MACHINE LEARNING TECHNIQUES FOR THE RATIONAL DESIGN OF ANALYTIC INTERATOMIC POTENTIALS

By

EUGENE RAGASA

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2019

I dedicate this to everyone that helped revamp this template. Aliquam molestie sed urna quis convallis. Aenean nibh eros, aliquam non eros in, tempus lacinia justo. In magna sapien, blandit a faucibus ac, scelerisque nec purus. Praesent fermentum felis nec massa interdum, vel dapibus mi luctus. Cras id fringilla mauris. Ut molestie eros mi, ut hendrerit nulla tempor et. Pellentesque tortor quam, mattis a scelerisque nec, euismod et odio. Mauris rhoncus metus sit amet risus mattis, eu mattis sem interdum.

ACKNOWLEDGMENTS

Thanks to all the help I have received in writing and learning about this tutorial. Acknowledgments are required and must be written in paragraph form. This mandates at least three sentences.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure page

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

MACHINE LEARNING TECHNIQUES FOR THE RATIONAL DESIGN OF ANALYTIC INTERATOMIC POTENTIALS

By

Eugene Ragasa

July 2019

Chair: Simon Phillpot
Major: Materials Science and Engineering

Abstracts should be less than 350 words. Any Greek letters or symbols not found on a standard computer keyboard will have to be spelled out in the electronic version so try to avoid them in the Abstract if possible. The best way to compile the document is to use the make_xelatex.bat file. If you are using Linux or Macintosh Operating Systems there are examples of make files for these systems in the Make Files Folder but they may be outdated and need to be modified for them to work properly. This document is the official tutorial outlining the use and implementation of the UF LaTeX2ϵ Template for use on thesis and dissertations. The tutorial will cover the basic files, commands, and syntax in order to properly implement the template. It should be made clear that this tutorial will not tell one how to use LaTeX2ϵ. It will be assumed that you will have had some previous knowledge or experience with LaTeX2ϵ, but, there are many aspects of publishing for the UF Graduate School that requires attention to some details that are normally not required in LaTeX2ϵ.

Pay particular attention to the section on references. NONE of the bibliography style files (.bst) are an assurance that your document's reference style will meet the Editorial Guidelines. You MUST get a .bst file that matches the style used by the journal you used as a guide for your references and citations. The files included in this document are examples only and are NOT to be used unless they match your sample article exactly!

You should have a .bib file (we have included several examples) that contains your reference sources. Place your .bib file in the bib folder and enter the name of the file in the list of bib files, or enter your reference information into one of our existing .bib files if you don't already have one. Just make sure to preserve the format of each kind of reference. Each time you cite a reference you enter the "key" (the first field in the reference listing in the .bib file) associated with that reference. During the compilation process LaTeX will gather all the references, insert the correct method of citation and list the references in the correct location in the proper format for the reference style selected.

CHAPTER 1
INTRODUCTION

As compututational power has increased so has the size and complexity of the simulations. The use of analytical empirical potentials has been part of computational materials science from the start, including problems with lattice dynamics, molecular dynamics, and other charges. The accuracy of a simulation depends on many factors, some involving the simulation method itself (e.g. numerical accuracy in solving sets of equations). Despite the promise of molecular dynamics, the difficulty in developing interatomic potentials leads to long developments when developing empirical potentials due to problems in determining an optimal parameterization. This work presents an emergent framework for the automated development of potentials based upon sampling from a distribution and evolution of that distribution so that the final distribution represents the set of parameterizations which can be described in a way as efficient. We present an novel framework for the automated development of potentials.

Our potential development approach takes a probabilistic approach, treating the vector of parameters as a random variable, where the variation in the values of the parameters represents the epistemic uncertainty associated with with the choice of parameter values.

However, the biggest errors in a simulation, as least with respect to how well it describes a real system, are the inadequacies of the models upon which the simulation is based.

1.1 Machine Learning Frameworks

In potential development, Despite promising work with the development of machine learning potentials, which promises to provide *ab initio* levels of fidelity, these approaches require large amounts of computational resources to development the requisite fitting datasets. Machine learning potential development is a data hungry approach as to ensure that the problem is not functionally underdetermine due to the large number of degrees of

freedom associated with the functional form. Moreover, neural networks models themselves are NP-hard problems which maps the current configuration of the neural net onto potential configuration space, requiring an extraordinary computational effort to determine the most effective parameter optimization. It should not be confused that neural networks are not an optimization solution, but actually an optimization problem, which are aided by different machine learning techniques which are dependent upon sampling

Instead, this work focuses on applying machine learning concepts to analytical interatomic potentials. These models have a functional form described by equations which attempt to capture the relevant physics associated with a material system. To adapt a functional form to a specific material system, a potential development process

1.2 Modernizing Analytical Potential Development

The modernization of analytical potential development needs to achieve the following goals: (1) clearly define the problem in its most general terms in what maybe, at times, a more rigorous mathematical exposition than that which is normally presented, (2) identify the problems with existing methodologies by bringing in terminology and notation from different fields to provide a more general framework for the problem of potential development, (3) provide a baseline implementation of this framework, which provides results which are analytical, transparent, and robust, and (4) provide an automation framework upon which to do the work.

1.3 Objectives and Outline

Chapter 2 provides an appropriate level of introduction to the major toolsets in computational materails science. Starting with quantum mechanical approaches to solutions to the potential energy surface, a function which maps the configuration space of atomic descriptors onto energies. Particular focus is placed on density functional theory (DFT), which is often used as the computational calculator to calculate difficult to observe experimentally. From here, we look to two tools where analytical potentials are most often used: molecular dynamic simulations and lattice dynamics simulations.

Chapter 3 describes which to is referred to the the traditional approach to potential development. Potential development is normally described as a quadratic programming optimization problem, applied to a description of a problem which it is not normally suited for. However, this problem has never really been discussed clearly and explains why more recently many potential developers have moved from potential development

In the third chapter, begins with a discussion of the empirical interatomic potential is described as approximation as the potential energy surface. The process of potential development starts with the selection of a fitting database, a finite set of material properties and the target values. The traditional approach to potential development uses a scalar optimization approach by minimizing an objective function which is referred to in literature as the cost function. The cost function is typically a weight sum of square differences between the predicted potential values and their respective target values. This objective function in minimized usually through either constrained or unconstrained optimization techniques dependent upon a quadratic programming algorithm dependent upon first-order derivatives for optimization. As a result, quadratic programming techniques are likely unsuitable, except when an initial guess to the target parameterization is already known. This problem is hinted at in discussions in literature in discussions in a move from local optimization vs global optimization algorithms. In this chapter, we show that this method for even identifying the global minimum due to issues with (1) regularization, (2) necessary conditions for a solution are likely not satisfied (Karush-Kuhn-Tucker conditions), and at the very least impossible to prove.

