error is perhaps not surprising. (Possible sources of the discrepancy between SAPT and CCSD(T)-f12 will be discussed in Section 5.8.) Nevertheless, and in the absence of fortuitous error cancellation, predictions from an ab initio force field can only be as good as the level of theory that they are parameterized against. Consequently, because SAPT underbinds CO₂ by a full 6 kJ/mol compared to CCSD(T)-f12, we would not expect to see good predictions for the CO₂ adsorption isotherm with a SAPT-based methodology. For CUS-MOFs and other similar systems, a new strategy for force field development is required.

As a second barrier to using a SAPT-based methodology, many of the compounds in the M-MOF-74 series are open-shell. Though this poses no fundamental

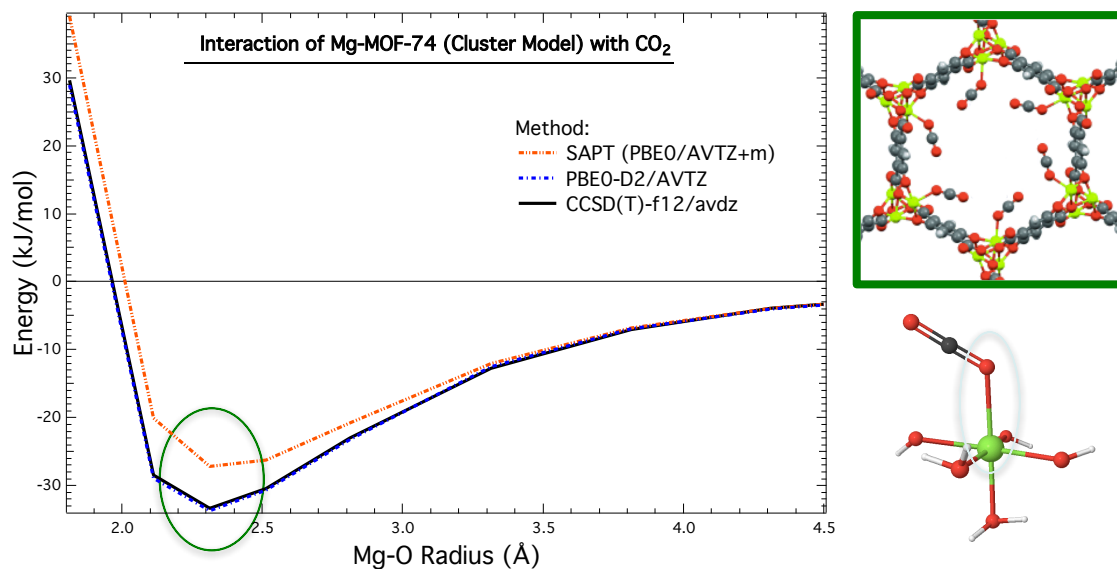


Figure 5.1: Model PES for interactions between CO₂ and Mg-MOF-74. (Left) Interaction energies between CO₂ and a cluster model of Mg-MOF-74 (shown bottom right), computed at a CCSD(T)-f12 (black), SAPT (orange), and/or PBE0-D2 (blue) level of theory. Discrepancies between SAPT and CCSD(T)-f12 in the minimum-energy region of the potential have been highlighted. (Top right) The structure of CO₂-bound Mg-MOF-74. (Bottom right) The structure of the cluster model used for Mg-MOF-74, where the circled atom pair indicates the relevant Mg-O radius from the x-axis in the leftmost figure.

issue, in practice most implementations of SAPT (aside from the seldom-used SAPT 2012 package developed in Krzysztof Szalewicz's group at Delaware) do not allow for computations of open-shell systems, and indeed SAPT-based studies of open-shell compounds are very rare.²² For these reasons, a new, open-shell-compatible electronic structure benchmark is highly preferable.

5.4 Parameterizing CUS-MOF force fields with LMO-EDA

Based on the results for Mg-MOF-74, it is clear that, at least for CUS-MOFs, a new methodology is required which simultaneously keeps the important advantages of the old development strategy (especially the component-by-component based parameterization, which is essential for generating transferable force fields) while overcoming the limitations of SAPT itself. Put differently, for CUS-MOFs we should seek a new electronic structure theory benchmark and associated Energy Decomposition Analysis (EDA) with the following qualities:

1. High accuracy with respect to CCSD(T)-f12 benchmark energies
2. Physically-meaningful energy decomposition into (at least) electrostatics, exchange, induction, and dispersion
3. Quantitative correspondence between the energy decompositions of SAPT and the new method for systems where total energies from SAPT, CCSD(T)-f12, and the new method agree

Assuming these three qualities are met, we expect to be able to generate force fields for CUS-MOFs that are both highly accurate and maximally-compatible with previous force fields developed for coordinatively-saturated MOF systems.

A substantial number of EDAs exist in the literature, and the interested reader is referred to Ref. 21 for a review and comparison of various popular methods. Aside from SAPT, which is a perturbative method, most EDAs are ‘variational’, meaning that the various energy components are calculated in stages from a series of constrained relaxations of the monomer wavefunctions into the optimized dimer wavefunction. For this reason, all variational EDAs are guaranteed to have total energies that match the result from a supermolecular interaction energy calculation. Furthermore, these EDAs are often implemented for wavefunction and DFT methods, thus allowing for significant flexibility (compared to the SAPT EDA) in terms of finding an EDA whose total energy closely matches CCSD(T)-f12. Indeed,

and as shown in Fig. 5.1, PBE0-D2 shows excellent agreement with CCSD(T)-f12 for a Mg-MOF-74 cluster model, and so any DFT-compatible EDA should meet our first criteria from above.

Although all variational EDAs yield the same total interaction energy for a given level of theory, many EDAs can differ substantially in terms of how this total energy is decomposed into chemically-meaningful components. At the time this research was completed, only a handful of variational EDAs distinguished each electrostatics, exchange, induction, and dispersion. (Notably, the recent second-generation ALMO-EDA²³ now separates their ‘frozen’ energy term into electrostatic, exchange, and dispersion components, and thus might be worth future investigation.) Of the popular EDA methods available in 2014, we found that LMO-EDA,^{24,25} GKS-EDA,²⁶ and PIEDA²⁷ decompose the total interaction in a manner philosophically similar to SAPT, and include each electrostatic, exchange, induction, and dispersion terms. These three methods thus meet our second criteria for an optimal energy decomposition scheme for CUS-MOFs, and complete formalisms and details for the methods can be found in Refs. 24–27.

As for the last criterion, that of maximum correspondence between SAPT and a variational EDA, we have performed component-by-component analyses to compare SAPT to both LMO-EDA and GKS-EDA. PIEDA is known to overestimate the relative magnitude of the polarization energy, compared to SAPT, and thus was not considered in detail.²¹ As for LMO-EDA and GKS-EDA (both of which are based on very similar theories, and tend to yield similar energy decompositions), we have in general found semi-quantitative to quantitative agreement with the SAPT energy decomposition, particularly for the electrostatic and exchange energies. Comparisons between LMO-EDA and SAPT are shown for the CO₂ dimer (Fig. 5.2) and for CO₂ interacting with a model Mg-MOF-74 compound (Fig. 5.3). GKS-EDA results are not shown, as the LMO-EDA and GKS-EDA results tend to be very similar, with the GKS-EDA results in slightly worse agreement with SAPT. For this reason, and because LMO-EDA does the best job of meeting our three criteria above, we choose in this work to use LMO-EDA as our new benchmark EDA for fitting CUS-MOF force fields.

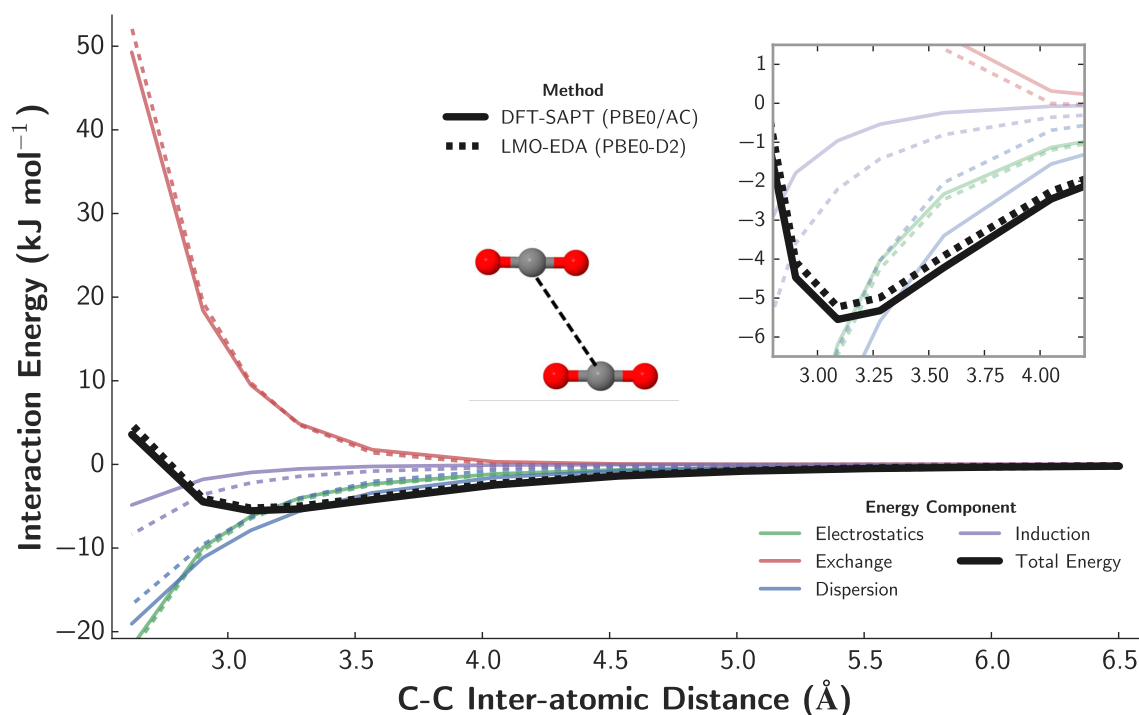


Figure 5.2: PES and associated energy decomposition for the slipped parallel geometry of the CO₂ dimer as a function of the C-C interatomic distance. The PES has been computed by both DFT-SAPT (PBE0) (solid lines) and LMO-EDA-PBE0-D2 (dashed lines), and each electrostatics (green), exchange (red), dispersion (blue), induction (purple), and total energy (black) components are displayed. Note that, for the DFT-SAPT energies, the δ HF contribution has been incorporated into the induction energy.

In addition to describing the advantages of the LMO-EDA method, it is worthwhile to overview some of its relevant shortcomings and limitations. As with most variational EDA methods,²¹ and especially for DFT-based methods, it becomes difficult to precisely assign and separate out the true ‘dispersion’ energy for a system. This limitation is also true of LMO-EDA, where the dispersion energy is defined as the difference in correlation energy between the monomer and dimer wavefunctions. (For density functionals employing Grimme’s -D dispersion correction, this correction is also added to the LMO-EDA dispersion energy.) For functionals that

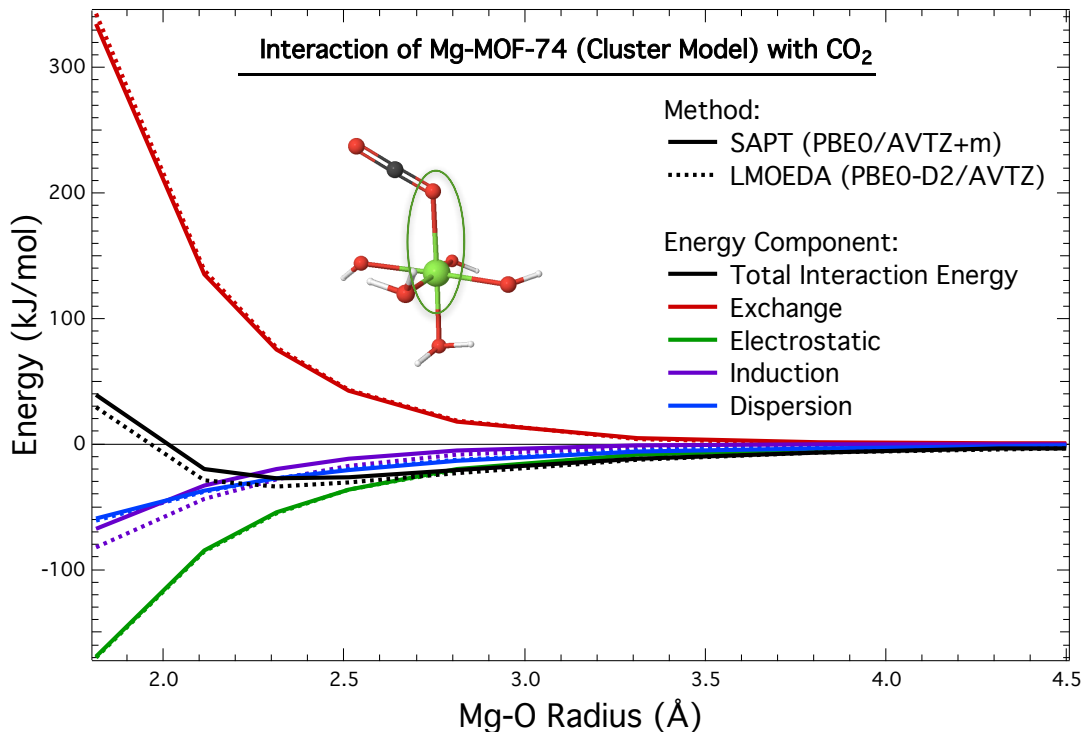


Figure 5.3: PES and associated energy decomposition for a $\text{CO}_2 + \text{MgO}_5$ cluster, as a function of the highlighted Mg-O interatomic distance. The PES has been computed by both DFT-SAPT (PBE0) (solid lines) and LMO-EDA-PBE0-D2 (dashed lines), and colors and labels for the energy decomposition are as in Fig. 5.2.

have a well-defined and theoretically-grounded distinction between the exchange and correlation functionals, the LMO-EDA energies tend to agree well with SAPT, and we have found good agreement (for instance) between SAPT and LMO-EDA-PBE0-D2. With other functionals, such as with our tests using the M06 functional, there is no separation between the exchange and correlation functionals, and LMO-EDA gives unphysical values for both the exchange and dispersion energies in this case. (Notably, GKS-EDA attempts to rectify this issue by changing the LMO-EDA formalism for dispersion. While this leads to qualitative agreement between SAPT and GKS-EDA for a wider variety of functionals, the quantitative agreement for the PBE0-D2 functional is somewhat worsened for the systems studied herein, and

we instead use LMO-EDA-PBE0-D2 for all results in this work.)

