

**NEW 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III Unpublished Work 8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 *New Methods for Coordinatively-Unsaturated (CUS)-Metal-Organic Framework (MOF) force fields* 13
 - 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 32

IV Practical Matters 33

- 7 Workflow for Intermolecular Force Field Development 34

7.1	<i>Overview</i>	34
7.2	<i>Geometry Generation</i>	36
7.2.1	Guiding Principles	36
7.2.2	Theory	38
7.2.3	Practicals	39
7.3	<i>Symmetry-Adapted Perturbation Theory (SAPT) Benchmarks</i>	39
7.4	<i>CCSD(T) Calculations</i>	40
7.5	<i>Monomer-Based Parameterization</i>	41
7.5.1	The CamCASP software	41
7.5.2	Multipoles	42
	Overview	42
	Advanced Multipole Parameterization Options	43
7.5.3	ISA Exponents	45
7.5.4	Dispersion Coefficients	45
	Theory	45
	Iterative-Distributed Multipole Analysis (DMA)-pol	47
	Iterated Stockholder Atoms (ISA)-pol	50
	Theory	50
	Practicals	51
	Method Comparison	51
	Dispersion Coefficient Post-processing	52
7.5.5	Polarization Charges	54
	Theory	54
	Practicals	54
7.6	<i>Dimer-Based Parameterization</i>	55
7.A	<i>Input Scripts</i>	56
7.B	<i>Extrapolation Algorithm for ISA Exponents</i>	59
8	POInter: A Program for Intermolecular Force Field Optimization	61
8.1	<i>Overview</i>	61
8.2	<i>Documentation</i>	61

8.3 *Examples* 61

V Conclusions and Future Work 62

9 Future Work 63

10 Conclusions 64

VICodes 65

A Force Field Development Workflow 66

Bibliography 70

Acronyms 73

Glossary 75

LIST OF TABLES

7.1	Overview of ISA- and DMA-based methods for obtaining distributed monomer properties	42
7.2	Comparison between the iDMA-pol and ISA-pol methods.	53

LIST OF FIGURES

5.1	Model potential energy surface (PES) for interactions between CO ₂ and Mg-MOF-74	12
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	The workflow for SAPT-based force field development.	36
7.2	Generalized form of a PES showing the repulsive wall, minimum energy, and asymptotic regions.	37
7.3	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.	61
A.1	The Semi-Automated Workflow for Force Field Development	69

NEW 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

- [40] 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!)

Part II

Published Work

Part III

Unpublished Work

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 water force field 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 requires us to parameterize the 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 interactions based on benchmark SAPT energies for a series of gas-phase dimer configurations.[†] We are usually interested in ob-

[†]This philosophy of force field development was one of the most counter-intuitive ideas I had to learn as a grad student. Still, and regardless of whether we are ultimately interested in studying

taining 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 interactions.

Regardless of the chosen dimer of study, developing force fields for the 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 describe a model 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:

Aside from the last step of this workflow, which will be the subject of 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.

a homogeneous liquid or a heterogeneous supercritical phase, the best starting point for ab initio force field development is always to model all relevant gas-phase dimer interactions.

*In general, force field development based on homo-monomeric interactions involves the fewest atomtypes, and thus the fewest number of free parameters. Alternately, hetero-monomeric-based force field development can yield the best accuracy for studying systems where either transferability is difficult (see Chapter 5 for an example) or computational expense is an issue. (For instance, running SAPT calculations on a naphthalene dimer may be infeasible, whereas studying naphthalene-Ar interactions is easily possible.)