To generalize the potential optimization process, we introduce the concept of multi-objective optimization (MOO) which is concerned with mathematical optimization problems involving more than one objective function to be minimized simultaneously. Each material property is assigned a loss function, a measure of how good a prediction model does in terms of being able to predict an expected outcome. In MOO approach to potential development, the choice of an optimal model choice must be aken in the presence

of trade-offs between the choice of sacrificing the fidelity in predicting on material property at the expense of an another. We accept that the development of an potential largely involves a decision of tradeoffs determined by the potential developer, these expression of preferences is inherently subjective.

We introduce the concept of a loss function as the primary mechanism for measuring the fidelity between the predictions of an empirical potential and the target reference value. and Pareto optimality as the more general problem of potential development.

However, more damning is that even when these conditions are met, the optimal paramaterization is dependent upon a vector of weights, which uniquely determines the parameterization. That is, potential development is inherently a subjective process dependent upon the preferences of the user, which largely cannot be determine with certainty at the beginning of the process.

The ramification of chapter 3 is that potential optimization is NP-hard when preferences are fixed is that probing acceptable potentials by a mechanism of varying weights is computationally brutal, and reevaluates areas in parameter space which have already been solved.

Chapter 4 introduces a probability framework for representing the uncertainty associated with the parameterization. After introducing an appropriate level of probability theory and notation, we connect the measure theoretic approach to probability to continuous probability distributions, generation of random variables, non-parametric probability distributions. In addition, we review the Bayesian framework for parameter estimation, and adapt that framework to solve MOO problems.

Chapter 4 starts a framework for sampling in parameter space beginning with statistical concepts and links it to approaches in machine learning with genetic algorithms.

This work first starts with what is hoped an appropriate level of theory required to understand the necessities for automating the problem. In much of the literature, there is rather extensive discussion of techniques and applications of techniques, but

the supporting justification and rationalization of these techniques is necessarily short due to the format of publications. Even when discussing older approaches, appropriate review articles have been identified from different fields into order to provide insight to the problem of potential development.

In the development of a solid methodology and a description of potential development automation, it is necessary first describe the major computational tools used within the framework of machine learning. Specifically, the tools of Density Functional Theory, Molecular Dynamics, and Lattice Dynamics are briefly discussed to provide a sense of what is common and what is different between these computational approaches. This is covered in chapter 2.

Chapter 5 describes an evolutionary framework which combines the concepts from Chapter 3 and Chapter 4 as an optimization framework. A monte carlo approach is used for parallelization. Chapter 6 describes the implementation of the evolutionary framework in `pypospack`. Instead of developing a large monolithic application, which is difficult to extend, modify, and implement. `pypospak`Pypospack is conceived largely as software library which defines structure property relationships, how the structure property relationships are calculated, the process management of molecular dynamic simulations, parallel sampling, and potential selection. The next three chapters provides application of this framework for three different types of material systems. Chapter 7 goes through the development of a Buckingham style potential on a prototypical oxide, magnesium oxide. Chapter 8 goes through the development of a Stillinger-Weber potential for silicon. Finally, Chapter 9 provides results for the developmen of an embedded atom method (EAM) potential for Nickel.

Chapter 7,8,9 provide application of the framework for three different types of material systems. ather than being competitive, *ab initio* computational techniques are often integrated into potential development

This work takes a contrary, but complimentary approach by addressing the concerns readily brought force by this new avenue of study. The promise of machine learning potentials is provide *ab initio* level accuracy at a fraction of the cost of *ab initio* techniques.

It is the opinion of the author, that the development of analytical empirical potentials has largely been retarded by the lack of standard tools and analytical frameworks. Despite the ubiquity of molecular dynamics Instead of looking to develop the optimal parameterization which is expected to replicate a large range of values.

CHAPTER 2
THE MANY BODY PROBLEM AND ATOMISTIC METHODS

The properties of a system may be obtained by solving the quantum mechanical (QM) wave equation which governs the system dynamics. For non-relativistic system, this equation is the Schrodinger's equation. For all but the simplest systems, this approach in an impossible task in practice; the resulting many body problem has only been solved for a limited number of system. Within this chapter we outline the many body problem, it's intractibility before considering the Hohenberg-Kohn-Sham formulation of density functional theory (DFT), particularly in it's formulation it's application for systems with periodic boundary conditions. This reformulates quantuim mechanics, using electron density as the fundamental parameter to solve, rather than the many-electron wavefuction. This takes the N-body problem and recasts it into N single-body problems; which is a dramatic simplification.

We then approach higher order models which reduces computational intensity by looking at classical empirical potentials and their role in both molecular dynamics and lattice dynamics.

2.1 The Many-Body Problem

The Hamiltonian for a real material is defined by the presence of interacting nuclei and electrons:

$$H = \sum_i \frac{P_i}{2M_i} + \sum_\alpha \frac{p_\alpha}{2m} + \frac{1}{2}\sum_{ij} \frac{Z_i Z_j e^2}{r_{ij}} + \frac{1}{2}\sum_{\alpha\beta} \frac{e^2}{r_{\alpha\beta}} - \sum_{i\alpha} \frac{Z_i e^2}{r_{i\alpha}} \tag{2–1}$$

The first terms are kinetic energy terms, the latter terms are the nuclei-nuclei, electron-electron, and nuclei-electron interactions. Ideally, the solution of Schrödinger's equation, $H\Psi = E\Psi$ could be solved providing the total wavefunction $\Psi(\boldsymbol{r}_i, \boldsymbol{r}_{\boldsymbol{\alpha}})$. Except for the simplest of systems, this approach is impossible computationally. This later equation does not contain any electronic degrees of freedom, because all electronic effects are incorporated into $U(\boldsymbol{R}_i)$ which is the interatomic potential.