A second, and purely practical, limitation of LMO-EDA is its memory-intensive implementation in GAMESS. As will be discussed in detail later, calculations on a large (43 heavy atom) cluster model of Mg-MOF-74 were infeasible for us (using the Phoenix cluster in 2014) in all but the smallest VDZ basis set, and calculations on an identical cluster model of Co-MOF-74 could not be completed at all. For this reason, the LMO-EDA method is practically restricted to studies of smaller systems and/or basis sets.

5.5 Computational Methods

5.5.1 Partial Charge Determination

Partial charges for Mg-MOF-74 were determined in a manner analogous to Ref. 19 using the Q_{SBU} method. Two cluster models, one a hydrogen-capped DOBDC ligand environment, and one a capped MgO_5 inorganic chain, were constructed and analysed using a Distributed Multipole Analysis (DMA). The resulting DMA charges were then used to obtain charge parameters for the ligand and inorganic SBU, respectively. See Section 5.A for final charge parameters.

5.5.2 Force Field Fitting

Two types of force field functional forms were considered in this work. The first, a ‘single-exponential’ functional form, exactly matches that used in Ref. 28, with the exception that δHF parameters were not fit to the Mg atomtype. This fitting choice was due to the fact that LMO-EDA only provides a total induction term (rather than splitting into 2nd- and higher-order induction energies, as with SAPT).

For the ‘double-exponential’ functional form used to fit the Mg-MOF-74-Yu cluster model, the same functional form was used as in the single-exponential case, with the exception that two sets of short-range interaction parameters (labeled Mg and Du in Section 5.A) were assigned to the Mg atomic center. This effectively

meant that Mg was described by two separate exponential decays, thus enabling additional parameterization flexibility for the force fields discussed in Section 5.6.

In all cases, force fields were fit using the Fortran code described in the Appendix of Ref. 29.

5.6 Results

5.6.1 Initial Force Field and Cluster Model Analysis

Originally, we attempted to fit Mg parameters on the basis of a small, 6 heavy atom cluster ('Mg-MOF-74-small', see Fig. 5.4 for chemical structure), which we felt would be representative of the Mg environment in Mg-MOF-74. Using the functional forms discussed in Section 5.5, force field parameters were fit to reproduce LMO-EDA-PBE0-D2 energies for a variety of CO₂/Mg-MOF-74-small interactions, with results shown in Fig. 5.4. Though select interaction energies disagree by several kJ/mol between LMO-EDA-PBE0-D2 and the force field energies, overall the agreement is reasonable, and the force field correctly reproduces trends in the interaction energies without significant systematic error.

Based on the agreement between PBE0-D2 and the force field, as well as between PBE0-D2 and CCSD(T)-f12, we expected to obtain good CO₂ adsorption isotherm predictions for the Mg-MOF-74 system itself. By contrast, our computed isotherm substantially underpredicts the experimental adsorption at low pressures, where Mg-CO₂ interactions are known to dominate. This underprediction strongly suggests that we had originally underestimated the magnitude of the Mg-CO₂ binding, a result which we were then able to attribute to our choice of cluster model (*vide infra*).

Cluster models for the M-MOF-74 series have been investigated by several groups, and it has been found in general that computed binding energies are sensitive both to the size of the cluster model as well as the treatment of geometry relaxation effects.^{31,32} Consequently, we calculated the CO₂ binding energies and geometries of both our original Mg-MOF-74-small cluster as well as for two larger

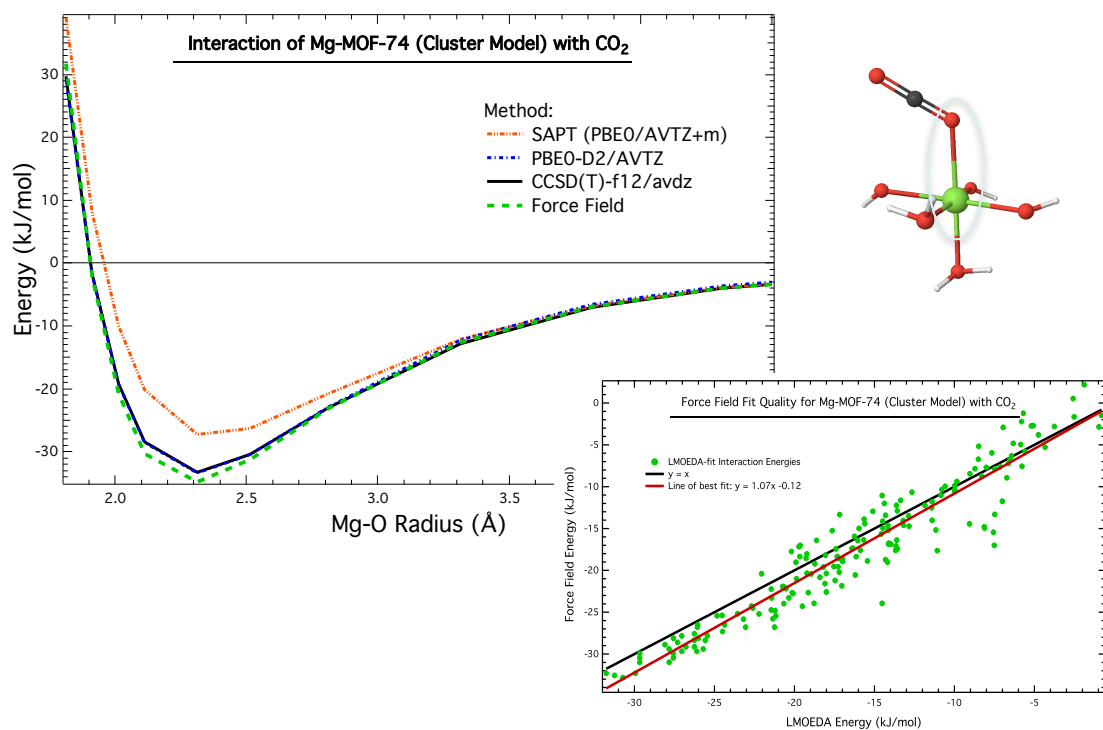
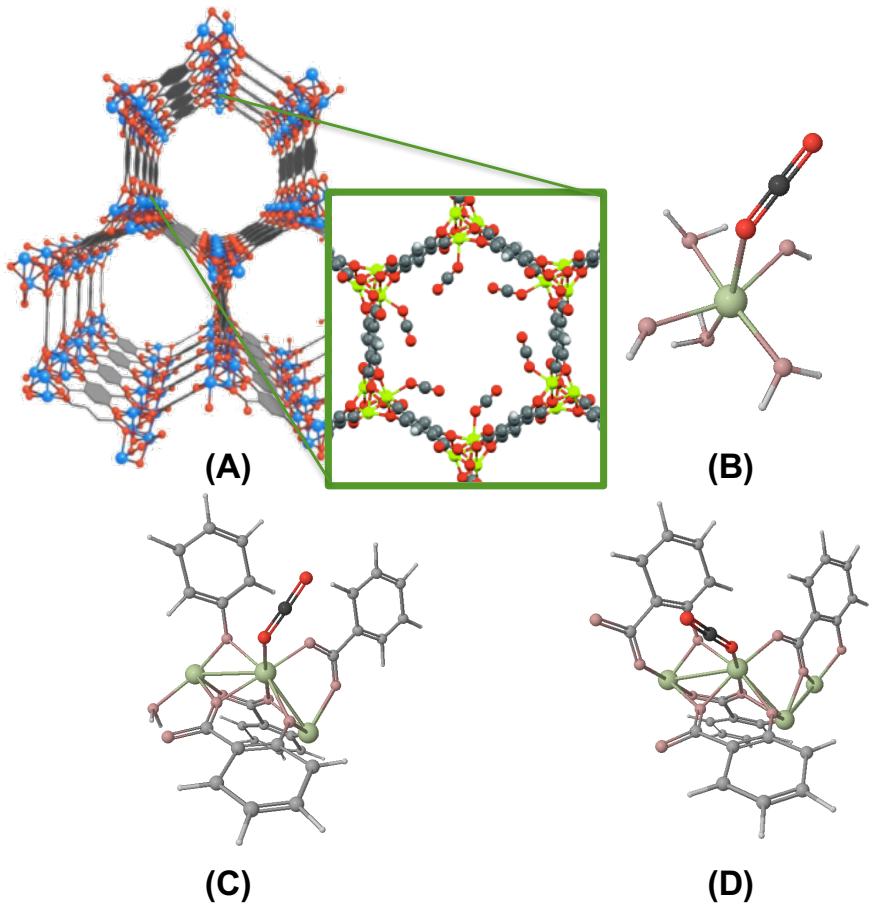


Figure 5.4: Force field fitting quality for the Mg-MOF-74-small cluster. (Top left) Various electronic structure benchmarks for Mg-MOF-74-small along with the classical potential. Each DFT-SAPT (orange dot-dashed), PBE0-D2 (blue dot-dashed), CCSD(T)-f12 (black solid), and the LMO-EDA-based force field (green dashed) are shown as a function of the Mg-O interatomic distance (non-bonding pair highlighted at top right). (Bottom right) Force field fit quality, as benchmarked against LMO-EDA-PBE0-D2, for a semi-random set of dimer configurations of the Mg-MOF-74-small cluster model interacting with CO₂. The black line establishes the $y = x$ benchmark, and the red line represents the line of best fit.



Model	CO ₂ Binding Energy (kJ/mol)	Mg–O Interatomic Distance (Å)	Mg–O–C Tilt Angle (°)
A ¹¹	-41.5	2.31	129
B	-23.3	2.31	122
C	-31.4	2.28	123
D	-41.7	2.20	149

Figure 5.5: Various structures and cluster models for Mg-MOF-74 interacting with CO₂. (A) Full periodic Mg-MOF-74 structure with inset showing adsorbed CO₂ positions. (B) Mg-MOF-74-small cluster, containing 6 heavy atoms (not including CO₂). (C) Yu et al. cluster model for Mg-MOF-74, denoted in text as Mg-MOF-74-Yu. (D) Dzubak et al. cluster model for Mg-MOF-74, denoted in text as Mg-MOF-74-Dzubak. All cluster models as shown with optimized CO₂ positions, and bond lengths and angles for adsorbed CO₂ are given in the bottom table. Data for (A) was taken from Valenzano et al. using a B3LYP-D level of theory,¹¹ whereas data for (B-D) was computed in this work using PBE0-D2. Finally, note that the binding energy for (A) includes framework geometry relaxation effects, whereas (B-D) were computed using semi-rigid cluster geometries and only optimizing the CO₂ position and exposed MgO₅ pocket.

clusters developed in Refs. 6, 30. These latter two clusters, respectively denoted Mg-MOF-74-Yu and Mg-MOF-74-Dzubak, are the same size (each with 60 atoms), but have distinct stoichiometries and geometries. To test the influence of model cluster on the CO₂ binding energy/geometry, we performed two sets of optimizations of the Mg-MOF-74-Yu and Mg-MOF-74-Dzubak clusters: one in which only the CO₂ position was optimized, and one in which the exposed MgO₅ pocket was additionally relaxed. Binding geometries were relatively insensitive to the geometry relaxation, though binding energies varied by 2-5 kJ/mol, in agreement with other studies that have tested geometry relaxation effects.³¹ Results for the CO₂ + MgO₅ relaxation are shown in Fig. 5.5.

Of the three studied cluster models, both Mg-MOF-74-small and Mg-MOF-74-Yu correctly reproduce the Mg-O interatomic distance and Mg-O-C tilt angle. These geometrical parameters arise primarily from electrostatic interactions between CO₂ and the MgO₅ pocket,¹¹ suggesting that both of these models capture such important interaction features. By contrast, the Mg-MOF-74-Dzubak model predicts a substantially shorter binding distance and increased tilt angle, both in contrast to results from the periodic system. In part, these deficiencies can be attributed to spurious CO₂ interactions with the exposed carbonyl capping groups in the Mg-MOF-74-Dzubak model, as these exposed carbonyls are not present in the periodic system or the other two cluster models. As a second distinction, a Mulliken charge analysis of the Mg-MOF-74-Dzubak cluster yields larger partial charges for the surrounding Mg atoms as compared to the Mg-MOF-74-Yu model, which may help explain the increased binding and shortened Mg-O contact in the Mg-MOF-74-Dzubak model.

There are also substantial differences in binding energies between the various cluster models. Importantly, Mg-MOF-74-small severely underbinds CO₂ compared to all other tested systems. These results for the Mg-MOF-74-small cluster indicate the inadequacy of such a small model, and likely explain the underprediction of the CO₂ adsorption isotherm from above. The Mg-MOF-74-Dzubak model shows best energetic agreement with the periodic system. Nevertheless, some of the Mg-MOF-74-Dzubak binding energy arises from truncation effects (as described above), and

the energetic agreement is thus due (at least in part) to error cancellation. Indeed, some of the binding energy in the periodic system arises from (attractive) long-range interactions, and thus we should expect to see a cluster model somewhat underpredict the binding energy. Primarily for its good agreement in binding geometries, and reasonable agreement in binding energy, we opt to use the Mg-MOF-74-Yu cluster model for the remainder of this work.

5.6.2 Final Mg-MOF-74 CO₂ Adsorption Isotherm

Using our new Mg-MOF-74-Yu cluster model, we next attempted to refit force field parameters for Mg. As discussed earlier, and because of the size of this new cluster (60 atoms), LMO-EDA-PBE0-D2 calculations became cost prohibitive in all but the smallest VDZ basis set, and thus could only be carried out for a limited set of points. Starting from the minimum energy configuration shown in Fig. 5.5, we fit Mg parameters to a 12-point scan along the Mg-O bond vector, with fit results shown in Fig. 5.6. Interestingly, though the functional form used in this fit was sufficient to accurately parameterize the interaction energies in the Mg-MOF-74-small cluster, the same force field methodology proved unsuccessful in parameterizing Mg-MOF-74-Yu-CO₂ interactions. We knew at the time that this inaccuracy was probably a consequence of uncertainties in correctly parameterizing the Mg short-range exponent. (See Chapter 3 for a full discussion of new methods for parameterizing the short-range potential.) Nevertheless, because the Slater-ISA methodology for short-range interactions had not yet been developed, we opted instead to fit the Mg interactions to a double exponential functional form, with each exponent corresponding to the ionization potential for either Mg⁺ or Mg²⁺ (the two atomic environments most likely to correctly represent the Mg cation). As shown in Fig. 5.6, this form could excellently reproduce the Mg-MOF-74-Yu model PES.

Using the double exponential functional form from above, we recomputed the adsorption isotherm of CO₂ in Mg-MOF-74. Before comparing to experiment, and as recommended by others,³³ we scaled the experimental isotherm in order to account for the pore blocking effects that are common in the M-MOF-74 series.