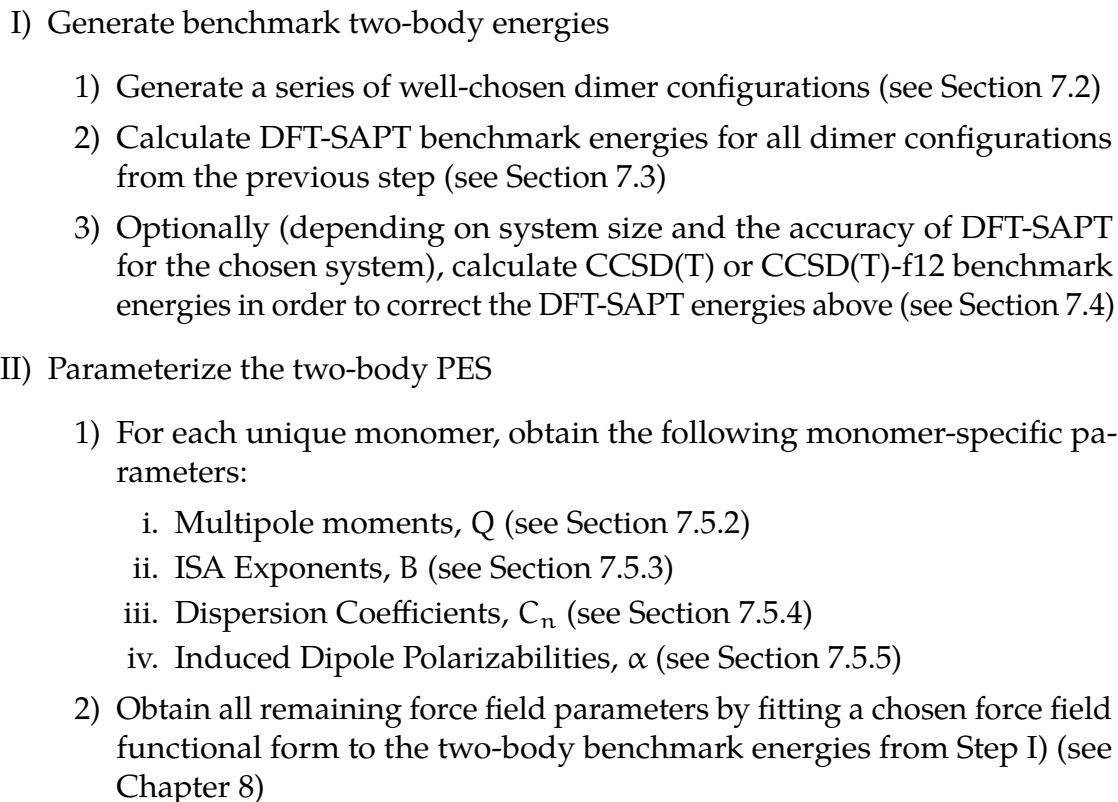
- 
- I) Generate benchmark two-body energies
 - 1) Generate a series of well-chosen dimer configurations (see Section 7.2)
 - 2) Calculate DFT-SAPT benchmark energies for all dimer configurations from the previous step (see Section 7.3)
 - 3) Optionally (depending on system size and the accuracy of DFT-SAPT 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)

Figure 7.1: The workflow for SAPT-based force field development.

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. 7.2, a given PES will have (qualitatively) three different regions: a repulsive wall, a minimum energy region, and an asymptotic