The kinetic energy is ignored since the heavy nulclei move more slowly than electrons. For the remaining interaction terms of the Hamiltonian, the nuclear positions are clamped at certain positions in space, the electron-nuclei interactions are not removed, since the electrons are still influenced by the Coulomb potential of the nuclei. This allows us to factor the wavefunction as

$$\Psi(\boldsymbol{R}_i, \boldsymbol{r}_\alpha) = \Xi(\boldsymbol{R}_i)\Phi(\boldsymbol{r}_\alpha; \boldsymbol{R}_i) \tag{2–2}$$

, where $\Xi(\boldsymbol{R}_i)$ describes the nuclei, and $\Phi(\boldsymbol{r}_\alpha; \boldsymbol{R}_i)$ describes the electrons parameterized by the clamped position of $\boldsymbol{R}_i$. In turn, the Hamiltonian is solve able as two Schrödinger's equations. The first equation contains the electronic degrees of freedom.

$$H_e\Phi(\boldsymbol{r}_\alpha; \boldsymbol{R}_i) = U(\boldsymbol{R}_i)\Phi(\boldsymbol{r}_\alpha; \boldsymbol{R}_i) \tag{2–3}$$

where

$$H_e = \sum_\alpha \frac{p_\alpha}{2m} + \frac{1}{2}\sum_{ij}\frac{Z_i Z_j e^2}{r_{ij}} + \frac{1}{2}\sum_{\alpha\beta}\frac{e^2}{r_{\alpha\beta}} - \sum_{i\alpha}\frac{Z_i e^2}{r_{i\alpha}} \tag{2–4}$$

Eqn. 2–3 gives the energy $U(\boldsymbol{R}_i)$ which depends on the clamped coodinates of $\boldsymbol{R}_i$. The electronic effects are contained in $U(\boldsymbol{R}_i)$ and is incorporated into the second equation which the motion of the nuclei

$$H_n\Xi(\boldsymbol{R}_i) = E\Xi(\boldsymbol{R}_i) \tag{2–5}$$

where

$$H_n = \sum_i \frac{P_i}{2m_i} + U(\boldsymbol{R_i}) \tag{2–6}$$

Direct solution of the Schrödinger equation for the electrons in a molecule is demanding because of the Coulomb repulsion between them.

2.1.1 Hartree and Hartree Fock Methods

The Hartree method[3–5] applies the variational principle to a product *ansatz* of orthogonal wave functions, known as the Hartree product, to represent the ground state function:

$$\Psi(\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N) = \psi_1(\boldsymbol{x}_1)\psi_2(\boldsymbol{x}_2)...\psi_n(\boldsymbol{x}_n) \tag{2–7}$$

where $\boldsymbol{x_i}$ is the set of space-spin coordinates, $\boldsymbol{x}_i = \{\boldsymbol{r}_i, \omega\}$ with $\omega \in \{\alpha, \beta\}$ being a spin-coordinate. However, since electrons are fermions they must follow the Pauli exclusion principle, and must be anti-symmetric under exchange of any of the space-spin coordinates. The Hartree product does not satisfy the anti-symmetry principle[6, 7], which can be demonstrated with a two particle system. For a two-particle system, the anti-symmetric property can be described as

$$\Psi(\boldsymbol{x}_1, \boldsymbol{x}_2) = -\Psi(\boldsymbol{x}_2, \boldsymbol{x}_1) \tag{2–8}$$

and Eq 2–7 clearly fails.

The HartreeFock method often assumes that the exact, N-body wave function of the system can be approximated by a single Slater determinant from a system of electrons. By invoking the variational method, one can derive a set of N-coipled equations for the N spin orbinals. A solution of these equations yields the Hatree-Fock wave function and energy of the system.

2.2 Density Functional Theory

Density Functional Theory is a simplification of the many body electron function by relating the wave function of the system of interest to the electron density of the system, where the firm theoretical underpinning are established by the Hohenberg-Kohn theorems[8].

The first Hohenberg-Kohn theorem demonstrates that the ground state properties of a many electron system are uniquely determined by an electron density that only depends on the three spatial coordinates. This establishes a one-to-one correspondence between between the electron potentials different densities. As a result, the electronic energy of a system as a functional of the density.

$$E_{el} = F_{HK}[n] + \int V_{ext}(\boldsymbol{r}) n(\boldsymbol{r}) d\boldsymbol{r} \tag{2–9}$$

This reduces the many-body problem of N electrons with $3N$ spartial coordinates to three spatial coordinates, through the use of functionals of the electron density.

The variational theorem for wavefunction

$$\int \Psi^{(1)*} \hat{H}_{el}^{(1)} \Psi^{(1)} d\boldsymbol{X} \leq \int \Psi^* \hat{H}_{el}^{(1)} \Psi d\boldsymbol{X} \tag{2–10}$$

When the variational theorem for the electon density, n, The second Hohenberg-Kohn theorem The second Hohenberg-Kohn theorem defines an energy functional and proves that the correct ground electron density also minimizes this energy functional. Since DFT is a ground state calculation.

2.2.1 The Exchange Correlation Term

In Kohn-Sham DFT, only the exchange-correlation energy, E_{XC}, as a functional of the electron spin densities $n(\boldsymbol{r})$ must be approximated. In the introduction to the Kohn-Sham equations[9], the local density approximation is introduced, The local density approximation is discussed by Kohn and Sham[9] in the introduction of the Kohn-Sham equations. In the LDA, the exchange energy per particle in each spatial point is taken as the exchange energy per particle from a uniform electron gas with a density equivalent to the density in this same point. [10]. Later LDA take similar approaches, [11–13].

$$E_{xc}^{LDA}[n] = \int n(\boldsymbol{r}) e_{xc}^{hom}(n(\boldsymbol{r})) d\tau \tag{2–11}$$

The generalized gradient approximation(GGA)[14, 15] introduces a gradient correction

$$E_{xc}^{GGA}[n] = \int n e_x^{LDA} F_{xc}^{GGA}(n, s)_{n=n(\boldsymbol{r})} d\tau, \tag{2–12}$$

where

$$s = \left. \frac{|\nabla n|}{2k_F n} \right|_{n=n(\boldsymbol{r})} \tag{2–13}$$

2.3 Density Functional Theory for Solids

2.3.1 Representation of an Infinite Solid

The potential energy, V is only dependent upon the relative internal coordinates of the system. For a system with N atoms indexed by i, there are $3N$ total coordinates for the system, $\boldsymbol{r_i} = (r_{i1}, r_{i2}, r_{i3})$ In the absence of an external field, an atomic system potential energy is invariant of translations and rotations in space.