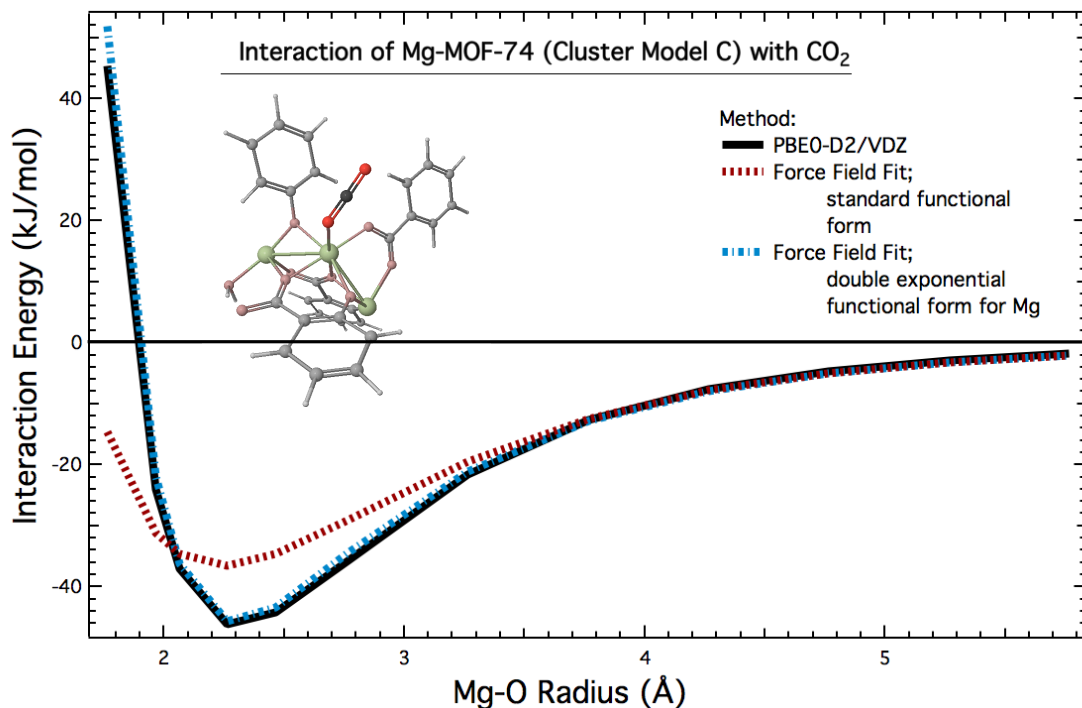


Figure 5.6: Force field fitting quality for the Mg-MOF-74-Yu cluster. A PBE0-D2 benchmark (black solid) is displayed along with two force field fits: single-exponential (red dashed) and double-exponential for Mg (blue dash-dotted). In either case, a cut of the PES is shown along the interatomic distance between the central Mg atom and the closest-contact oxygen atom in CO₂.

Using this scaled isotherm, we then obtain excellent agreement between our model potential and experiment (Fig. 5.7). Crucially, this accuracy is seen both at low- and high-pressure ranges, indicating the accuracy of the force field in modeling both the strong Mg-CO₂ binding as well as the weaker physisorption regime.

5.6.3 Transferability to Other Adsorption Isotherms

In addition to using our Mg parameters to compute the CO₂ adsorption isotherm, we also used our Mg force field in conjunction with the N₂ parameters developed by Yu et al.³⁴ to predict the N₂ adsorption isotherm. These predictions were generally

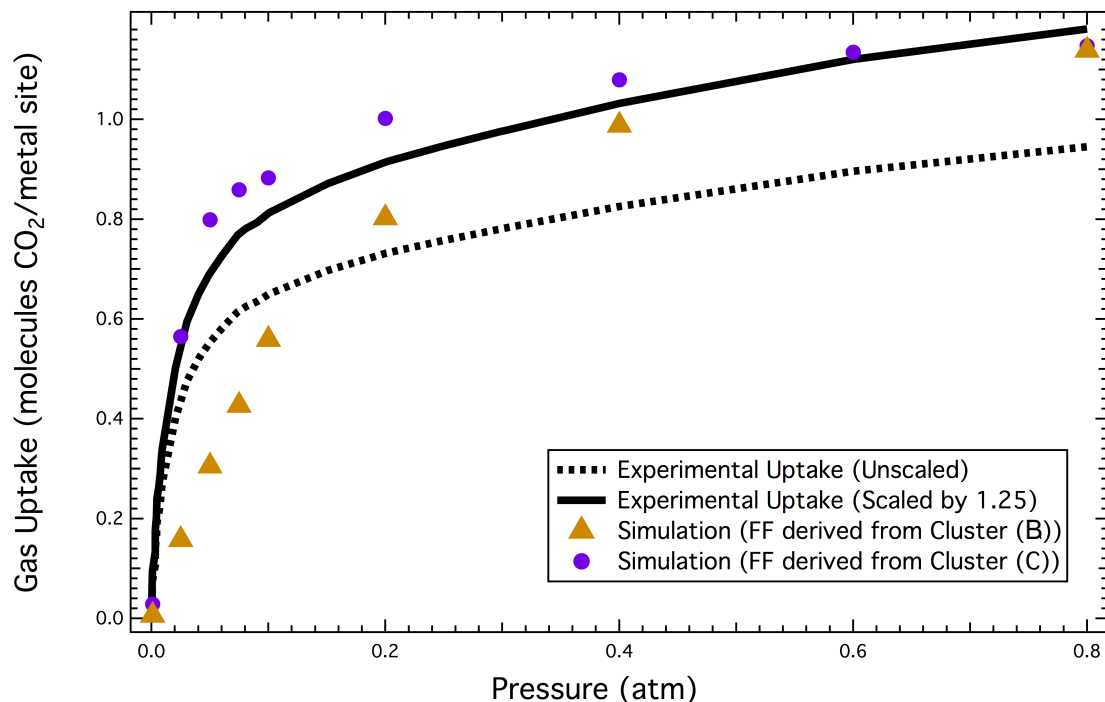


Figure 5.7: Predicted CO_2 adsorption isotherm for Mg-MOF-74. Two experimental isotherms are shown, one as directly measured by experiment (dashed line) and one scaled to account for pore block effects (solid line). Predictions from two force fields are also shown, where Mg parameters for each force field were fit either to the Mg-MOF-74-small cluster (gold triangles) or to the Mg-MOF-74-Yu cluster (purple circles). Cluster geometries are given in ??.

poor, and results are not shown. Nevertheless, the poor N_2 results suggest a lack of transferability of our Mg parameters, possibly (and as discussed in the section on Future Work) due to the unphysical double-exponential functional form used to parameterize Mg.

5.6.4 Transferability to Other M-MOF-74 systems

As a second test of transferability, we also attempted to develop force fields for other compounds in the M-MOF-74 series, starting with Co-MOF-74. Unfortunately, the open-shell nature and increased electron count of Co-MOF-74 made LMO-EDA

calculations computationally prohibitive for any reasonable basis set, and these systems were not investigated further.

5.7 Conclusions

In this unpublished work, we have determined a new methodology for fitting force fields to CUS-MOFs. While largely following our previous methodology for MOF force field development, we have shown how an LMO-EDA Energy Decomposition Analysis can be used in lieu of SAPT to generate accurate ab initio benchmark energies in cases where SAPT itself is inaccurate. Using this new methodology, we have successfully modeled CO₂ interactions in Mg-MOF-74, and have simulated the adsorption isotherm for CO₂ in Mg-MOF-74 with good accuracy compared to experiment.

Ultimately, the methods presented herein suffered from a number of practical and fundamental issues (vide infra). Until these limitations can be fully addressed, we do not anticipate that the LMO-EDA method can be broadly used to develop transferable force fields for CUS-MOFs or other large systems where SAPT is in error.

5.8 Future Work

Throughout this Chapter, we have attempted to highlight some of the key limitations of our force field development methodology for CUS-MOFs. In summary, the following issues would need to be resolved in order to expand the scope and utility of the present research:

1. **Memory Limitations in GAMESS:** As evidenced in this work, relatively large (60+ atom) cluster models are required to correctly parameterize force fields for the M-MOF-74 system. While these cluster sizes do not present difficulties for standard DFT calculations with reasonable basis sets, the corresponding LMO-EDA calculations were, as implemented in the GAMESS software pack-

age, infeasible due to memory requirements. Some time was spent attempting to address these memory issues, particularly for the memory-intensive Edmiston-Ruedenberg localization subroutine that is the source of the problem. However, due to our lack of familiarity with the GAMESS software and the LMO-EDA source code, this pursuit was eventually dropped.

2. **Fundamental Issues with LMO-EDA:** As discussed in Section 5.4, the LMO-EDA method has several theoretical limitations. In particular, and especially for functionals with no defined separation between exchange and correlation functionals, LMO-EDA does not offer a clean separation between the exchange and dispersion energies. Furthermore, and unlike some recent EDA methods,^{23,35} LMO-EDA cannot separate induction into charge transfer and polarization components.
3. **Transferability of the Force Field Functional Form:** While our final force field for studying CO₂ interactions in Mg-MOF-74 is highly accurate (both with respect to ab initio theory and with respect to experiment), it does not appear that this accuracy extends to models for the adsorption of other small molecules, such as N₂. This transferability limitation is almost certainly due to the chosen double-exponential functional form and/or the parameterization process used to obtain Mg parameters, and improvements to this methodology will be essential to make our work on the CO₂-Mg-MOF-74 system applicable to general force field development for CUS-MOFs. In particular, future work will require a better force field for describing short-range interactions, as the functional forms and parameters used in this work struggled to both accurately and transferably model the Mg-MOF-74 exchange energies.

While several of these issues (particularly practical limitations with the LMO-EDA implementation) have yet to be addressed in a meaningful fashion, several recent theoretical advances may pave the way for continued work on this project. Thus for CUS-MOFs and other systems where DFT-SAPT might be in error, we offer the following recommendations:

1. **Improved SAPT energies:** Recently, it has been proposed that the commonly used single-exchange (S^2) approximation can lead to errors in the description of the induction energy, particularly for ionic systems.^{20,36} While it is difficult to attribute errors in SAPT to a particular energy component, it may well be that SAPT poorly describes Mg-MOF-74 due to the S^2 treatment of the induction energy. In this case, eliminating the S^2 approximation might improve the DFT-SAPT total interaction energies, thus enabling SAPT to be used for (at least) closed-shell CUS-MOFs.
2. **New SAPT Correction Schemes:** As discussed in Chapter 4, deviations between SAPT and CCSD(T) can be rectified by adding a $\delta\text{CCSD(T)}$ correction to the total SAPT energy. As discussed in Chapter 4, we have empirically had good success modeling this $\delta\text{CCSD(T)}$ correction as part of the dispersion energy. Though this partitioning choice may require adjustment for treating Mg-MOF-74, the results in Chapter 4 indicate that simply correcting (rather than entirely ignoring) the DFT-SAPT energies is a promising strategy for transferable force field development.
3. **New EDA schemes:** Since this work was completed, a second-generation ALMO-EDA scheme has been implemented in the Q-Chem software package.²³ Crucially, and unlike its predecessor, this ALMO-EDA scheme now breaks up the interaction energy into electrostatic, exchange, polarization, charge-transfer, and dispersion components. While there is no guarantee that such an EDA could serve as the basis for CUS-MOF force field development (see Section 5.4), these and other recently developed EDAs may be worth investigation, and could eventually replace the (practically problematic) LMO-EDA method.
4. **Improved Force Field Functional Forms:** Since 2014, we have made significant progress in developing more accurate and transferable intermolecular force fields (see Chapters 3 and 4), and many of these advances particularly improve the description of the short-range potential itself. There is a good

chance that either the Slater-ISA FF or MASTIFF methodologies would yield high-quality force fields for Mg-MOF-74 and other systems. In this case, continued work on this project might be an exciting avenue for showcasing the MASTIFF methodology in the context of accurate inorganic/organometallic force field development.

5.A Force Field Parameters for CO₂ and Mg-MOF-74

Final force field parameters, fit using the double-exponential functional form above and the Mg-MOF-74-Yu cluster model, for CO₂ and Mg-MOF-74. These parameters should be read in as input into our group's lattice simulation code, see <http://schmidt.chem.wisc.edu/montecarlosimulationcodes> for details.

Listing 5.1: co2_mof74.pmt

```
lennard_jones_type      1      "1 for buckingham, 2 for lennard jones"

" parameters are listed as charge, A,B,C, polarizability
" units are A:kj/mol, B:A^-1, C: KJ/mol*A^6  epsilon:KJ/mol,  sigma:A,
  polarizability: A^3 "
" polarizability is defined as q^2/k, spring constant is set to .1*1.8897^3 e^2/A^3"

solute_species
atom_type_parameters  ( q, Aexch, Aelec, Aind, Adhf, Adisp, C6, C8, C10, alpha )
2
C0      0.6573800      95510.43      -27846.98      -13425.1      2065.044
        0.0      6.891E02      0.0      0.0      1.1926153
O0      -0.328690      521902.066      -163908.84      -4475.8095      -26042.04
        0.0      1.8341E03      0.0      0.0      0.9009290

solute parameters for framework cross terms  ( Aexch, Aelec, Aind, Adhf, Adisp, B, C6, C8
, C10, C12 )
C0      74376.65      24130.18      12513.37      795.36
        0.0      3.4384      1147.41867      6329.41038      29659.50100      183546.714
O0      354373.13      108208.5      2544.89      -18178.7
        0.0      3.7795      867.27598      4266.54582      28761.10636      132581.301

solute dhf cross terms (check code for input format if more than one cross term)
-6124.0
```

solute-solute exponents (Bii , Bij , Bjj)							
	3.5105206		3.6993494		3.9288490		
framework_species							
atom_type_parameters (q, Aexch, Aelec, Aind, Adhf, Adisp, B, C6, C8, C10, C12, alpha)							
9							
C1	-0.1639350	612892.611	229850.679	-6511.529	-60036.930		
	0.000	3.438	1628.820002	6821.530007	44464.989999	193602.980000	
		0.0					
H1	0.2637500	8538.651	1678.771	-612.739	-502.639		
	0.000	3.778	129.439978	679.640001	4995.299998	0.000000	
		0.0					
C2	-0.3191850	612892.611	229850.679	-6511.529	-60036.930		
	0.000	3.438	1628.820002	6821.530007	44464.989999	193602.980000	
		0.0					
C3	0.4964850	612892.611	229850.679	-6511.529	-60036.930		
	0.000	3.438	1628.820002	6821.530007	44464.989999	193602.980000	
		0.0					
O3	-1.0339500	3398.424	1965.168	-182.412	-178.025		
	0.000	2.457	2237.635879	29956.090890	561056.184030	7451461.601923	
		0.0					
C4	0.9468300	263600.161	112896.479	-11.681	-2837.170		
	0.000	3.438	772.870024	2349.180008	27539.189998	102366.260000	
		0.0					
O4	-0.8903225	656757.170	174054.351	-45410.640	-33954.271		
	0.000	3.779	1799.560008	11576.089993	50164.639999	0.000000	
		0.0					
Mg	1.5906500	917.037	2417.463	-12542.799	0.000		
	0.000	2.834	630.723467	0.000000	0.000000	0.000000	
		0.0					
Du	0.0000000	29176.333	0.000	142260.481	0.000		
	0.000	3.973	0.000000	0.000000	0.000000	0.000000	
		0.0					

5.B Simulation Parameters CO₂ Adsorption in Mg-MOF-74

Lattice simulation parameters for CO₂ adsorption in Mg-MOF-74. Of particular importance is the 'orientation_try' keyword, which is necessary to sample the specific binding geometries CO₂ adopts when binding to the open-metal site. These simulation parameters should be read in as input into our group's lattice simulation code, see <http://schmidt.chem.wisc.edu/montecarlosimulationcodes> for details.