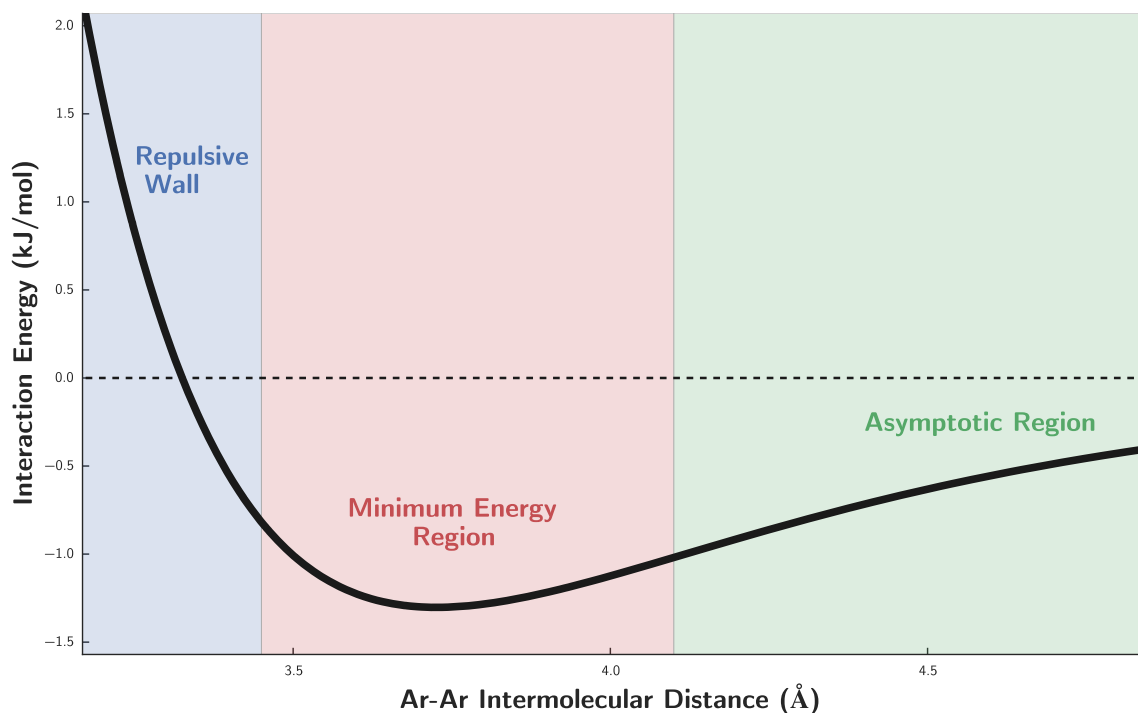


Figure 7.2: Generalized form of a PES showing the repulsive wall, minimum energy, and asymptotic regions of the argon dimer. Cutoffs between the different regions should be taken qualitatively.

region (which is often attractive, but which may be repulsive due to unfavorable electrostatic interactions). Due to the nature of the Boltzmann distribution, *routine molecular simulation will most frequently sample the minimum energy and asymptotic regions of the potential, making these portions of the PES the most important to model correctly*. 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 fitted force field parameters largely determine the shape of the repulsive wall and (to a lesser

*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.

extent) the minimum energy regions, and so in practice we primarily focus on including dimer configurations in these regimes.* In other words, and based on a combination of importance in molecular simulation and fit sensitivity to the specific parameters which we are directly optimizing, *the repulsive wall and especially the minimum energy regions are the most important to effectively sample in order to achieve good force field fits.*

As described below, a standard procedure for the properly sampling across the PES has been established for the Slater-ISA FF and MASTIFF force fields. Only when fitting different functional forms will the user need to reconsider the relative sampling of dimer configurations.

7.2.2 Theory

Assuming rigid monomer geometries, a dimer configuration can be completely determined (without loss of generalization) by fixing the first monomer at the origin and by placing the second monomer according to the following six variables: r , θ , and ϕ determine the position of center of mass of the second monomer, and the three-dimensional variable Ω determines the relative orientation of the monomer about its center of mass. In practice, Ω is most easily described by a quaternion approach, see Ref. 31.

For both the Slater-ISA FF and MASTIFF fits, dimer configurations are sampled psuedo-randomly using Shoemake’s algorithm³² (which ensures even sampling across the center of mass orientation) subject to some exclusion criteria. In particular, and in order to achieve representative sampling of the repulsive wall and minimum energy regions, the following dimer configurations are excluded from sampling:

1. Configurations with any atom-atom contact distance $r_{ij} \leq 0.8 \times (r_i^{\text{vdw}} + r_j^{\text{vdw}})$,

*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 is highly sensitive to the relative representation of repulsive and attractive dimer configurations. (Including both too few or too many repulsive configurations can be problematic). Only with the advent of Slater-ISA FF and MASTIFF has the fit quality become strictly improved by including repulsive configurations.

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 $r_{ij} \geq 1.3 \times (r_i^{\text{vdw}} + r_j^{\text{vdw}})$

7.2.3 Practicals

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

Once these input files have been modified, the geometry generation process can be carried out very simply from the main force field fitting directory by executing the command

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

7.3 SAPT Benchmarks

After geometry generation, the next step 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. 24, 33, 34. DFT-SAPT calculations can be performed in a fairly black-box manner, though the points are worth mention:

1. For best accuracy, and as described in Ref. 29, the monomer Density Functional Theory (DFT) calculations need to be asymptotically-corrected (AC) for best accuracy. This asymptotic correction is computed as the difference between the HOMO and the vertical ionization potential for each monomer, and is 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 in the same basis as the DFT-SAPT calculations themselves.