2.3.2 Bloch's Theorem

2.3.3 Plane Wave Formulation

2.3.4 k-point sampling

2.4 Molecular Dynamics

Molecular dyanmics (MD) is a simulation approach where the time evolution of aset of interacting atoms is followed by numerically solving their equations of motion. In MD, the behavior of atoms follow Newtonian mechanics:

$$M\frac{d\boldsymbol{r}(t)}{dt} = F(\boldsymbol{r}(t)) = -\nabla V(\boldsymbol{r}(t)) \tag{2–14}$$

where t is time, $\boldsymbol{r}(t) = (\boldsymbol{r}_1(t), \boldsymbol{r}_2(t), ..., \boldsymbol{r}_N(t))$ represents the forces on the particles, and M is the mass matrix, which is a diagonal matrix with the mass, m_k, for $M_{k,k} = m_k$ for all diagonal entries. The total energy is conserved, even if the kinetic energy and potential energy can change dynamically. In Hamiltonian form, the Newtonian equation of motion can be written Hamiltonian form (Allen, Tildesley, et al 1989).

$$\frac{d\boldsymbol{r}}{dt} = \frac{\partial H\boldsymbol{r}, \boldsymbol{p}}{\partial \boldsymbol{p}}, \frac{d\boldsymbol{p}}{dt} = \frac{\partial H\boldsymbol{r}, \boldsymbol{p}}{\partial \boldsymbol{r}} \tag{2–15}$$

Therefore,

$$\frac{dH}{dt} = \frac{\partial H(\boldsymbol{r}, \boldsymbol{p})}{\partial \boldsymbol{r}}\frac{d\boldsymbol{r}}{dt} + \frac{\partial H(\boldsymbol{r}, \boldsymbol{p})}{\partial \boldsymbol{p}}\frac{d\boldsymbol{p}}{dt} = 0 \tag{2–16}$$

2.4.1 Numerical Integration

A dynamical simulation computes atomic positions as a function of time given their initial position $\boldsymbol{r}(t = 0)$ and velocities $\boldsymbol{v}(t = 0)$. Since Newton's equations of motion are 2nd order diffferential equations, an initial condition needs to specify both positions and velocities of all atoms at the initial condition. To solve the equation of motion computationally, we need to descretize time. Usually, time is descretized uniformly, $t_n = n\Delta t$, where Δt is referred to as the time step. The task of the simulation algorithm is to find $\boldsymbol{r}(t_n)$ for $i = 1, 2, 3...$ (Allen, Tildesley *et al* 1989).

The Verlet algorithm begins by approximating

$$\frac{d^2r(t)}{dt^2} = \frac{\boldsymbol{r}(t + \Delta t) - 2\boldsymbol{r}(t) + \boldsymbol{r}(t - \Delta t)}{\Delta t^2} \tag{2–17}$$

Thus,

$$\frac{\boldsymbol{r}(t + \Delta t) - 2\boldsymbol{r}(t) + \boldsymbol{r}(t - \Delta t)}{\Delta t^2} = -\frac{1}{m}\frac{dU(\boldsymbol{r}(t))}{d\boldsymbol{r}} \tag{2–18}$$

$$\boldsymbol{r}(t + \Delta t) = 2\boldsymbol{r}(t) - \boldsymbol{r}(t - \Delta t) - \Delta t\frac{1}{m}\frac{dU(\boldsymbol{r}(t))}{d\boldsymbol{r}} \tag{2–19}$$

2.5 Empirical Interatomic Potentials

The interatomic potential $U(\boldsymbol{R}_i)$ derived from the Born-Oppenheimer approximation is derived from a quantum-mechanical perspective. The computational cost of *ab initio* such as density-functional theory (DFT) can provide accurate structural energies and forces, but their computational cost limits approaches to compute $U(\boldsymbol{R}_i)$ makes the scientific inquiry of systems requiring longer simulation times or larger number of atoms to captures relevant feature sizes unreasonable. An empirical interatomic potential $\hat{V}(\boldsymbol{R}_i; \boldsymbol{\theta})$ is an analytical function parameterized by $\boldsymbol{\theta} = (\theta_1, ..., \theta_n)$ which is meant to approximate $U(\boldsymbol{R}_i)$. The total energy of a potential of N atoms with an interaction described by the empirical potential, V, can be expanded in a many body expansion.

$$V(\boldsymbol{r}_1, ..., \boldsymbol{r}_N) = \sum_i V_1(\boldsymbol{r}_i) + \sum_i \sum_{i<j} V_2(\boldsymbol{r}_i, \boldsymbol{r}_j) + \sum_i \sum_{i<j} \sum_{j<k} V_3(\boldsymbol{r}_i, \boldsymbol{r}_j, \boldsymbol{r}_k) + ... \tag{2–20}$$

The first term V_1 is the one body term, due to an external field or boundary conditions, which is typically ignored in classical potentials. The second term V_2 is the pair potential, the interaction of the term is dependent upon the distance between $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$. The three-body term potential V_3 arises when the interaction interaction of a pair of atoms is modified by the presence of a third. Based upon this expansion, we can classify certain potentials into two classes: pair potentials when only V_2 is present and many-body potentials when V_3 and higher order terms are included.

Over the last few decades, a large number of potentials have been developed to descibe various bonding types and environments. To take representative examples, the Lennard Jones was developed for the van der Waals interactions of noble gases, pair potentials such as the Buckingham potential can be used for ionic solids, the embedded atom model (EAM) is developed for metallic systems, the Assisted Model Building with Energy Refinement (AMBER) for biomolecules, the tersoff potential for covalently bonded materials. To deal with bonding and chemical environments for heterogenous materials like metal/metal oxide interfaces have led to extensions such as MEAM, REBO, COMB, and ReaxFF.

More recently, potentials such as GAP and SNAP represent the atomic environment of an atom not from a collection of a vectors of atomic positions which feed into formulaic functional forms, but to calculate the bispectrum of the neighborhood of atoms. The bispectrum combined with an orthogonal expansion of components is dependent upon large amounts of density functional images to use in the fitting dataset to produce DFT fidelity reproductions of interatomic forces on an atom. This chapter reviews typical approaches to fitting empirical potentials.