Listing 5.2: simulation_parameters.pmt

Simulation Methodology		
energy_decomposition	yes	! yes for our force fields , no for UFF LJ , etc
solute_cross_parameter_set	yes	! this should be set to yes if using different
solute parameters		
	! for solute-solute	and solute-framework interactions as in our force fields ,
	no otherwise	
C8_10_dispersion_terms	yes	! set to yes if using C8, C10 dispersion terms
as in our force fields		
C12_dispersion	yes	
electrostatic_type	pme	! either "pme" for particle-mesh ewald , "
cutoff", or "none"		
lj_comb_rule	ZIFFF	! "opls" or "standard" for lj , "standard" or "
ZIFFF" for bkghm		
Simulation Parameters		
temperature	296.0	! temperature in Kelvin
too_close	1.8	! reject move if molecules are within
this separation in Angstroms.		
	! helpful to avoid unnecessary energy calculations and to prevent drude	
	oscillator catastrophes	
lj_bkghm	1	! 1 for bkghm force field , 2 for lj
screen_type	1	! screening for coulomb potential: 0 = no
screening , 1 = Tang-Toennies type screening		for our force fields
springconstant	0.1	! spring constant for drude oscillators (
au). set to 0.1 for our CO2/N2 force fields		
thole	2.0	! thole parameter for intra-molecular
drude oscillator screening. Set to 2.0 for		our CO2/N2 force fields.
drude_simulation	1	! set to 1 if drude-oscillators are being
used, 0 otherwise		
pme_grid	100	! size of the pme grid
alpha_sqrt	0.6	! alpha sqrt for the electrostatic

interactions		
lj_asqrt	0.6	! alpha sqrt for the pme dispersion
lj_cutoff	7.5	! cutoff for long range LJ or C6,C8,C10
dispersion interactions		
ewald_cutoff	5.0	! cutoff for real space pme
cav_grid_a	30	
cav_grid_b	30	
cav_grid_c	30	
na_nslist	30	! neighbour list searching grid
nb_nslist	30	! neighbour list searching grid
nc_nslist	30	! neighbour list searching grid
orientation_try	2000	! max number of orientation samplings
REL_THRSH	0.05	! sampling threshold
ABS_THRSH	3.0	
BZ_CUTOFF	100.0	

6 BENCHMARK DATABASE FOR AB INITIO FORCE FIELD DEVELOPMENT

Part IV

Practical Matters

7 WORKFLOW FOR INTERMOLECULAR FORCE FIELD

DEVELOPMENT

Due in part to the improvements in Chapters 3 and 4, the development protocol for Symmetry-Adapted Perturbation Theory (SAPT)-based, *ab initio* force fields is now fairly robust with respect to many parameterization details. Consequently, much of the workflow is now automated and requires little user input. The following sections are designed to give future users familiarity this workflow, not only as a “blackbox” tool, but also as a starting point for more complex and/or system-specific force field development. To this end, we first provide an overview of the workflow itself, and then describe the theoretical and practical details of each step in subsequent sections.

In order to gain expertise in practical force field development, new force field developers are encouraged to read through (in order) Chapters 2, 7 and 8 to obtain a conceptual understanding of the force field development process, after which they should work on developing their own force field using the semi-automated workflow (Chapter 7) and the **Parameter Optimizer for *Inter*-molecular Force Fields (POInter)** software (Chapter 8). Developing a force field for water makes for an excellent teaching example, however any interesting (and preferably small!) molecule will suffice.

7.1 Overview

As discussed in Chapter 2, our SAPT-based force field methodology principally relies on modeling two-body interactions for a given system of interest. These two-body (i.e. dimer) interactions are completely defined by the positions and relative orientation of the two constituent monomers, and in practice we parameterize the two-body model based on benchmark SAPT energies for a series of gas-phase

dimer configurations.* We are usually interested in obtaining transferable parameters for a new molecule or atomtype, in which case it is often easiest to model the interactions between two identical monomers (a so-called homo-monomeric dimer interaction).[†] Still, there are reasons why it can be advantageous to instead study hetero-monomeric dimer interactions, and the workflow described herein applies equally to studying both homo-monomeric and hetero-monomeric dimer interactions.

Regardless of the chosen dimer of study, modeling a two-body potential energy surface (PES) involves two major steps. First, we must obtain benchmark two-body energies for a series of well-chosen dimer configurations. Second, we must calculate and/or fit all force field parameters so as to completely develop a force field for the two-body interaction energies. For the SAPT-based force fields described in Chapters 3 and 4, these two overarching steps lead to the following workflow:

* At first, it may seem counter-intuitive to focus so heavily on modeling the energetics of gas-phase dimers. After all, aren't we interested in simulating a wider variety of chemically-relevant systems, including homogeneous and heterogeneous liquids, solids, and super-critical phases? This apparent discrepancy can be resolved by looking at the many-body expansion (MBE) described in Chapter 2. From this expansion, we see that *any* system can be modeled as a sum of two- and many-body interactions, with the two-body interactions plus N-body polarization (an energy term which we obtain automatically in Section 7.5.5) accounting for upwards of 90 – 95% of the total N-body energy.³⁷ Consequently, and regardless of whether we are ultimately interested in studying a homogeneous liquid or a heterogeneous supercritical phase, for ab initio force field development it's critical that we develop and parameterize accurate models for all two-body interactions. Thus in practice, our focus is often on developing new and improved force fields for gas-phase dimer interactions, always with the goal of using the MBE to run simulations on any N-body system of interest.

[†]In general, force field development based on homo-monomeric interactions involves the fewest atomtypes (and thus the fewest number of free parameters!), and is to be preferred. On the other hand, hetero-monomeric-based force field development can yield the best accuracy for studying specific systems where either transferability is difficult (see Chapter 5 for an example) or where computational expense is an issue. (Running large-basis-set SAPT calculations on the naphthalene dimer, as an example, is currently infeasible, whereas benchmark calculations on naphthalene-Ar interactions are affordable.)

I) Generate benchmark two-body energies

- 1) Generate a series of well-chosen dimer configurations (see Section 7.2)
- 2) Calculate DFT-SAPT benchmark energies for all dimer configurations from the previous step (see Section 7.3)
- 3) Optionally (depending on system size and the accuracy of DFT-SAPT for the chosen system), calculate CCSD(T) or CCSD(T)-f12 benchmark energies in order to correct the DFT-SAPT energies above (see Section 7.4)

II) Parameterize the two-body PES

- 1) For each unique monomer, obtain the following monomer-specific parameters:
 - i. Multipole moments, Q (see Section 7.5.2)
 - ii. ISA Exponents, B (see Section 7.5.3)
 - iii. Dispersion Coefficients, C_n (see Section 7.5.4)
 - iv. Induced Dipole Polarizabilities, α (see Section 7.5.5)
- 2) Obtain all remaining force field parameters by fitting a chosen force field functional form to the two-body benchmark energies from Step I) (see Chapter 8)

The entire force field development process has been made reasonably ‘black-box’, and can be carried out via a handful of input files and easy-to-use run scripts. This semi-automated workflow for SAPT-based force field development is available for download at <https://github.com/mvanvleet/workflow-for-force-fields>, and should be sufficient for most routine force field development. Installation and usage instructions are included on the website, and are also reprinted in Fig. A.1 for convenience. The remainder of this Chapter is designed to give new users a sense of the theory and practice involved in using the workflow, and we now turn to an in-depth discussion of each step.

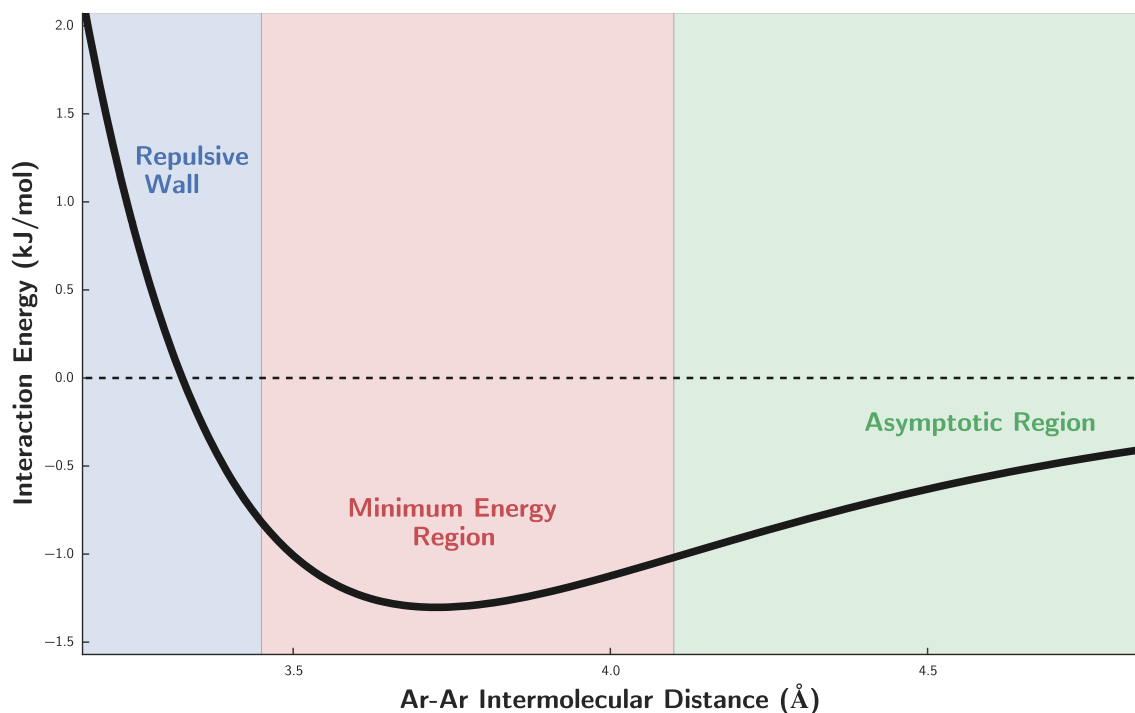


Figure 7.1: Generalized form of a PES showing the repulsive wall, minimum energy, and asymptotic regions of the argon dimer. Cutoffs between the different regions should be taken qualitatively.

7.2 Geometry Generation

7.2.1 Guiding Principles

For any given monomer(s) of interest, the first step in the force field development process is to choose a series of optimal dimer configurations. This ‘optimal’ set is highly dependent on the type of force field that is being fit, and indeed the recommendations offered below are specific to the SAPT-based force fields described in Chapters 3 and 4.

In general, and as shown in Fig. 7.1, a given PES will have (qualitatively) three different regions: a repulsive wall, a minimum energy region, and an asymptotic region. (Energies in the asymptotic region are usually attractive, but are sometimes

repulsive due to unfavorable electrostatic interactions). Based on the principles of statistical mechanics and the Boltzmann distribution, we know that (for a system at constant temperature T) the probability P_i of observing a system in state i is exponentially-dependent on the energy of that state:³⁸

$$P_i \propto \exp(-E_i/k_B T) \quad (7.1)$$

where k_B is the Boltzmann constant.

Due to the exponential relationship between the energetic stability of a given state and the probability of experimentally observing said state, *routine molecular simulation will most frequently sample dimer configurations near the minimum energy and asymptotic regions of the potential*.^{*} Consequently, these two portions of the PES are the most important to accurately model with a force field. Nevertheless, and as discussed in Section 7.5, the asymptotic region of the PES primarily depends on functional forms whose parameters are calculated from monomer properties, making the dimer-based fits described in Chapter 8 relatively insensitive to inclusion of configurations in this region.[†] By contrast, the force field parameters we directly fit to the dimer PES are primarily sensitive to the shape and location of the repulsive wall and (to a lesser extent) the minimum energy regions. Consequently, and based on the combination of their observation probability in molecular simulation as well as their importance in the improving the force field fitting process, *dimer configurations along the repulsive wall and (even more importantly) in the minimum energy region*

^{*} Although it's straightforward to see why the minimum energy region gets sampled in molecular simulation, the importance of the asymptotic region may be hard to understand simply by looking at Eq. (7.1) and the 2-body PES shown in Fig. 7.1. We should recognize, however, that the two-body energy of an N -body system is determined both by nearest neighbor interactions (whose configurations are typically in the minimum energy region of the 2-body PES) and by the more distant, non-nearest neighbor interactions (whose configurations lie in the asymptotic region). The number of non-nearest neighbor interactions outweigh the closer-contact interactions, which in turn makes *both* the minimum energy and asymptotic regions of the potential important to correctly model.

[†]By contrast, other functional forms (e.g. Lennard-Jones) *do* have parameters that effect the asymptotic region, and for these force fields it would be important to include this region in the parameterization process.

*are the most important to parameterize in order to develop highly-accurate force fields.**

In practice, a standard procedure for optimally sampling across the PES has been established for the Slater-ISA FF and MASTIFF force fields. Though use of different functional forms might require a different relative sampling of the dimer PES, the next sections completely outline the theory and practical calculations that are involved in generating dimer configurations for the development of Slater-ISA FF and MASTIFF force fields.

7.2.2 Theory

Assuming rigid monomer geometries, a dimer configuration can be completely determined (without loss of generalization) by fixing the center of the first monomer at the origin and by placing the second monomer according to six variables. r , θ , and ϕ determine the position of the center of the second monomer, and the three-dimensional variable Ω determines the relative orientation of this second monomer about its center. In practice, Ω is most easily described by a quaternion, and the interested reader is referred to Ref. 39 for details.