2. Accurate SAPT dispersion energies generally require use of midbond functions, as described in Ref. 29. 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 sufficiently accurate for most systems, though an aVTZ+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.³⁵ 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

7.5.1 The CamCASP software

CamCASP is a collection of scripts and programs useful for (among other things) the calculation of distributed multipoles and polarizabilities.³⁶ Of particular theoretical importance is the 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)³⁷ and Iterated Stockholder Atoms (ISA).³⁸ The theory behind the ISA procedure has already been detailed in Chapter 2, and monomer property calculations using DMA are described in Ref. 24, 37, 39. 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 a workflow for obtaining ISA-based dispersion coefficients are discussed below. A complete overview of available properties and distribution schemes, along with relevant references, are listed in Table 7.1.

Property	Parameterization Scheme	
	ISA	DMA
Multipoles	Section 7.5.2 Ref. 38	– Ref. 24, 37
Exponents	Section 7.5.3 Ref. 40	–
Polarizabilities	–	Section 7.5.5 Ref. 23
Dispersion Coefficients	Section 7.5.4 –	Section 7.5.4 Ref. 23

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

Overview

ISA-based multipoles are described in detail in Ref. 38, 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 notation 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, neglect) of the higher-order multipole moments. For most routine force field development, the L2 model is preferred, however below we discuss situations in which different electrostatic models may be desirable.

Advanced Multipole Parameterization Options

For many molecules, and 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 quadrupolar (i.e. 'rank 2' or L2)² contributions. Nevertheless, due to computational and/or software limitations, there may be practical cases where it becomes advantageous to exclude higher-order multipole moments.⁴¹ In these cases, two additional types of long-range electrostatic models are possible. For reasonably isotropic molecules, one option is to further truncate the ISA multipoles to point charge contributions, yielding a so-called 'atom-centered point charge model'. For more anisotropic functional groups, such as those described in Ref. 42, an atom-centered point charge model can be insufficiently accurate, making it necessary to model the long-range electrostatics with additional 'off-center' point charges. Given a well-chosen set of off-site charges, an off-site point charge model usually can reasonably reproduce the effects of the neglected higher-order multipole moments.⁴³ Locations for the off-center charges are often manually tuned or optimized in a system-specific manner, though some recent work suggests the possibility of using non-empirical methods to directly calculate/optimize positions for the extra-atom sites.^{44,45}

For an atom-centered point charge model, 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/ subdirectory of the workflow. Assuming the mulfit executable is in your \$PATH, the rank transformation can be performed using the following steps:

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

where 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 using a 4-site water model as an example. Note that 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 functions that describe the radial decay of the AIM density.³⁸ 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 ??).

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,^{2,12,23,24,47,48} and Ref. 24 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. Under

the simplifying assumption that we can treat these polarizabilities as isotropic, the dispersion energy expression is given by

$$E_{\text{disp}}^{\text{ab}} \approx -\frac{C_6^{\text{ab}}}{r_{\text{ab}}^6} - \frac{C_8^{\text{ab}}}{r_{\text{ab}}^8} - \dots \quad (7.1)$$

for each atom pair, where

$$C_6^{\text{ab}} = \frac{3}{\pi} \int_0^\infty \bar{\alpha}_1^{\text{a}}(i\omega) \bar{\alpha}_1^{\text{b}}(i\omega) d\omega, \quad (7.2)$$

$$C_8^{\text{ab}} = \frac{15}{2\pi} \int_0^\infty \bar{\alpha}_1^{\text{a}}(i\omega) \bar{\alpha}_2^{\text{b}}(i\omega) + \bar{\alpha}_2^{\text{a}}(i\omega) \bar{\alpha}_1^{\text{b}}(i\omega) d\omega, \quad (7.3)$$

and the higher order terms are defined analogously. C_n^{ab} are the atom-atom dispersion coefficients, and $\bar{\alpha}_l^{\text{a}}$ are the rank l , isotropic, AIM frequency-dependent polarizabilities. The formalisms in Eqs. (7.2) and (7.3) 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 environment.

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. (7.2) and (7.3), 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.⁴⁷ 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

As described in Ref. 24, 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,

$$\alpha_{PQ} = -q_Q \sum_{a,lm} T_{0,lm}^{Pa} \bar{\alpha}_l^a T_{lm,0}^{aQ} \quad (7.4)$$

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. (7.4) are directly calculated in CamCASP, enabling us to fit the isotropic polarizabilities based on a CamCASP properties calculation (see Appendix A of Ref. 24 for details).