2.5.1 Thermodynamic Ensembles

The statistical ensembles for molecular dynamics simulations can be defined by looking at the ideal gas law:

$$PV = Nk_BT \tag{2–21}$$

The left hand of the equation is the potential energy, E_P, of the system, which is balance by right hand of the equation, which is the kinetic energy, E_k, of the system. i Pressure (P), volume (V), and temperature (T), and the number of atoms (N) are the state variables of the system.

To relate the ideal gas law equation to molecular dynamics, we first start with the simplest state variable, volume, which is the volume of the simulation cell. The other state variables are fairly easy to determine from quantites which can be calculated from MD simulation.

In molecular dynamics, through Verlet integration we directly calculate the position, $\boldsymbol{r}_i$ and velocity, $\boldsymbol{v}_i$, of each atom. The kinetic energy of an atom is

$$E_k = \frac{1}{1} m|\boldsymbol{v}|^2 \tag{2–22}$$

In the first of ensemble, the NVE system constant number (N), volume (V), and energy (E); the sum of kinetic (KE) and potential energy (PE) is conserved, T and P are unregulated

NVT: constant number (N), volume (V), and temperature (T); T is regulated via a thermostat, which typically adds a degree of freedom to the conserved Hamiltonian; for the CPT module in CHARMM, this is a piston whose KE and PE are included in the Hamiltonian; P is unregulated

NPT: as for NVT, but pressure (P) is regulated; again, for the CPT module this is one or more pistons whose KE and PE are added to the Hamiltonian

2.6 Lattice Dynamics

2.7 Calculation of Material Properties

2.7.1 Structural Properties

2.7.1.1 Minimization Techniques

Greatest Descent Conjugate Gradient

2.7.2 Phase Order Properties

2.7.3 Point Defect Formation Energies

2.7.4 Surface Energies

2.7.5 Stacking Fault Energies

2.8 Notation

2.8.1 Simulation Cell

A simulation cell is defined by the lattice basis and the atomic basis. The lattice vectors which describes the periodic boundary conditions three lattice vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$Euclidean space which forms the basis for the crystallographic system when periodic boundary conditions are applied. The translational properties of a crystal allows the simulation of an infinite bulk material from a fixed volume. In traditional crystallography, the boundaries of the unit cell were defined as a, b, c corresponding to the length of each lattice vector and the angles α, β, γ. In computatonal materials, a more convenient representation

CHAPTER 3
POTENTIAL DEVELOPMENT AND PARETO OPTIMALITY

In the previous chapter, an overview of the different computational tools associated to atomistic simulation were presented to give some idea of interconnectivity and breath of atomistic simulations. Within this chapter, we outline the Pareto approach to potential optimization within the context of more typical current appraoches to potential development.

The structure of this chapter starts off by framing potential development the broadest mathematical framework possible since the purpose of this dissertation is to identify new techniques and directions for potential development which have been enabled by a broad set of computational tools which harnesses large amounts of computational power to solve analytically intractable problems in a field broadly known as machine learning.

This chapter consists of three sections. In the first section, we introduce the idea of a potential energy surface and how interatomic potentials can be thought of as computational inexpensive surrogate models, and we introduce the terminology of potential development as well as some broad concepts of potential optimization. Next, we build up the notation and terminology of general optimization from a broad mathematical standpoint, first covering the more familiar single-objective optimization then multiobjective optimization. In the third section, we combine the discussion of the first and second section, to cast the problem of potential development as a multi-objective optimization utilizing the concepts and notation common to most.

In particular, we discuss the the current techniques to potential optimization which are applied to potential optimzation as well as some of the mathematical problems and issues of certain approach, and then clearly demonstrate that multi-objective optimization is dependent upon the concept of Pareto optimality.

3.1 Potential Energy Surfaces

The potential energy surface is the energy of a collection of atoms as a function of the positions of its nuclei, $\{\boldsymbol{R}\}$. Using the analogy of a landscape, the potential energy surface (PES) represents a mapping of the positions of the atoms of a material system to an energies. This creates an energy landscape which allows materials systems to viewed from a topological perspective, which allows the PES to describe the evolution of a system.

From a mathematical representation, the PES is a a function, $V : \boldsymbol{R} \rightarrow E$, which maps a high dimensional interatomic configuration, $\boldsymbol{R}$, onto the set of real numbers representing energies, $E \in \mathbb{R}$. Given an atomic arrangement $\boldsymbol{R}$, the evaluated potential surface $V(\boldsymbol{R})$ gives the height of the energy landscape for any atomic configuration, providing an approximation of the potential energy surface, so that the concept of a potential energy surface arises. To study the evolution of a system, such as kinetic properties and chemical reactions, it is necessary to calculate the energy for every atomic arrangement of interest.

To describe empirical potentials, we first start with a a mathematical description of configuration space both from a crystallographic perspective, but how this crystallgraphic perspective can be translated into other representation common with empirical potentials.

3.1.1 Configuration Space

In solid materials, atoms are typically represented as infinite crystalline solids, with th atomic positions placed within a representative unit volume. This representative unit is referred to as a unit cell, which defines that boundaries, the volume, the lattice positions of each atom.

The boundaries of the unit cell are defined are defined by three lattice vectors, defined in three dimensional Euclidean space, $\mathbb{R}^3$. The three lattice vectors, $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, and $\boldsymbol{a}_3$, with $\boldsymbol{a_i} \in \mathbb{R}^3$, which defines an alternative coordinate sytem in Euclidean space in which to describe a lattice. The triplet of lattice vectors, $(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3)$, describes an alternative coordinate system in Euclidean space in which to describe a lattice. Throughout this

work, the collection of lattice vectors is also referred to as a unit cell or simulation cell depending upon context.