For both the Slater-ISA FF and MASTIFF fits, dimer configurations are sampled psuedo-randomly using Shoemake’s algorithm,⁴⁰ which ensures even sampling of the dimer configurations. Additionally, and in order to achieve a proper balance of sampling between the repulsive wall and minimum energy regions, the following dimer configurations are excluded from sampling:

1. Configurations with any atom-atom contact distance $r_{ij} \leq 0.8 \times (r_i^{\text{vdw}} + r_j^{\text{vdw}})$, where r_{ij} is the contact distance and r^{vdw} is the tabulated van der Waals radius for a given element
2. Configurations with all atom-atom contact distance $r_{ij} \geq 1.3 \times (r_i^{\text{vdw}} + r_j^{\text{vdw}})$

*Historical note: For force field functional forms which poorly model the repulsive wall (e.g. Lennard-Jones force fields or the Born-Mayer-IP FF described in Chapter 3), the force field fit quality strongly depends on the relative representation of repulsive and attractive dimer configurations, and including either too few or too many repulsive configurations can be problematic. Only with the advent of Slater-ISA FF and MASTIFF is the fit quality strictly improved by including repulsive configurations.

A working code for this sampling algorithm has been developed and is described below.

7.2.3 Practicals

In practice, generation of the dimer configurations is fairly straightforward. The three required input files – `input/dimer_info.dat`, `input/generate_grid_settings.inp`, and `input/<mona>_<monb>.inp` – are listed in Listings 7.1 to 7.3 and using the pyridine dimer as an example. (Here and throughout, we use angle brackets to indicate required arguments.) Each input file may need to be modified for the specific dimer under consideration, and comments within these input files explain any necessary system-specific changes.

Once all required input files have been created/modified, the geometry generation process can be carried out very simply from the main workflow directory by executing the command

```
./scripts/make_geometries.sh
```

7.3 SAPT Benchmarks

After geometry generation, the next step in the Workflow is to run benchmark DFT-SAPT calculations on all dimer configurations. For a detailed analysis of SAPT, and DFT-SAPT in particular, the reader is referred to Refs. 29, 41, 42. DFT-SAPT calculations can be performed in a fairly black-box manner using the Molpro software, though the following points are worth note:

1. For best accuracy, and as described in Ref. 34, monomer Density Functional Theory (DFT) calculations need to be asymptotically-corrected (AC) in order to achieve best accuracy. This asymptotic correction is computed as the difference between the HOMO and the vertical ionization potential for each monomer, and can be calculated automatically by running the command

```
./scripts/submit_ip_calcs.py
```

(The calculation takes only a few minutes for small molecules, but may take longer for larger systems.) Importantly, the HOMO calculation should be computed using the same basis set as the DFT-SAPT calculations themselves.

2. Accurate SAPT dispersion energies generally require use of midbond functions, as described in Ref. 34. Locations for the midbond functions can be specified in the `dimer_info.dat` file. For most small molecules (such as those described in Chapter 3), it is often sufficient to place a single midbond at the midpoint between each monomer's center of mass. For larger molecules, additional midbonds (especially ones near close-contact interaction sites) may be required.
3. The included workflow assumes an aVTZ+m basis set (where the +m represents the midbond functions). This is generally of sufficient accuracy for most systems, though an aVQZ+m basis set should be used when possible to ensure convergence of the DFT-SAPT dispersion energies.

Once the appropriate midbond functions have been added to the `dimer_info.dat` input file, and the AC calculations have finished, the DFT-SAPT input files can be generated by executing the command

```
./scripts/make_sapt_ifiles.py
```

The resulting input files can then be run using the Molpro software, either in serial or in parallel. *Care should be taken to ensure that multiple calculations do not end up on the same compute node, as this can often result in i/o caching issues and reduced computational efficiency.*

7.4 CCSD(T) Calculations

When affordable, CCSD(T) calculations should be run on (at least a subset of) the dimer configurations, both in order to benchmark the DFT-SAPT energies and to provide a δ CCSD(T) correction for later fitting of the SAPT potential. Recently, an explicitly-correlated CCSD(T)-f12 method has been proposed, which for practical

purposes is identical to CCSD(T) but with faster basis set convergence.⁴³ Usually CCSD(T)-f12a/aVTZ+m is an excellent approximation of the CCSD(T)/CBS limit. The input files for CCSD(T)-f12/aVTZ+m calculations can be set up by executing the command

```
./scripts/make_ccsdt_ifiles.py
```

and by running each input file using the Molpro software package.

7.5 Monomer-Based Parameterization

While/after running the DFT-SAPT calculations, the next step in the Workflow is to compute various force field parameters which only depend on the identities of the individual monomers themselves. The following subsections describe the calculations of multipole moments (Section 7.5.2), short-range exponents (Section 7.5.3), dispersion coefficients (Section 7.5.4), and induced dipole polarizabilities (Section 7.5.5). First, however, we outline the scope and useful features of the CamCASP software used to perform these monomer property calculations.

7.5.1 Distributed Property Calculations using CamCASP

CamCASP is a collection of scripts and programs useful for (among other things) the calculation of distributed multipoles and polarizabilities.⁴⁴ Of particular importance is the choice of distribution method, as this determines how the various molecular properties of interest should be mapped onto corresponding atom-in-molecule properties. Currently, two main distribution (or ‘charge partitioning’) schemes are available in CamCASP: Distributed Multipole Analysis (DMA)⁴⁵ and Iterated Stockholder Atoms (ISA).⁴⁶ The theory behind the ISA procedure has already been detailed in Chapter 2, and monomer property calculations using DMA are described in Ref. 29, 45, 47. In general, and where available, ISA-based properties are to be preferred, and we recommend an ISA-based parameterization scheme for obtaining multipoles and atom-in-molecule exponents. A DMA-based method is currently required for obtaining dispersion coefficients and static polarizabilities, though

ISA-based strategies for these properties are under active development and (in the case of dispersion) are discussed in Section 7.5.4. A complete overview of available property calculations and distribution schemes, along with relevant references, is given in Table 7.1.

Property	Parameterization Scheme	
	ISA	DMA
Multipoles	Section 7.5.2 Ref. 46	– Ref. 29, 45
Exponents	Section 7.5.3 Ref. 48	–
Dispersion Coefficients	Section 7.5.4 –	Section 7.5.4 Ref. 28
Dipole Polarizabilities	–	Section 7.5.5 Ref. 28

Table 7.1: Overview of ISA- and DMA-based methods for obtaining distributed monomer properties. Details for each monomer parameterization are given in the listed section and/or reference.

7.5.2 Multipoles

Practicals

ISA-based multipoles are described in detail in Ref. 46, and can be calculated using the CamCASP software. To set-up the ISA calculations, execute the command

```
./scripts/make_isa_files.py
```

which creates the necessary ISA files for calculating both distributed multipoles and exponents (see Section 7.5.3). After running these calculations (a process which may require several hours, depending on the molecule), the multipoles can be extracted simply by running

```
./scripts/workup_isa_charges.py
```

This work-up script produces several output files,

```
<monomer>_ISA_L4.mom
```

```
<monomer>_ISA_L2.mom
```

```
<monomer>_ISA_L0.mom
```

which correspond to multipole moments for various long-range electrostatic models. Using Stone's notation,² the Lx suffix refers to the highest order of multipole moments (L0 = point charges, L1 = dipoles, L2 = quadrupoles, etc.) included in the model. The L4 model is output by the CamCASP software package, and the L2 and L0 models are generated by rank-truncation (that is, zeroing out) of the higher-order multipole moments. For most routine force field development, the L2 model is to be preferred for its balance of accuracy and computational expense. Next, however, we discuss situations in which different electrostatic models may be desirable.

Advanced Multipole Parameterization Options

As stated above, for the purposes of obtaining sub- kJ/mol accuracy force fields it is often important to model the long-range electrostatics using ISA-based multipoles truncated to no farther than quadrupolar (i.e. 'rank 2' or L2)² contributions. Due to computational and/or software limitations, however, there exist practical cases where it becomes advantageous to exclude all higher-order multipole moments.⁴⁹ In such cases, two different types of long-range electrostatic models are useful. First, for reasonably isotropic molecules a good option is to rank-truncate the ISA multipoles to the L0 point charge contributions, thus yielding a so-called 'atom-centered point charge model'. On the other hand, for more anisotropic functional groups such as those described in Ref. 50, an atom-centered point charge model can be insufficiently accurate, making it necessary to model the long-range electrostatics by including additional 'off-center/off-site' point charges. Given a well-chosen set of off-site charges, an off-center point charge model usually can reasonably

reproduce the effects of the neglected higher-order multipole moments.⁵¹ In the past, locations for the off-center charges have usually been manually tuned or optimized in a system-specific manner, though recent work suggests the possibility of switching to non-empirical methods in order to more easily calculate/optimize positions for the extra-atom sites.^{52,53}

For atom-centered point charge models, the output of the `workup_isa_charges.py` script automatically provides the required rank-truncated multipole file (listed as `<monomer>_ISA_L0.mom` in the `isa/` sub-directory). Note that, because the `<monomer>_ISA_L0.mom` file is given as a simple rank-truncation of the more complete `<monomer>_ISA_L2.mom` multipoles, the L0 moments (that is, point charges) are identical between the two files.

For developing rank-transformed point charge models, Ferenczy et al. has programmed a method for calculating electrostatic potential-fitted charges, which can be thought of as a ‘rank transformation’ procedure. The author’s MULFIT program can be downloaded online at <http://www-stone.ch.cam.ac.uk/pub/gdma/index.php>, and documentation for the program is found in the `documentation/` sub-directory of the Workflow. Assuming the `mulfit` executable is in your `$PATH`, a basic rank transformation can be performed using the following steps:

```
cp templates/mulfit.inp isa/<monomer>/OUT/
cd isa/<monomer>/OUT/
mulfit < mulfit.inp
```

Here the default `mulfit.inp` file is set to take in the L4 rank multipoles and rank-transform them to an L0 model. In this case, note that the L0 moments between the rank-transformed and rank-truncated moments will *not* be identical, and testing is required to ascertain which moments yield optimal force field parameters.

The MULFIT program can additionally be used to develop off-site point charge models. In this case, the input multipole file (default `ISA_L4.mom`) should be edited to include the additional sites, and an example of the required syntax is given in `documentation/examples/ISA_L4_offsites.mom` for a 4-site water model. Importantly, the MULFIT program does not help optimize the position(s) of the off-site charge(s), and thus the task of choosing the number and position(s) of the off-site(s)

is left to the user.

After fitting multipole parameters with the MULFIT program, the program output gives two indications of fit quality. First, the agreement between the total reference and fitted multipoles moments is listed, and this should be taken as a primary indication of multipole quality. Second, the program gives a ‘Goodness of fit’ parameter, expressed as an energy. While difficult to interpret in an absolute sense, in comparing different rank-transformed models we have generally found that models with lower ‘Goodness of fit’ parameters yield better force field fits.

7.5.3 ISA Exponents

As described in Chapters 2 and 3, the ISA procedure produces a set of distributed atom-in-molecule (AIM) electron densities. The orientational average of each of these AIM densities, or ‘shape-functions’, are spherically-symmetric quantities that describe the radial decay of the AIM density.⁴⁶ As described in Chapter 3, and using the algorithm detailed in Section 7.B, the shape-functions can be fit to a Slater-type function in order to yield an isotropic, exponentially-decaying model for the ISA densities. Importantly, the Slater-exponents in this density model directly yield the exponents necessary to describe short-range effects (such as exchange-repulsion and charge penetration) in the two-body force field (see Chapter 3 for details).

Assuming the ISA calculations have already been run to obtain multipole moments (see previous section), the ISA exponents can be obtained very simply by running the command

```
./scripts/workup_isa_exponents.py
```

The resulting exponents are given in the file `isa/<monomer>.exp`, which uses a file format recognized by the POInter pre-processing scripts (see Chapter 8).

7.5.4 Dispersion Coefficients

Theory

Dispersion coefficients can also be determined from distributed molecular (that is, AIM) property calculations, using either an ISA- or DMA-based approach. The method for obtaining distributed dispersion coefficients has been described in detail elsewhere for an assortment of DMA-based approaches,^{2,17,28,29,55,56} and Ref. 29 in particular provides a useful summary of the different equations and molecular properties that are needed to derive the types of dispersion models used in Chapters 3 and 4. In brief, AIM dispersion energies can be obtained by integrating over distributed frequency-dependent polarizabilities for each monomer, and the interested reader is referred to Chapter 9 of Ref. 2 for complete details. Under the simplifying assumption that we can treat these frequency-dependent polarizabilities as isotropic, the dispersion energy expression is given by

$$E_{\text{disp}}^{ab} \approx -\frac{C_6^{ab}}{r_{ab}^6} - \frac{C_8^{ab}}{r_{ab}^8} - \dots \quad (7.2)$$

for each atom pair, where

$$C_6^{ab} = \frac{3}{\pi} \int_0^\infty \bar{\alpha}_1^a(i\omega) \bar{\alpha}_1^b(i\omega) d\omega, \quad (7.3)$$

$$C_8^{ab} = \frac{15}{2\pi} \int_0^\infty \bar{\alpha}_1^a(i\omega) \bar{\alpha}_2^b(i\omega) + \bar{\alpha}_2^a(i\omega) \bar{\alpha}_1^b(i\omega) d\omega, \quad (7.4)$$

and higher order terms are defined analogously. Here C_n^{ab} are the atom-atom dispersion coefficients, and $\bar{\alpha}_l^a$ are the rank l , isotropic, AIM frequency-dependent polarizabilities. The formalisms involved in evaluating Eqs. (7.3) and (7.4) can be somewhat involved, but for our purposes the important take-away is the understanding that the dispersion coefficients can be entirely determined by calculating the frequency-dependent polarizabilities for each atom in its molecular environ-

ment.

Although it is straightforward to calculate *molecular* frequency-dependent polarizabilities, a central difficulty in obtaining transferable dispersion coefficients is that, in order to evaluate Eqs. (7.3) and (7.4), we must have some physically-meaningful method for calculating *atom-in-molecule* polarizabilities. Many distribution strategies exist in the literature, and here we focus on two such techniques. First, and as we have used in Chapters 3 and 4, one can utilize a DMA-based approach to partition the polarizabilities into AIM contributions. In this case, and due to deficiencies in the DMA partitioning scheme, the resulting atomic polarizabilities are not always positive-definite and monotonically-decaying, and this unphysical behavior can lead to a breakdown in transferable parameterization.⁵⁵ To correct for this undesirable behavior, McDaniel and Schmidt have proposed a constrained fitting procedure whereby atomic polarizabilities can be optimized in an iterative fashion, thereby generating transferable atomic polarizabilities at the expense of requiring a fairly large training set for each unparameterized atomtype (see Section 7.5.4 for details).