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. (7.2), 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 applied. Previously-fit atomtype polarizabilities from Ref. 23 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. 23 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.²⁷ 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 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:

$$\alpha_{lm,l'm'}(\omega) = \iint \hat{Q}_{lm}(\mathbf{r}) \alpha(\mathbf{r}, \mathbf{r}' | \omega) \hat{Q}_{l'm'}(\mathbf{r}') d\mathbf{r} d\mathbf{r}' \quad (7.5)$$

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

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

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

$$1 = \sum_a \left(\frac{\bar{w}^a(\mathbf{r})}{\sum_m \bar{w}^m(\mathbf{r})} \right) = \sum_a \bar{p}_a(\mathbf{r}), \quad (7.6)$$

where the bars indicate that we have normalized the atom-in-molecule densities and weight functions. Substituting this equation into Eq. (7.5), we arrive at an ISA-based definition of the AIM polarizabilities:

$$\begin{aligned} \alpha_{lm,l'm'}(\omega) &= \iint \hat{Q}_{lm}(\mathbf{r}) \alpha(\mathbf{r}, \mathbf{r}' | \omega) \hat{Q}_{l'm'}(\mathbf{r}') d\mathbf{r} d\mathbf{r}' \\ &= \sum_a \sum_b \iint \hat{Q}_{lm}(\mathbf{r}) p_a(\mathbf{r}) \alpha(\mathbf{r}, \mathbf{r}' | \omega) p_b(\mathbf{r}') \hat{Q}_{l'm'}(\mathbf{r}') d\mathbf{r} d\mathbf{r}' \\ &\equiv \sum_a \sum_b \alpha_{lm,l'm'}^{ab}(\omega) \end{aligned} \quad (7.7)$$

While this formula bears similarity to DMA-based polarization approaches,^{47,48} the advantage of Eq. (7.7) 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 for up-to-date details and required input files.

Method Comparison

Preliminary results for the ISA-pol method, tested on the 91 dimer test set, 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.*

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). 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.

* A main difference between the iDMA-pol and ISA-pol coefficients is that iDMA-pol fits more strongly to the p2p file, whereas ISA-pol coefficients are set to the 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:

iDMA-pol	ISA-pol
Ease of Parameterization	
<ul style="list-style-type: none"> • For systems with 1-2 unparameterized atomtypes, little cost 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 at rank 3 • Physicality highly-dependent on 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.

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 drude oscillator charges that get used in modeling the 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. 24. 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 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,
-
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

```

C
1
0.0

N
1
-11.7529643

H
1
-1.254

```

which lists each constrained atomtype along with the rank 1 static polarizabilities, 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. 23 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 this section. Dimer-based parameterization is carried out by the POInter program, which will be the subject of the next chap-

ter. Still, the Workflow is useful for preparing input files for this dimer-based parameterization:

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

The output of these scripts will generate an output .sapt file (with atomtype descriptions taken from each input/<monomer>.atomtypes file and a new directory, ff_fitting, which sets up all of the input files and monomer parameters needed to easily run the POInter fitting code.

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 separated 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 Extrapolation Algorithm for 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 r_1 and r_2 , respectively.
4. Calculate the slope m and intercept b for the line defined by r_1 , r_2 , and their respective values of logdens.
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 7.3.