Each atom i is identified by a chemical species, and its atomic position, $\boldsymbol{r}$. If the Cartesian unit vectors,$[\hat{\boldsymbol{\imath}}, \hat{\boldsymbol{\jmath}}, \hat{\boldsymbol{k}}]$, then the atomic positions can be represented as the ordered triplet, (r_x, r_y, r_z). More commonly, atomic positions are represented in the coordinates system define by the lattice vectors, (r_1, r_2, r_3) and $[\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3]$, respectively. The relationship between the two coordinates system being represented by the equation

$$r_x\hat{\boldsymbol{\imath}} + r_y\hat{\boldsymbol{\jmath}} + r_z\hat{\boldsymbol{k}} = r_1\boldsymbol{a}_1 + r_2\boldsymbol{a}_2 + r_3\boldsymbol{a}_3 \tag{3–1}$$

The boundaries of the unit cell also describes the periodic boundary conditions, since they also reflect that translational symmetry of the crystalline system. Each lattice vector can be represented as a translational operator, T_i for $i \in \{1, 2, 3\}$, such that, $T_i(\boldsymbol{r}) = \boldsymbol{r} + n_i\boldsymbol{a}_i = \boldsymbol{r}$, and collectively,

$$T(\boldsymbol{r}) = \boldsymbol{r} + n_1\boldsymbol{a}_1 + n_2\boldsymbol{a}_2 + n_3\boldsymbol{a}_3 = \boldsymbol{r}, \forall n_i \in \mathbb{Z} \tag{3–2}$$

The geometry of a molecule can be described by the collection of the positions of N atoms $\boldsymbol{R} = \{\boldsymbol{r}_1, ..., \boldsymbol{r}_N\}$, where the $\boldsymbol{r}_i$ could be the set of Cartesian coordinates of the atoms. These collections of atoms can be transformed to reflect the distances between the two atoms i and j, $\boldsymbol{r}_{ij} = \boldsymbol{r}_i - \boldsymbol{r}_j$ as used in pair potentials, the angles between atoms $\boldsymbol{r}_{ijk}$ as in three body potentials.

3.1.2 Empirical Interatomic Potentials

The goal of developing a interatomic potential, $\hat{V}$ is identify a computationally efficient surrogate model, which models the potential energy surface, V. The use of a hat over a variable indicates that the quantity is an approximation of the actual value; this notation is used to identify approximating quantities throughout this work. Since energy is a scalar value, V and $\hat{V}$ are functions in the same measure space that assigns energies to atomic configuration (e.g. $V : \{\boldsymbol{R}\} \to \mathbb{R}$ and $\hat{V} : \{\boldsymbol{R}\} \to \mathbb{R}$, the approximating relationship

by the addition of a difference term ϵ

$$V(\boldsymbol{R}) = \hat{V}(\boldsymbol{R}) + \epsilon(\boldsymbol{R}) \tag{3–3}$$

$$\epsilon(\boldsymbol{R}) = V(\boldsymbol{R}) - \hat{V}(\boldsymbol{R}). \tag{3–4}$$

Interatomic potentials are often expressed as a series expansion of functional terms which explain the relevant physics of a system. In these cases, the total energy of the system is sum of the individual contribution of each atom i. For a system with N atoms,

$$\hat{V} = \sum_i \hat{V}(\boldsymbol{r}_i|\boldsymbol{R}) \tag{3–5}$$

The total energy of a potential of N atoms with an interaction described by the empirical potential, V, can be expanded in a many body expansion as described in LeSar[16]

$$V(\boldsymbol{r}_1, ..., \boldsymbol{r}_N) = \sum_i V_1(\boldsymbol{r}_i) + \sum_i \sum_{i<j} V_2(\boldsymbol{r}_i, \boldsymbol{r}_j) + \sum_i \sum_{i<j} \sum_{j<k} V_3(\boldsymbol{r}_i, \boldsymbol{r}_j, \boldsymbol{r}_k) + ... \tag{3–6}$$

The first term V_1 is the one body term, due to an external field or boundary conditions, which is typically ignored in classical potentials. The second term V_2 is the pair potential, the interaction of the term is dependent upon the distance between $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$. The three-body term potential V_3 arises when the interaction interaction of a pair of atoms is modified by the presence of a third. Based upon this expansion, we can classify certain potentials into two classes: pair potentials when only V_2 is present and many-body potentials when V_3 and higher order terms are included.

An empirical interatomic potential $\hat{V}(\boldsymbol{R}_i; \boldsymbol{\theta})$ is an analytical function parameterized by $\boldsymbol{\theta} = (\theta_1, ..., \theta_n)$ which is meant to approximate $U(\boldsymbol{R}_i)$.

$$\hat{V}(\boldsymbol{R}) = \sum_i \hat{V}_1 + \sum_{i<j} V_2 + \sum_{i<j<k} V_3 \tag{3–7}$$

represents the total energy of the system, where $\hat{V}(\boldsymbol{r}_i|\boldsymbol{R})$ which is the contribution of the ith atom of a system An analytical potential is represented by a set of equation known as

a formalism, which decomposes the energy of a structure as the individual contributions of each atom i in its local environment. The simplest of these potentials is the Lennard-Jones (LJ) potential[17], that approximates the interaction between a pair of neutral atoms, such as a noble gas. A common expression the LJ potential is

$$V_{LJ} = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^{6} \right] \tag{3–8}$$