As an alternative to the above iterative polarization partitioning scheme, recently Misquitta has developed an ISA-based partitioning scheme to extract the atomic frequency-dependent polarizabilities. While this approach requires further testing, and is not yet published, the resulting ‘ISA-pol’ method appears to lead to a more physically-meaningful partitioning of the molecular polarizabilities. For practical purposes, this more physical partitioning enables us to determine transferable dispersion coefficients without resorting to large training sets. Formalisms and technical details related to ISA-pol are the subject of Section 7.5.4, and a comparison between the two methods for obtaining dispersion coefficients is given in Section 7.5.4. Finally, each method for obtaining dispersion coefficients requires a small amount of post-processing, and this is discussed in Section 7.5.4.

Iterative-DMA-pol

Theory As described in Ref. 29, the iterative-DMA-pol (iDMA-pol) method of McDaniel and Schmidt performs a constrained optimization of atomtype-specific frequency dependent polarizabilities by fitting all polarizabilities to reproduce the so-called ‘point-to-point response’, α_{PQ} . This point-to-point response is a molecular quantity that describes the change in electrostatic potential at point P due to an induced change in the electron density of a molecule caused by a point charge perturbation q_Q at point Q. For an isotropic polarizability model,

$$\alpha_{PQ} = -q_Q \sum_{a,lm} T_{0,lm}^{Pa} \bar{\alpha}_l^a T_{lm,0}^{aQ} \quad (7.5)$$

where the T are the spherical harmonic interaction functions described above and in Ref. 2. Aside from the isotropic polarizabilities $\bar{\alpha}_l^a$, all quantities in Eq. (7.5) are directly calculated in CamCASP, enabling us to fit the isotropic polarizabilities on the basis of CamCASP property calculations (see Appendix A of Ref. 29 for details).

Practicals Using the iDMA-pol method in the Workflow has two software dependencies:

1. The iDMA-pol fitting program itself, which can be downloaded at <https://github.com/mvanvleet/p2p-fitting>. Three executables (`main_dispersion`, `main_drude`, and `localize.sh`) need to be added to your bash `$PATH` for the scripts listed in this section to work properly.
2. CamCASP, which can be downloaded from <http://www-stone.ch.cam.ac.uk/programs/camcasp.html>. CamCASP also requires several environment variables to be added to your bash `$PATH`, and some of these environment variables are also used by the iDMA-pol fitting program.

and requires two additional input files:

1. `input/<monomer>.atomtypes`: The iDMA-pol fitting program performs a constrained optimization whereby the $\bar{\alpha}_i^a$ are set to be identical for atoms with the same atomtype. Consequently, the `<monomer>.atomtypes` input file is required to specify the atomtypes in each monomer. This `.atomtypes` file has the same format as an `.xyz` file, with the exception that the element names for each atom are replaced with a user-defined atomtype. See Listing 7.4 for an example with pyridine.
2. `templates/dispersion_base_constraints.index`: As described below, with iDMA-pol it is usually advisable to only fit one or two atomtype polarizabilities at a time, with the remaining atomtype polarizabilities read in as hard constraints. The `dispersion_base_constraints.index` file lists these hard constraints in a block format,

```
CT
1
  7.14483224  7.11095841  6.87452508  6.19718464  4.87589777
  3.17818610  1.56461102  0.51670933  0.09175313  0.00367230
2
  20.26394042  20.00584110  17.66562710  14.33668329  12.03179893
  11.49156262  7.86254302  3.10936998  0.53746459  0.01774391
3
  77.37303638  73.13014787  24.68682297  -13.48390193  0.40172836
  29.76747226  34.31668916  17.88515654  3.13260459  0.10137127
```

which lists each constrained atomtype along with 10 frequency-dependent polarizabilities for each polarizability rank (1-3). (CamCASP uses numerical integration to solve Eq. (7.3), and the 10 polarizabilities per rank correspond to the frequencies CamCASP needs to perform the numerical quadrature. See the CamCASP user manual for details.) Each polarizability block should be separated by a blank line, and the atomtypes listed in the `.index` file *must* match those in the `.atomtypes` file for any hard constraints to be successfully applied. Previously-fit atomtype polarizabilities from Ref. 28 are already included in `dispersion_base_constraints.index` so as to minimize the number of hard constraints that the user will need to add manually, and these hard constraints should be used whenever possible.

Once all required input files have been created, and assuming the IP calculations from Section 7.3 have already been performed, the CamCASP calculations necessary to run the iDMA-pol program can be performed by executing the command

```
./scripts/make_dmapol_files.py
```

and running the resulting input files through the CamCASP software (a process which can take several hours). Once the CamCASP calculations finish, dispersion coefficients can be obtained by running the following work-up script:

```
./scripts/workup_dispersion_files.sh
```

The resulting dispersion coefficients will be listed in the `dispersion/<monomer>.cncoeffs` output file.

When generating dispersion coefficients using iDMA-pol, the following sanity-checks should always be performed:

1. The `<monomer>_fit_dispersion.out` file lists the number and names of unconstrained atomtypes. Ensure that the number and type of unconstrained atomtypes match your expectations, and that the number of fit atomtypes is kept relatively small (1-2 max). If you need to fit multiple atomtypes simultaneously, or you obtain unphysical dispersion coefficients (see next point), you'll likely need to utilize the iterative fitting algorithm outlined in Ref. 28 or obtain dispersion coefficients from an ISA-based scheme (Section 7.5.4).*
2. Dispersion coefficients should always be positive. Any negative dispersion coefficients are likely a sign of unphysical atomic polarizabilities (see next point).
3. Physically-speaking, the atomic polarizabilities at each rank should be positive definite, and monotonically-decreasing.^{2,55} Unphysical behavior (especially at rank 3) is sometimes unavoidable, but often indicates poor fit quality and can lead to inaccurate and/or non-transferable dispersion coefficients. Always check the output `.casimir` files for the physicality (positive-definiteness and

*Scripts to perform the iterative iDMA-pol fitting algorithm can be made available upon request.

monatomic-decrease) of the frequency-dependent polarizabilities for each atomtype and each rank.

Finally, given a set of physical atomic polarizabilities and dispersion coefficients, dispersion coefficients from the iDMA-pol method can be worked-up using the post-processing scripts described in Section 7.5.4.

ISA-pol

Theory Rather than iteratively fitting polarizabilities to reproduce the point-to-point response, with ISA it is possible to compute the atomic polarizabilities directly. First, note that the frequency-dependent, molecular polarizabilities are given by the following formula:

$$\alpha_{lm,l'm'}(\omega) = \int \int \hat{Q}_{lm}(\mathbf{r}) \alpha(\mathbf{r}, \mathbf{r}' | \omega) \hat{Q}_{l'm'}(\mathbf{r}') d\mathbf{r} d\mathbf{r}' \quad (7.6)$$

Here \hat{Q} are the regular spherical harmonic operators (defined in Appendix A of Ref. 2) of rank l and order m , and $\alpha(\mathbf{r}, \mathbf{r}' | \omega)$ is the frequency-dependent density susceptibility (FDDS), or charge density susceptibility, which measures the change in charge density at \mathbf{r}' that results from a delta-function change in the electric potential at point \mathbf{r} . From ??, we have that

$$1 = \sum_a \left(\frac{\bar{w}^a(\mathbf{r})}{\sum_m \bar{w}^m(\mathbf{r})} \right) = \sum_a \bar{p}_a(\mathbf{r}), \quad (7.7)$$

where the bars indicate that we have normalized the atom-in-molecule densities and weight functions. Substituting this equation into Eq. (7.6), we arrive at an

ISA-based definition of the AIM polarizabilities:

$$\begin{aligned}
 \alpha_{lm,l'm'}(\omega) &= \iint \hat{Q}_{lm}(\mathbf{r}) \alpha(\mathbf{r}, \mathbf{r}' | \omega) \hat{Q}_{l'm'}(\mathbf{r}') d\mathbf{r} d\mathbf{r}' \\
 &= \sum_a \sum_b \iint \hat{Q}_{lm}(\mathbf{r}) p_a(\mathbf{r}) \alpha(\mathbf{r}, \mathbf{r}' | \omega) p_b(\mathbf{r}') \hat{Q}_{l'm'}(\mathbf{r}') d\mathbf{r} d\mathbf{r}' \quad (7.8) \\
 &\equiv \sum_a \sum_b \alpha_{lm,l'm'}^{ab}(\omega)
 \end{aligned}$$

While this formula bears similarity to DMA-based polarization approaches,^{55,56} the advantage of Eq. (7.8) is that the AIM polarizabilities are defined in a physically-meaningful and transferable manner. Consequently, with little refinement these ISA-based polarizabilities (ISA-pol) can be used to directly obtain transferable dispersion coefficients for individual atom-in-molecule, all without recourse to the iterative fitting process required in Section 7.5.4.

Practicals The ISA-pol method has been completely implemented as of CamCASP-6.0, though the input scripts are (as of this writing) still in beta. Consult the CamCASP user manual or contact Alston Misquitta for up-to-date details and required input files.

Comparison between iDMA-pol and ISA-pol

Preliminary results for the ISA-pol method, tested on the 91 dimer test set from Chapter 3, appear to be of similar accuracy compared to the iDMA-pol method, though both methods appear to have their own strengths and weaknesses when it comes to obtaining dispersion coefficients for different atomtypes. A comparison between the two different methods is given in Table 7.2. Overall, ISA-pol appears to give more physically-meaningful atomic polarizabilities, whereas an isotropic iDMA-pol description is (for anisotropic systems) sometimes a better ‘effectively anisotropic’ model.*

* A main difference between the iDMA-pol and ISA-pol coefficients is that iDMA-pol fits more strongly to the point-to-point (p2p) response function, whereas ISA-pol coefficients are set to the

iDMA-pol	ISA-pol
Ease of Parameterization	
<ul style="list-style-type: none"> • For systems with a single (or possibly two) unparameterized atom-type(s), straightforward to parameterize new atomtypes • For systems requiring dispersion coefficients for several unparameterized atomtypes, requires a library of systems containing these atomtypes, and an iterative procedure to fit the new atomtypes 	<ul style="list-style-type: none"> • Straightforward for all molecules, regardless of number of unparameterized atomtypes
Physicality of the Distributed Polarizabilities	
<ul style="list-style-type: none"> • Polarizabilities tend to be positive-definite and monotonically-decaying at low rank, but not always for rank 3 • Physicality is highly-dependent on the quality of previously parameterized atomtypes 	<ul style="list-style-type: none"> • With few exceptions, polarizabilities are positive-definite and monotonically-decaying at all ranks
Accuracy of the Dispersion Coefficients	
<ul style="list-style-type: none"> • Good to excellent accuracy for atomtypes which have been fit to reproduce large library of molecular systems • Fair accuracy for certain atomtypes (such as chlorine or bromine) not parameterized to an extensive library • For anisotropic systems (such as CO₂), tends to give a better isotropic description than ISA-pol – we hypothesize that this is a result of directly fitting the point-to-point response, leading to an ‘effectively-anisotropic’ model 	<ul style="list-style-type: none"> • Good to very good accuracy for all tested systems, regardless of what atomtypes are represented • Isotropic dispersion coefficients tend to give worse accuracy for anisotropic systems compared to iDMA-pol, whereas anisotropic dispersion models (see Chapter 4 based on ISA-pol are of similar accuracy to the iDMA-pol method

Table 7.2: Comparison between the iDMA-pol and ISA-pol methods.

Dispersion Coefficient Post-processing

Regardless of which distribution method is used, some post-processing is needed to transform the ISA-pol/iDMA-pol coefficients into optimal dispersion force field parameters. In particular, while the DFT-SAPT energies from Molpro and CamCASP should agree, in practice the different software packages use different kernels (ALDA+LHF and ALDA+CHF, respectively) to calculate the linear response functions. Consequently, this means that the dispersion coefficients calculated in CamCASP are intended to reproduce the CamCASP-calculated DFT-SAPT dispersion energies, but may only be approximately accurate for Molpro-calculated DFT-SAPT dispersion energies.* In practice, the CamCASP-calculated dispersion coefficients slightly underestimate the Molpro dispersion energies, and the coefficients need to be scaled (usually by a factor of 1.03-1.10, depending on the atomtype) to reproduce the Molpro energies. This scaling can be carried out by executing the command

```
./scripts/get_scaled_dispersion.py <scale_factor>
```

where `<scale_factor>` is chosen to reproduce the asymptotic Molpro DFT-SAPT energies (see Chapter 8 for details). This choice may require some testing, but 1.10 is usually a good default. The above script outputs files `dispersion/<monomer>.disp`, which can be used as input to the POInter program discussed in Chapter 8.

values calculated as in Section 7.5.4. Consequently, iDMA-pol is able to perform better as an ‘effectively anisotropic’ model. In principle, changing the defaults in CamCASP to use weight type 4 (which uses dipole-dipole terms as anchors, but completely fits higher ranking terms and thus fits the p2p better) or 3 (uses all terms as anchors) and a weight coefficient of 1e-5 (rather than 1e-3) should yield dispersion coefficients more similar to iDMA-pol, though this idea requires further testing.

* Additional reasons for discrepancies between CamCASP and MOLPRO dispersion coefficients include the following:

1. For PBE calculations, CamCASP uses ALDA with PW91c correlation, whereas Molpro uses VWN
2. CamCASP writes kernels completely in the auxiliary basis set, whereas Molpro writes the kernel in a variety of basis sets

7.5.5 Polarization Charges

Theory In addition to frequency-dependent polarizabilities, some of the same techniques described in Section 7.5.4 can be applied to obtain the static polarizabilities that get used in modeling the SAPT induction energy. Though in principle ISA-based polarizabilities could be used, this technique has not yet been developed. Instead, an iDMA-pol-type procedure can be used to extract the necessary polarization parameters. The algorithms used to perform this procedure are described in Appendix A of Ref. 29. Due to the reduced number of coefficients that need to be fit, this optimization is generally more robust, and leads to more transferable dispersion parameters, than do the algorithms described in Section 7.5.4.

Practicals The drude oscillator fitting code has the same dependencies and input files as iDMA-pol, with the exception that the `dispersion_base_constraints.index` file is replaced with the following constraint file:

1. `drude_base_constraints.index`: As with iDMA-pol, it is usually advisable to only fit a few atomtype static polarizabilities at a time, with the remaining atomtype polarizabilities read in as hard constraints. The `drude_base_constraints.index` file lists these hard constraints in a block format,

```
C
1
0.0

N
1
-11.7529643

H
1
-1.254
```

which lists each constrained atomtype along with the rank 1 static polarizabilities. Each block should be separated by a blank line. Unlike with the generation of dispersion coefficients, an initial guess must be given for *all* atomtypes in the `<monomer>.atomtypes` input file. The format for the

`drude_base_constraints.index` is such that positive polarizabilities correspond to these initial guesses, whereas zero or negative entries for the polarizabilities indicate that the atomtype should be treated as a hard constraint. Previously-fit atomtype polarizabilities from Ref. 28 are already included in `drude_base_constraints.index` so as to minimize the number of hard constraints that the user will need to add manually, and these hard constraints should be used whenever possible.