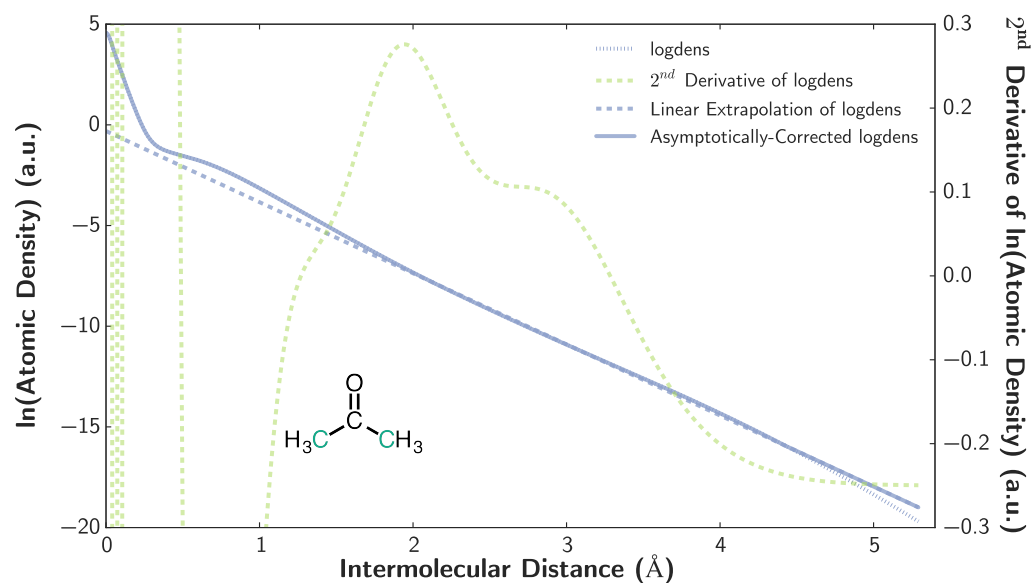


Figure 7.3: Linear extrapolation algorithm for the methyl carbon in acetone. Depicted are (in legend order) Steps 1, 2, 4, and 5 in the extrapolation algorithm. Note that some portions of the 2nd derivative extend off the graph; also note that most of logdens is located underneath the asymptotically-corrected curve.

Part V

Conclusions and Future Work

Part VI

Codes

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, C. O.; Bordiga, S. *J. Phys. Chem. C* **2010**, *114*, 11185–11191.
- [12] McDaniel, J. G.; Schmidt, J. R. *J. Phys. Chem. C* **2012**, *116*, 14031–14039.
- [13] McDaniel, J. G.; Yu, K.; Schmidt, J. R. *J. Phys. Chem. C* **2012**, *116*, 1892–1903.
- [14] McDaniel, J. G.; Li, S.; Tylianakis, E.; Snurr, R. Q.; Schmidt, J. R. *J. Phys. Chem. C* **2015**, *119*, 3143–3152.
- [15] Lao, K. U.; Schaeffer, R.; Jansen, G.; Herbert, J. M. *J. Chem. Theory Comput.* **2015**, 150417132228001.

- [16] Pastorczak, E.; Corminboeuf, C. *J. Chem. Phys.* **2017**, *146*, 120901.
- [17] Żuchowski, P. *Chem. Phys. Lett.* **2008**, *450*, 203–209.
- [18] Horn, P. R.; Head-gordon, M. *Phys. Chem. Chem. Phys.* **2016**, *18*, 23067–23079.
- [19] Su, P.; Li, H. *J. Chem. Phys.* **2009**, *131*, 014102.
- [20] Chen, Y.; Li, H. *J. Phys. Chem. A* **2010**, *114*, 11719–24.
- [21] Su, P.; Jiang, Z.; Chen, Z.; Wu, W. *J. Phys. Chem. A* **2014**, *118*, 2531–42.
- [22] Fedorov, D. G.; Kitaura, K. **2006**,
- [23] McDaniel, J. G.; Schmidt, J. R. *J. Phys. Chem. A* **2013**, *117*, 2053–2066.
- [24] McDaniel, J. G. Development and Application of Physically-Motivated First-Principles Force Fields for Complex Chemical Systems. Ph.D. thesis, UW-Madison, 2014.
- [25] Yu, K.; Kiesling, K.; Schmidt, J. R. **2012**,
- [26] Verma, P.; Xu, X.; Truhlar, D. G. **2013**,
- [27] Valenzano, L.; Civalleri, B.; Sillar, K.; Sauer, J. **2011**, 21777–21784.
- [28] 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.
- [29] Yu, K.; McDaniel, J. G.; Schmidt, J. R. *J. Phys. Chem. B* **2011**, *115*, 10054–10063.
- [30] Jansen, G.; Scha, R. **2012**,
- [31] Guibas, L. Representing rotations with quaternions. 1992; graphics.stanford.edu/courses/cs164-09-spring/Handouts/handout12.pdf.
- [32] Shoemake, K. *Graph. Gems 3*; 1992; Chapter 6, pp 124–132.
- [33] Jeziorski, B.; Moszynski, R.; Szalewicz, K. *Chem. Rev.* **1994**, *94*, 1887–1930.