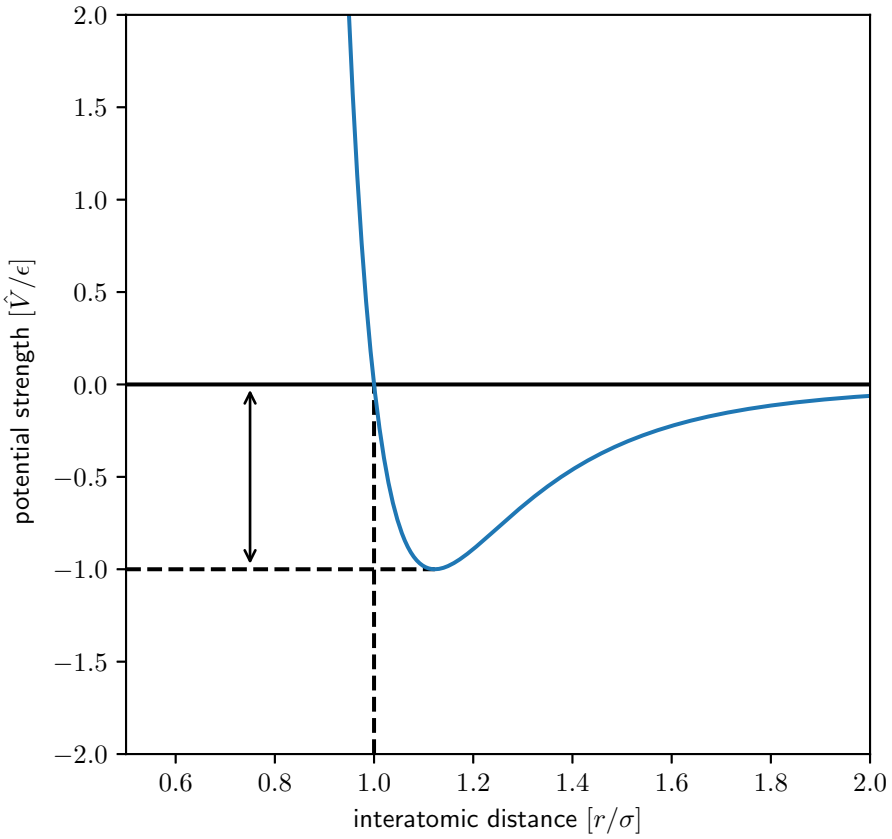

Figure 3-1. EPS format diagram. Note: no filetype is designated by adding an extension. The file type is determined and the correct procedure is automatically chosen by xelatex.

For an analytical potential, $\hat{V}$ is parameterized with a vector of P parameters, $\boldsymbol{\theta} = [\theta_1, ..., \theta_P] \in \mathbb{P}]$

3.2 Traditional Approaches to Potential Deveopment

With an empirical potential, $\hat{V}$, is parameterized by a vector of N values, $\boldsymbol{\theta} = \{\theta_1, ..., \theta_N\}$. In the forward use of a potential, the parameters are considered to be fixed, $\hat{V}(\boldsymbol{R} : \boldsymbol{\theta})$, where the parameterization $\boldsymbol{\theta}$ is treated as fixed, and the function is considered vary with respect to the configuration of atoms, $\boldsymbol{R}$. However, potential development is an inverse problem where the potential developer is looking to determine the best parameterization, $\boldsymbol{\theta}^*$, based upon a fixed set of atomic configurations, as in the force-matching method of Ercolessi and Adams [18] , or are probed indirectly through the calculation of specific material properties. As a result, the empirical potential treated as varying due to changing the parameterization, $\boldsymbol{\theta}$, while keeping the set of atomic configurations fixed $\boldsymbol{R}$. For the application of the potential development, we can rewrite Equation3–4, where the error function is a function of the parameterization, while keeping the set of interatomic configurations fixed.

$$\epsilon(\boldsymbol{\theta} : \boldsymbol{R}) = V(\boldsymbol{R}) - \hat{V}(\boldsymbol{\theta} : \boldsymbol{R}) \tag{3–9}$$

3.2.1 Fitting Database

A more popular approach in potential developmet comes from fitting an interatomic potential so that predicted values of material properties predicted by the interatomic potential corresponds with either high-fidelity predictions done through more computationally expensive *ab initio* calculations or directly through experimental observations. The material properties calculated are referred to as a quantity of interest (QOI), q, while the predictions made by the potential denoted, $\hat{q}$. A quantity of interest can be related to the PES, by considering aa high-fidelity method used to calculate the potential of iteresst. In the case when the target QOI values are calculated from a high-fidelity method, such as density functional theory (DFT), the computation of the material property can be decomposed as a function of energy evaluation of the PES,

$$q(\boldsymbol{R}_1, ..., \boldsymbol{R_N}) = F(V(\boldsymbol{R}_1), ..., V(\boldsymbol{R}_N)) \tag{3–10}$$

Since $\hat{q}$ is ultimately detemined from energy calculations upon atomic structures, either directly or though numerical estimation, we can represent the energy difference between the predicted QOIs and the target QOIs.

$$\epsilon(\boldsymbol{\theta} : \{\boldsymbol{R}_1, ..., \boldsymbol{R}_N\}) = \hat{q}(\boldsymbol{\theta} : \{\boldsymbol{R}_1, ..., \boldsymbol{R}_N\}) - q \tag{3–11}$$

A fitting database is a collection of structure property functions q_i with an associated atomic configurations, also referred to as structures. The set of all possible atomic configuration is referred to as the configuration space. The goal of a fitting database to find to find a representative set of structures in which to calculate the structure property relationships q_i.

Lattice constant, bulk modulus, vacancy formation energy, or anything that can be defined from energy structures. In the fitting database, the structure proerty functions evaluated using an empirical potentials and compared to target reference values, with values either determined from experimental values or a a high-fidelity structure such as DFT. The collection of structure property relationships, is denoted $\boldsymbol{q} = (q_1, q_2, ...q_N)$ for N structure property relationships. Usually accuracy and transferribility are tested against an external database.

In literature, the developers of potentials tend to use 0 K properties. A more important reason why potentials are fit to 0 K properties, is that it alows the incorporation of first-principles data. The most ubiquitous *ab initio* techniques are calculations using density functional theory (DFT). DFT allows the calculation of structural properties which are experimentally difficult to access, as well as provide energic information from kinetically unstable structures. The incorporation of first-principles data in the fitting database significantly improves the reliability of semi-empirical potentials by sampling a larger area of configuration space[21-28]. This is covered in detail in a review article by Payne *et al* [19].

From a computational standpoint, at 0 K the calculation of material properties become precise because atomic motion stops, and only a single evaluation of a parameterization needs to be evaluated against the reference value. When the $T > 0$, issues with sampling arise. In the long time limit, the sampled trajectory yields detailed information about the Hamiltonian. Shorter trajectories yield incomplete information and confound comparison of parameters with experimental values.

When many $\hat{\boldsymbol{q}}(\theta)$ has to be evaluated many times, fitting to structure property relationships which are dependent upon themodynamic ensembles for $T > 0$ becomes quickly computational infeasible.

3.2.2 Cost Function

When the fitting database is defined, potential development then proceeds by the determination of the optimal database. With the description of the fitting database achieved, our discussion turns on how this potential database can be used to obtain an optimizal parameterization, $\boldsymbol{\theta}^N$. The goal of potential development is to achieve a high-level of fidelity between the approximating potential model, $\hat{V}$ and the potential model V, with the ultimate goal of reducing the error function $\epsilon(\{\boldsymbol{R}\}) = V(\{\boldsymbol{R}\}) - \hat{V}(\{\boldsymbol{R}\}) \rightarrow 0$. When applied to predictions of a QOI database, then for a high-fidelity equation $\epsilon_i \rightarrow 0$.