Assuming that the iDMA-pol calculations have already been run in CamCASP, the drude oscillator coefficients can be obtained simply by executing

```
./scripts/workup_drude_files.sh
```

As with the dispersion coefficients, care should be taken to ensure that the resulting drude oscillator charges are physically-meaningful (i.e. negative).

7.6 Dimer-Based Parameterization

After obtaining monomer parameters for a given system of interest, the final remaining task is to fit the remaining force field parameters to reproduce the DFT-SAPT calculations performed earlier in the Workflow. Dimer-based parameterization is carried out by the POInter program, which will be the subject of the next chapter. The Workflow is useful for preparing input files for this dimer-based parameterization, as follows:

```
./scripts/workup_sapt_energies.py
./scripts/gather_pointer_input_files.py
```

The output of these scripts will generate a `.sapt` file (containing results from the DFT-SAPT calculations, with atomtype labels taken from each `input/<monomer>.atomtypes` file) and a new directory, `ff_fitting`, which automatically sets up all of the input files and monomer parameters needed to easily run the POInter fitting code. The theory and practice of the POInter fitting code is the subject of the next chapter.

7.A Input Scripts

In total, the workflow for force field development requires four input files, as follows:

Listing 7.1: generate_grid_settings.inp

```
# Generate Grid Settings file. Version 04.28.15
#
# General Scan Parameters:
n_points      1000 # Number of grid points (.xyz files) to output
geometry_file  pyridine_pyridine.inp #name of geometry file
output_name    pyridine_pyridine      #output file base name

# Hard Sphere cutoff parameters:
#
# Parameters below are used to define minimum and maximum acceptable distances
# for neighbor-neighbor interactions. 'cutoff_type' can either be set to
# 'absolute' or 'vdw'. In the former case, the hard sphere cutoff will be set
# to the absolute distances (in Angstroms) given by cutoff_min and cutoff_max,
# respectively. In the latter case, the hard sphere cutoff will be set to a
# fraction of the Van der Waals distance between two atoms.
cutoff_type    vdw # either vdw or absolute
cutoff_min     0.8 # a positive float (ex. 0.8 for vdw or 2.0 for absolute)
cutoff_max     1.3 # a positive float (ex. 1.2 for vdw or 6.0 for absolute)

# The following are parameters defining the centers of monomer's a and b as well as the
# scan
# vector.
#
# The 'center' of each monomer is defined by default to be each
# monomer's center of mass, but can also be set to be either the center of an
# atom or a point in 3-space (relative to monomer coordinates given in input
# geometry file).
mona_origin_type 1 # choose 0 for center of mass (COM), 1 for atom#, and 2 for a
# specific point
mona_origin      6 # (either 'COM', point x,y,z , or atom# in monomer (indexing
# starts at 1), depending on choice of mona_origin_type above)
monb_origin_type 1 # choose 0 for COM, 1 for atom#, and 2 for a specific point
monb_origin      6 # (either 'COM', point x,y,z , or atom# in monomer (indexing
# starts at 1), depending on choice of mona_origin_type above)

# The scan vector should be a vector (given relative to the coordinates in
# monomer a) that defines the direction of internuclear separation between the
# two monomers. It can either be given as a 3-membered list or by listing two
# monomer indices (scan vector will point from atom1 to atom2, indexing starts at 1).
```



```

scan_vector_type 0 # choose 0 for monomer indices, 1 for a specific point
scan_vector      9,6 # Give either as a 2 (if scan_vector_type==0) or a 3 (if
                    scan_vector_type==1) membered,
                    # comma seperated list without spaces, i.e. '1.0,2.7,4.2' (no
                    quotes)

# Set bounds on moving the center of monomer b relative to the center of
# monomer a. min/max_r refers to the distance between the centers, while theta
# and phi correspond to the azimuthal and polar angles, respectively, of
# rotation about the vector scan_vector (given above).
#
# Give min/max angles as either integers/floats in terms of pi (i.e. setting
# 'max_theta 2' (no quotes) will yield max_theta=2pi).
min_r            2.0
max_r            8.0
min_theta        0
max_theta        2
min_phi          0
max_phi          1

```

Listing 7.2: dimer_info.dat

```

#####
# DIMER INFORMATION FILE #
#####

# String names for monomers A and B:
# -----
MonA_Name        pyridine
MonB_Name        pyridine

# Charges for monomers A and B:
# -----
MonA_Charge      0
MonB_Charge      0

# Midbond position(s); two integers indicating atom indices (indexed from 1)
# on monomers A and B, respectively, between which to place the midbond site.
# In lieu of an integer, COM can also be used to indicate the center of mass
# of the monomer.
# Multiple arguments can be given to produce multiple midbond functions.
# -----
midbond          com    com

```

Listing 7.3: pyridine_pyridine.inp

Pyridine Dimer; Optimized with PBE0/cc-pVTZ Gaussian03 by AJ Misquitta

```

11
H      -2.050322    1.274414    0.000000
H      -2.147113   -1.203259    0.000000
H       0.000000   -2.487558    0.000000
H       2.147113   -1.203259    0.000000
H       2.050322    1.274414    0.000000
N       0.000000    1.382844    0.000000
C      -1.134410    0.690452    0.000000
C      -1.190513   -0.695795    0.000000
C       0.000000   -1.403912    0.000000
C       1.190513   -0.695795    0.000000
C       1.134410    0.690452    0.000000
11
H      -2.050322    1.274414    0.000000
H      -2.147113   -1.203259    0.000000
H       0.000000   -2.487558    0.000000
H       2.147113   -1.203259    0.000000
H       2.050322    1.274414    0.000000
N       0.000000    1.382844    0.000000
C      -1.134410    0.690452    0.000000
C      -1.190513   -0.695795    0.000000
C       0.000000   -1.403912    0.000000
C       1.190513   -0.695795    0.000000
C       1.134410    0.690452    0.000000

```

Listing 7.4: pyridine.atomtypes

Pyridine Dimer; Optimized with PBE0/cc-pVTZ Gaussian03 by AJ Misquitta

```

11
H      -2.050322    1.274414    0.000000
H      -2.147113   -1.203259    0.000000
H       0.000000   -2.487558    0.000000
H       2.147113   -1.203259    0.000000
H       2.050322    1.274414    0.000000
N       0.000000    1.382844    0.000000
C      -1.134410    0.690452    0.000000
C      -1.190513   -0.695795    0.000000
C       0.000000   -1.403912    0.000000
C       1.190513   -0.695795    0.000000
C       1.134410    0.690452    0.000000
11
H      -2.050322    1.274414    0.000000
H      -2.147113   -1.203259    0.000000
H       0.000000   -2.487558    0.000000
H       2.147113   -1.203259    0.000000
H       2.050322    1.274414    0.000000

```

N	0.000000	1.382844	0.000000
C	-1.134410	0.690452	0.000000
C	-1.190513	-0.695795	0.000000
C	0.000000	-1.403912	0.000000
C	1.190513	-0.695795	0.000000
C	1.134410	0.690452	0.000000

7.B Algorithm for Obtaining ISA Exponents

Unphysical asymptotic charge density decays occasionally arise in the ISA procedure due to basis set incompleteness and numerical instabilities. These unphysical decays can skew optimization of ISA-based exponents, B_i^{ISA} , and need to be corrected. Generally speaking, there exists some range of distances in the valence region that *does* exhibit the expected exponential decay; we extrapolate the decay from this intermediate region to describe the asymptotic region using the following algorithm:

1. Take the log of each atomic density (henceforth logdens) to linearize the asymptotic density.
2. Compute the 2nd derivative of logdens. This can be done analytically, as the ISA procedure outputs an analytical expression (in terms of Gaussian basis functions) for the atomic density.
3. Determine the ‘intermediate region’ of exponential decay by locating the largest range where the 2nd derivative of logdens is zero to within a fixed tolerance. Here we utilize a tolerance of 0.3 a.u. (absolute cutoff) or 190% of the smallest exponent in the Gaussian basis set (relative cutoff), whichever is smaller. The latter cutoff accounts for the eventual asymptotic Gaussian-type decay dictated by the smallest ζ in the ISA basis. The endpoints of this intermediate region are denoted $r1$ and $r2$, respectively.
4. Calculate the slope m and intercept b for the line defined by $r1$, $r2$, and their respective values of logdens.

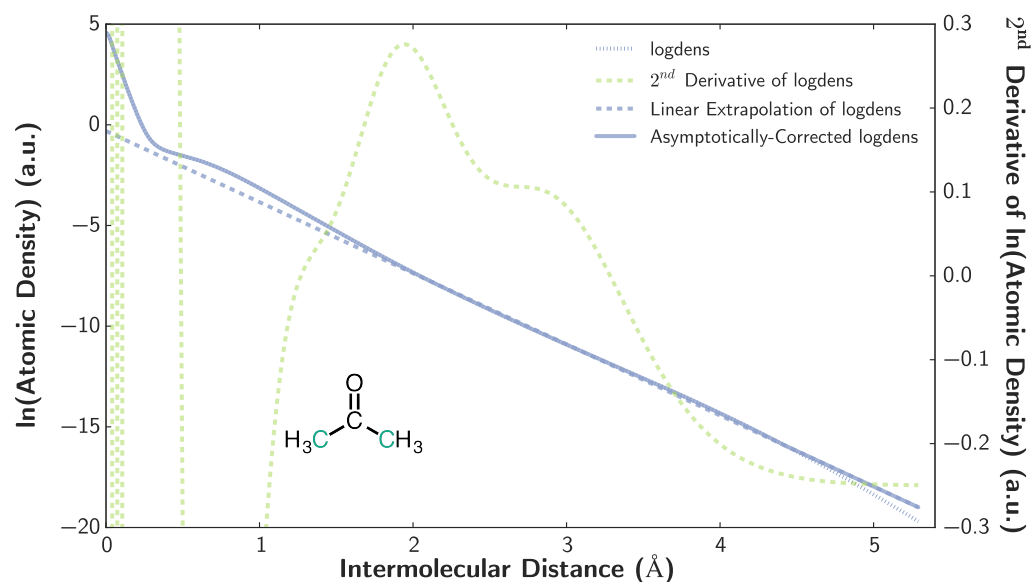


Figure 7.2: Linear extrapolation algorithm for the methyl carbon in acetone. Depicted are (in legend order) Steps 1, 2, 4, and 5 in the extrapolation algorithm. Note that some portions of the 2nd derivative extend off the graph; also note that most of logdens is located underneath the asymptotically-corrected curve.

5. Replace all values of logdens after r_2 with $mr + b$. The resulting atomic density is labeled in the main text as 'Asymptotically-corrected ISA densities'.

A visual of these steps is shown in 7.2.

8 POINTER: A PROGRAM FOR INTERMOLECULAR FORCE FIELD OPTIMIZATION

8.1 Overview

8.1.1 What is POInter?

overview

8.1.2 What parameters do we get from POInter?

=====

What is POInter

input and output parameters (as a workflow chart?) or as highlighted equation workflow

any important algorithmic details of the code

required input sapt file anisotropy file settings file (for advanced users) defaults file

tricky (i.e. not-yet-standardized) aspects of parameterization ccsdt correction electrostatics electrostatic damping (discuss how to change sapt file!) off-site point charges polarization damping - how to choose? exponents - when to fit vs. hold fixed? dispersion - when and how to fit (ccsdt correction vs sapt, how to choose asymptotic correction factor, caution in fitting)

documentation of various options and settings

wiki

Release version number

8.2 Documentation

8.3 Examples

8.A Input Files

In total, POInter requires modification of three input files, as follows:

Listing 8.1: settings.py

```
#####
##### General Settings #####
#####
# Monomer Names (should match ordering in .sapt file)
mon1          = 'chloromethane'
mon2          = 'chloromethane'

# Constrained Atomtype settings: Make a list of all atomtypes whose parameters
# should *not* be fit, and include parameters for these atomtypes in the
# relevant <monomer>.constraints file
constrained_atomtypes = []

# Names for output files
file_prefix      = 'fit_exp_'
file_suffix      = '_unconstrained'

#####

#####
##### Component-Specific Settings #####
#####
# Electrostatic Settings: choose which multipole files the program should use
multipoles_suffix = '_ISA-GRID_L2.mom'

# Exchange Settings: fit_bii selects whether or not to treat the ISA
# short-range exponents are soft- (fit_bii=True) or hard-constraints
# (fit_bii=False)
fit_bii          = True

# Induction Settings: Choose the type and parameters for the polarization
# damping functions. Options for thole_damping_type are 'thole_tinker' and
# 'thole_linear', and good defaults for thole_param are the 0.33 and 2.0 with
# respect to the two different damping types
# respectively
thole_damping_type = 'thole_tinker'
thole_param        = 0.33

# Dispersion Settings: Choose which parameters to fit to the dispersion energies. Fit
# options
# include 'none' (to fit no parameters), 'anisotropic' (to just fit
# anisotropic dispersion parameters, but to leave isotropic dispersion
# coefficients unscaled), and 'all' (to fit both anisotropic and isotropic
```

```

# dispersion coefficients)
fit_dispersion          =    'anisotropic'

# Residual error Settings: If set to true, fits a final A parameter to errors in the
# total
# energy in an effort to reduce systematic errors in the total energy
fit_residuals           =    False

#####

#####

#####
##### Functional Form Settings #####
#####
# Radial functional forms f(r); see Stone's book for more details.
# Options are 'slater', 'stone', 'born-mayer', 'born-mayer-sisa', or 'lennard-jones'
functional_form         =    'slater'

# Combination rule settings: Select combination rules for each A prefactors, B
# exponents, and C dispersion coefficients. Options are as follows:
#   aij: 'saptff', 'waldman-hagler5', 'geometric'
#   bij: 'saptff', 'waldman-hagler5', 'geometric_mean', 'arithmetic_mean'
#   cij: 'geometric'
aij_combination_rule    =    'geometric'
bij_combination_rule    =    'geometric_mean'
cij_combination_rule    =    'geometric'

#####
#####

```


Listing 8.2: defaults.py

```
#####
##### POInter Defaults #####
#####
# The following defaults should be used for most routine force field
# development, however advanced users may wish to change some of the
# following settings:
exponent_source          = 'ISA'
lone_pair_flags          = ['Du' , 'lp']
scale_weighting_temperature = 5.0
separate_induction_exponents = False
springcon                = 0.1
weighted_rmse_cutoff     = 0.0
electrostatic_damping_type = 'None'
include_slater_charge_penetration = False
induction_damping_type    = 'Thole'

# Unless you know what you're doing, the following settings should only be
# changed by developers:
__version__ = '1.1.0'

#####
#####
```

Part V

Conclusions and Future Work

9 FUTURE WORK

10 CONCLUSIONS

Part VI

Codes

A FORCE FIELD DEVELOPMENT WORKFLOW

Purpose

Derive a first-principles, SAPT-based force field.