- [34] Szalewicz, K. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2012**, 2, 254–272.
- [35] Knizia, G.; Adler, T. B.; Werner, H.-J. *J. Chem. Phys.* **2009**, 130, 054104.
- [36] 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.
- [37] Stone, A. J. *J. Chem. Theory Comput.* **2005**, 1, 1128–1132.
- [38] Misquitta, A. J.; Stone, A. J.; Fazeli, F. J. *Chem. Theory Comput.* **2014**, 10, 5405–5418.
- [39] Misquitta, A. J.; Stone, A. J. *J. Chem. Phys.* **2006**, 124, 024111.
- [40] Van Vleet, M. J.; Misquitta, A. J.; Stone, A. J.; Schmidt, J. R. *J. Chem. Theory Comput.* **2016**, 12, 3851–3870.
- [41] Cardamone, S.; Hughes, T. J.; Popelier, P. L. a. *Phys. Chem. Chem. Phys.* **2014**, 16, 10367.
- [42] Kramer, C.; Spinn, A.; Liedl, K. R. *J. Chem. Theory Comput.* **2014**, 10, 4488–4496.
- [43] Dixon, R. W.; Kollman, P. a. *J. Comput. Chem.* **1997**, 18, 1632–1646.
- [44] Chaudret, R.; Gresh, N.; Cisneros, G. A.; Scemama, A.; Piquemal, J.-p. *Can. J. Chem.* **2013**, 91, 804–810.
- [45] Unke, O. T.; Devereux, M.; Meuwly, M. J. *Chem. Phys.* **2017**, 147, 161712.
- [46] Ferenczy, G. G.; Winn, P. J.; Reynolds, C. a. *J. Phys. Chem. A* **1997**, 101, 5446–5455.
- [47] Williams, G. J.; Stone, A. J. *J. Chem. Phys.* **2003**, 119, 4620–4628.
- [48] Misquitta, A. J.; Stone, A. J. *Mol. Phys.* **2008**, 106, 1631–1643.

ACRONYMS

Symbols | A | C | D | E | F | I | M | P | S

Symbols

 δ HF Delta-Hartree Fock. 16, 18

A

AIM atom-in-molecule. 45, 46, 51

C

CUS Coordinatively-Unsaturated. iv, 10, 11, 13–15, 26–28

D

DFT Density Functional Theory. 11, 13–15, 26, 39

DFT-SAPT Density Functional Theory Symmetry-Adapted Perturbation Theory.
11–13, 16, 17, 20, 27, 28, 36, 39, 40, 52

DMA Distributed Multipole Analysis. v, vii, 12, 18, 41, 42, 45–55

E

EDA Energy Decomposition Analysis. 14, 15, 26–28

F

FDDS frequency-dependent density susceptibility. 50

I

isa Iterated Stockholder Atoms. v, vii, 36, 41–43, 45, 47, 50–54, 59, *Glossary*: ISA

M

MOF Metal-Organic Framework. iv, 9–15, 26–28

P

PES potential energy surface. viii, 9, 12, 16, 17, 23, 24, 35–38, 75

POInter Parameter **O**ptimizer for **I**nter-molecular Force Fields. 34, 45, 52

S

SAPT Symmetry-Adapted Perturbation Theory. v, viii, 9, 11–16, 18, 26–28, 34–36, 39, 40, *Glossary*: SAPT

GLOSSARY

C | H | 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, 9, 13, 28, 36, 40, 41

CCSD(T)-f12 Explicitly-correlated CCSD(T). Given a sufficiently large (aVDZ or aVTZ) basis set, used throughout this work as a ‘gold-standard’ estimate of the exact potential energy surface. 12–14, 19, 20, 36, 40, 41

H

hetero-monomeric FILL . 35

homo-monomeric FILL . 35

I

induction FILL . 16, 18

ISA Iterated Stockholder Atoms, FILL . v, 41

L

LMO-EDA FILL . iv, viii, 9, 15–20, 23, 25–28

P

POInter FILL . 34

S

SAPT Symmetry-Adapted Perturbation Theory, a perturbative treatment of inter-molecular interactions which is pretty cool. v, 9, 34