Relevant Literature

- VanVleet2016: 10.1021/acs.jctc.6b00209
- VanVleet2017: TBA
- McDaniel2013: 10.1021/jp3108182
- Schmidt2015: 10.1021/ar500272n
- Yu2011: 10.1021/jp204563n

Overview

To generate a SAPT-based force field, the following inputs are required: 1. Benchmark dimer energies from SAPT, computed for a variety of dimer configurations 2. Long-range multipole moments, induced dipoles, and dispersion parameters, computed from monomer properties (and BS-ISA in particular) 3. Short-range exponents computed from monomer properties (and BS-ISA in particular) 4. Short-range pre-factors fit to dimer energies

The following scripts are designed to simplify (as much as is possible) the workflow for force field generation.

Method

1. Generate the necessary input files upon which the scripts in step #2 depend. The following files must be manually created/edited, and can all be found in the templates subdirectory (with an example set of input files given for the pyridine dimer):
 1. `dimer_info.dat`
 - For each monomer, list the monomer's name and the charge on the monomer. The appropriate file format should be clear from the pyridine example.
 - In the manner described in `dimer_info.dat`, list all midbonds that should be added between monomers. Midbonds are important for running accurate SAPT calculations; see Yu2011 for details.
 2. `generate_grid_settings.inp`
 - This is the input file for `GenerateGridPoints`, which generates the dimer configurations for running SAPT calculations. The input file is commented so as to be self-explanatory; you will need to change (at the very least) the 1st, 3rd, and 4th input sections based on the identities of the two monomers
 3. `MONA_MONB.inp` (where MONA and MONB are replaced by the monomer names listed in `dimer_info.dat`)
 - This file contains a title line (line 1), and (for each monomer) the number of atoms followed by a list of coordinates in .xyz format. See `pyridine_pyridine.inp` for an example.

4. MONA.atomtypes, MONB.atomtypes

- Each .atomtypes file has the format of a .xyz file, where the element names have been replaced by atomtypes. This file will be used to generate the CamCASP input files needed for ISA calculations, and is also necessary for pre-processing the input files for force field fitting.

2. To generate all files necessary to run force field calculations, run the following pre-processing scripts (from this main directory).

```
./scripts/make_geometries.sh
./scripts/get_global_coordinates.py
./scripts/submit_ip_calcs.py
```

(wait until IP calculation is finished)

```
./scripts/make_sapt_ifiles.py
./scripts/make_isa_files.py
./scripts/make_dispersion_files.py
```

3. Submit all SAPT and ISA calculations to relevant locations. At the time of this writing, SAPT calculations should preferably be run on HCTC (Condor). ISA and dispersion calculations should be run on Phoenix using Camcasp 5.8. Copy all output files back to Pople.

4. Workup the results of the SAPT and ISA calculations by running the following post-processing scripts:

```
./scripts/workup_sapt_energies.py
./scripts/workup_dispersion_files.sh
```

(Depending on the force field, dynamic polarizabilities may need to be added to templates/dispersion_base_constraints.index before running this script. See Jesse McDaniel's thesis and \cite{McDaniel2013} for a full description of the parameterization process for dispersion coefficients.)

```
./scripts/workup_drude_files.sh
```

(Depending on the force field, static polarizabilities may need to be added to templates/drude_base_constraints.index before running this script. See Jesse McDaniel's thesis and \cite{McDaniel2013} for a full description of the parameterization process for drude oscillator charges.)


```
./scripts/workup_isa_charges.py  
./scripts/workup_isa_exponents.py
```

After running these scripts, you should have the SAPT energies, long-range coefficients, and short-range exponents required to run the force fitting code (which is needed to generate short-range pre-factors, see \cite{VanVleet2016}). The proper running of this code is described in the POInter documentation, see

https://git.chem.wisc.edu/schmidt/force_fields/wikis/home

Overview of Important Files

- dimer_info.dat <- monomer names and midbond positions
- dispersion_template.clt <- CamCASP input file for getting induction and dispersion paramters
- generate_grid_settings.inp <- geometry configuration settings
- isa_template.clt <- CamCASP input file for getting ISA exponents
- pbe0_template.com <- DF-DFT-SAPT template for the PBE0 functional
- pyridine.atomtypes <- change elements to atomtypes; only matters for dispersion
- pyridine_pyridine.inp <- monomer geometries

For most systems, only dimer_info.dat, the .inp files, and the .atomtypes file will need to be changed. The examples provided for these files should hopefully make the format self-explanatory.

System Requirements

Python dependencies: * numpy * scipy * chemistry (mvanvleet package; not standard, so this needs to be * downloaded and added to your `$PYTHONPATH`)

Figure A.1: An overview of the semi-automated force field development process. The full workflow and required scripts can be found at <https://github.com/mvanvleet/workflow-for-force-fields>.

BIBLIOGRAPHY

-
- [2] Stone, A. J. *The Theory of Intermolecular Forces*, 2nd ed.; OUP Oxford, 2013.
 - [3] Furukawa, H.; Cordova, K. E.; O’Keeffe, M.; Yaghi, O. M. *Science* (80-.). **2013**, *341*, 1230444–1230444.
 - [4] Millward, A. R.; Yaghi, O. M. *J. Am. Chem. Soc.* **2005**, *127*, 17998–17999.
 - [5] Dietzel, P. D. C. et al. *J. Mater. Chem.* **2009**, *19*, 7362.
 - [6] Dzubak, A. L.; Lin, L.-C.; Kim, J.; Swisher, J. a.; Poloni, R.; Maximoff, S. N.; Smit, B.; Gagliardi, L. *Nat. Chem.* **2012**, *4*, 810–816.
 - [7] Czaja, A. U.; Trukhan, N.; Müller, U. *Chem. Soc. Rev.* **2009**, *38*, 1284.
 - [8] Krishna, R.; van Baten, J. M. *Phys. Chem. Chem. Phys.* **2011**, *13*, 10593–10616.
 - [9] Getman, R. B.; Bae, Y.-s.; Wilmer, C. E.; Snurr, R. Q.; Carlo, M. *Adsorpt. J. Int. Adsorpt. Soc.* **2012**, 703–723.
 - [10] Yazaydin, a. O.; Snurr, R. Q.; Park, T.-H.; Koh, K.; Liu, J.; Levan, M. D.; Benin, A. I.; Jakubczak, P.; Lanuza, M.; Galloway, D. B.; Low, J. J.; Willis, R. R. *J. Am. Chem. Soc.* **2009**, *131*, 18198–9.
 - [11] Valenzano, L.; Civalleri, B.; Chavan, S.; Palomino, G. T.; Areañ'n, C. O.; Bordiga, S. *J. Phys. Chem. C* **2010**, *114*, 11185–11191.
 - [12] Poloni, R.; Lee, K.; Berger, R. F.; Smit, B. **2014**,
 - [13] Lin, L.-c.; Lee, K.; Gagliardi, L.; Smit, B. *J. Chem. Theory Comput.* **2014**, *10*, 1477–1488.
 - [14] Haldoupis, E.; Borycz, J.; Shi, H.; Vogiatzis, K. D.; Bai, P.; Queen, W. L.; Gagliardi, L.; Siepmann, J. I. *J. Phys. Chem. C* **2015**, *119*, 16058–16071.

- [15] Mercado, R.; Vlaisavljevich, B.; Lin, L.-C.; Lee, K.; Lee, Y.; Mason, J. A.; Xiao, D. J.; Gonzalez, M. I.; Kapelewski, M. T.; Neaton, J. B.; Smit, B. *J. Phys. Chem. C* **2016**, acs.jpcc.6b03393.
- [16] Becker, T. M.; Heinen, J.; Dubbeldam, D.; Lin, L.-C.; Vlugt, T. J. H. *J. Phys. Chem. C* **2017**, acs.jpcc.6b12052.
- [17] McDaniel, J. G.; Schmidt, J. R. *J. Phys. Chem. C* **2012**, 116, 14031–14039.
- [18] McDaniel, J. G.; Yu, K.; Schmidt, J. R. *J. Phys. Chem. C* **2012**, 116, 1892–1903.
- [19] McDaniel, J. G.; Li, S.; Tylianakis, E.; Snurr, R. Q.; Schmidt, J. R. *J. Phys. Chem. C* **2015**, 119, 3143–3152.
- [20] Lao, K. U.; Schaeffer, R.; Jansen, G.; Herbert, J. M. *J. Chem. Theory Comput.* **2015**, 150417132228001.
- [21] Pastorczak, E.; Corminboeuf, C. *J. Chem. Phys.* **2017**, 146, 120901.
- [22] Żuchowski, P. *Chem. Phys. Lett.* **2008**, 450, 203–209.
- [23] Horn, P. R.; Head-gordon, M. *Phys. Chem. Chem. Phys.* **2016**, 18, 23067–23079.
- [24] Su, P.; Li, H. *J. Chem. Phys.* **2009**, 131, 014102.
- [25] Chen, Y.; Li, H. *J. Phys. Chem. A* **2010**, 114, 11719–24.
- [26] Su, P.; Jiang, Z.; Chen, Z.; Wu, W. *J. Phys. Chem. A* **2014**, 118, 2531–42.
- [27] Fedorov, D. G.; Kitaura, K. **2006**,
- [28] McDaniel, J. G.; Schmidt, J. R. *J. Phys. Chem. A* **2013**, 117, 2053–2066.
- [29] McDaniel, J. G. Development and Application of Physically-Motivated First-Principles Force Fields for Complex Chemical Systems. Ph.D. thesis, UW-Madison, 2014.
- [30] Yu, K.; Kiesling, K.; Schmidt, J. R. **2012**,

- [31] Verma, P.; Xu, X.; Truhlar, D. G. **2013**,
- [32] Valenzano, L.; Civalleri, B.; Sillar, K.; Sauer, J. **2011**, 21777–21784.
- [33] Haldoupis, E.; Borycz, J.; Shi, H.; Vogiatzis, K. D.; Bai, P.; Queen, W. L.; Gagliardi, L.; Siepmann, J. I. *J. Phys. Chem. C* **2015**, 74, 150616135429005.
- [34] Yu, K.; McDaniel, J. G.; Schmidt, J. R. *J. Phys. Chem. B* **2011**, 115, 10054–10063.
- [35] Misquitta, A. J. *J. Chem. Theory Comput.* **2013**, 9, 5313–5326.
- [36] Jansen, G.; Scha, R. **2012**,
- [37] McDaniel, J. G.; Schmidt, J. R. *J. Phys. Chem. B* **2014**, 118, 8042–8053.
- [38] Frenkel, D.; Smit, B. *Acad. Press*; 2002; Vol. New York,; p 638.
- [39] Guibas, L. Representing rotations with quaternions. 1992; graphics.stanford.edu/courses/cs164-09-spring/Handouts/handout12.pdf.
- [40] Shoemake, K. *Graph. Gems 3*; 1992; Chapter 6, pp 124–132.
- [41] Jeziorski, B.; Moszynski, R.; Szalewicz, K. *Chem. Rev.* **1994**, 94, 1887–1930.
- [42] Szalewicz, K. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2012**, 2, 254–272.
- [43] Knizia, G.; Adler, T. B.; Werner, H.-J. *J. Chem. Phys.* **2009**, 130, 054104.
- [44] Misquitta, A. J.; Stone, A. J. CamCASP: a program for studying intermolecular interactions and for the calculation of molecular properties in distributed form, version 5.8. University of Cambridge, 2015.
- [45] Stone, A. J. *J. Chem. Theory Comput.* **2005**, 1, 1128–1132.
- [46] Misquitta, A. J.; Stone, A. J.; Fazeli, F. J. *Chem. Theory Comput.* **2014**, 10, 5405–5418.
- [47] Misquitta, A. J.; Stone, A. J. *J. Chem. Phys.* **2006**, 124, 024111.

- [48] Van Vleet, M. J.; Misquitta, A. J.; Stone, A. J.; Schmidt, J. R. *J. Chem. Theory Comput.* **2016**, *12*, 3851–3870.
- [49] Cardamone, S.; Hughes, T. J.; Popelier, P. L. a. *Phys. Chem. Chem. Phys.* **2014**, *16*, 10367.
- [50] Kramer, C.; Spinn, A.; Liedl, K. R. *J. Chem. Theory Comput.* **2014**, *10*, 4488–4496.
- [51] Dixon, R. W.; Kollman, P. a. *J. Comput. Chem.* **1997**, *18*, 1632–1646.
- [52] Chaudret, R.; Gresh, N.; Cisneros, G. A.; Scemama, A.; Piquemal, J.-p. *Can. J. Chem.* **2013**, *91*, 804–810.
- [53] Unke, O. T.; Devereux, M.; Meuwly, M. *J. Chem. Phys.* **2017**, *147*, 161712.
- [54] Ferenczy, G. G.; Winn, P. J.; Reynolds, C. a. *J. Phys. Chem. A* **1997**, *101*, 5446–5455.
- [55] Williams, G. J.; Stone, A. J. *J. Chem. Phys.* **2003**, *119*, 4620–4628.
- [56] Misquitta, A. J.; Stone, A. J. *Mol. Phys.* **2008**, *106*, 1631–1643.

ACRONYMS

C | D | I | M | P | S**C**

CUS Coordinatively-Unsaturated. iv

D

DMA Distributed Multipole Analysis. v, vii

I

isa Iterated Stockholder Atoms. v, vii, *Glossary*: ISA

M

MOF Metal-Organic Framework. iv

P

PES potential energy surface. viii, 78

POInter Parameter Optimizer for **I**nter-molecular Force Fields. 63

S

SAPT Symmetry-Adapted Perturbation Theory. v, viii, *Glossary*: SAPT

GLOSSARY

C | I | L | P | S

C

CCSD(T) Coupled Cluster methods including singles, doubles, and perturbative triples excitations. CCSD(T). Given a sufficiently large (aVQZ or better) basis set, can be used as a 'gold-standard' estimate of the exact potential energy surface. v

I

ISA Iterated Stockholder Atoms, FILL . v

L

LMO-EDA FILL . iv, viii

P

POInter FILL . 63

S

SAPT Symmetry-Adapted Perturbation Theory, a perturbative treatment of inter-molecular interactions which is pretty cool. v