For the purposes of numerical optimization, it is convenient to define a scalar valued function, which will determine the optimal parameterization, $\boldsymbol{\theta}^*$, when this scalar function is minimized. Typically, the cost function is defined as the weighted sum of squared differences between the qoi predicted value, $\hat{q}$, and the qoi target value $\hat{q}$. A typical approach is the weighted least squares approach, in this approach the squared difference between predicted qois, $\hat{q}$ and its target value, q is coupledeach of the qois, $\hat{q}_i$.

$$C(\boldsymbol{\theta}) = \sum_{i=1}^{N} w_i \left(q_i - \hat{q}_i(\boldsymbol{\theta})\right)^2 = \sum_{i_1}^{N} \epsilon_i(\theta) \tag{3–12}$$

The impact of weights

3.2.3 single-objective optimization

We can contrast this to single objective function optimization. If $\boldsymbol{F}(\boldsymbol{x}) = F(\boldsymbol{x})$, then this problem becomes a scalar optimization problem. The Karush-Kuhn-Tucker conditions [20, 21] are necessary to solve non-linear optimzation problems. Often this is stated as the condition that $F\boldsymbol{x})$ is convex with respect to the convex set $\boldsymbol{x}$, that is that the inequality constraints, $g_i(\boldsymbol{x}$. If $F(\boldsymbol{x})$ is convex with respect to the domain of $\boldsymbol{x}$, the solution can be solved with elementary multivariate calculus methods.

3.2.4 Convex Optimization

Numerical algorithms make heavy use of scalarization results, and most papers in the field of MOO and economics deal with non-linear programming problems, corresponding duality theorems, and the repeated application of the simplex method.

However, within the literature of potential development approaches focus upon local minimization techniques and global optimization techniques.

objective function is concave. constraint set is convex. KKT requirements for uniqueness.

3.2.5 Global Approaches

The task of global optimization is to find the best parameterization, $\boldsymbol{\theta} = (\theta_1, ..., \theta_n)^T \in \Theta$ according to a set of objective functions, which we take here to be a set of loss functions,$\boldsymbol{L} = \{L_1, ...L_m\}$. When $m = 1$ the problem is a single objective optimization problem and the goal is to minimize a single loss function f, i.e.,

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} L(\boldsymbol{\theta}. \tag{3–13}$$

When $m > 1$, the problem becomes a multiple-objective optimization problem(MOO). In this case, no global optimum exists since the objective function We look at two approaches, the first is a scalar optimization approach which combines the set of loss functions into a single objective function, which converts the problem into scalar optimization. The second is to look at the problem as a multi-objective optimization

problem, For $m = 1$, the problem is a single objective optimization problem, and the goal Genetic algorithms are a popular meta-heuristic that is particularly well-suited for this class of problems. Traditional GA are customized to accomodate multi-objective problems by using specialized fitness functions and introducing methods to promote solution diversity.

The second general approach is to determine an entire Pareto optimal solution set or a representative subset. A Pareto optimal set is a set of solutions that are nondominated with respect to each other. While moving from one Pareto solution to another, there is always a certain amount of sacrifice in one objective(s) to achieve a certain amount of gain in the other(s). Pareto optimal solution sets are often preferred to single solutions because they can be practical when considering real-life problems since the final solution of the decision-maker is always a trade-off. Pareto optimal sets can be of varied sizes, but the size of the Pareto set usually increases with the increase in the number of objectives.

The ultimate goal of a multi-objective optimization algorithm is to identify solution in the Pareto optimal set. However, identifiying the entire Pareto optimal set, for multi-objective problems, is impossible to its size. Proof of solution optimality is computationally infeasible. Therefore, a practical approach is achieve successively better approximations of the Pareto surface that represent the Pareto set as well as possible.

A multi-objective optimization approach should achieve the following conflicting goals as described by Zitzler *et al*[22]: (1) the best known Pareto front should be as close as possible to the true Pareto front. Ideally, the best-known Pareto set should be a subset of the Pareto set, (2) solutions in the best known Pareto set should be uniformly distributed and diverse over the Pareto front in order to provide the decision-maker a true picture of trade-offs, and (3) the best-known Pareto front should capture the whole spectrum of the Pareto front at the extreme ends of the spectrum. While the first two goals are important for multi-objective optimization, the last goal is erroneous. In general, when developing potentials, the DM is interested in compromise solutions and a parameterization with

high fidelity with respect to one material property at the expense of a loss of fidelity with respect to all other prediction would be a pathological parameterization.

3.2.5.1 Genetic Algorithms

The method which will be proposed in chapter 5 is not a genetic algorithm, but has many similarities as Genetic Algorithms but tailored to create an ensemble of Pareto optimal parameters. However, it is a genetic solution and the iterative approach of generating new populations is akin to previous solutions. As a result, the section of review in this section is necessarily incomplete but refer to an introductory review by Konak *et al*[23] as well as the book by Deb[24]

3.3 Multi-objective optimization

Parameter estimation can be stated as a MOO problem. Many decision and plannning problems involve multiple conflicting criteria which must be considered simultaneously. In the field of optimization, problems which have multiple criteria are deferred to as multiple critieria decision making problems (MCDM) and the algorithms used to solve them as multiple-objective optimization (MOO).

3.3.1 Loss Functions

At this point we turn to generalizing the concept of the cost function described in **??**, The cost function is a piece construction of $w_i\epsilon_i^2(\boldsymbol{\theta}$. What follows is a discussion of single objective optimization within the context of potential development before discussing multiple objective optimization before discussing multiple objective optimization we will first discuss single-objective optimization to introduce the terminology and notation used throughout the rest of this book. We To introduce the terminology used within this work, it is instructive to dispose of implementation specifics of algorithms and numerical estimation techniques, and think of an optimization problem in terms of sets. Here A is typically subset of Euclidean space $\boldsymbol{R}^n$, but could be mathematically formulated to include non-quantitative data or functionals. Then the goal of single objective optimization is to select the element $\boldsymbol{x_0} \in A$, such that $F(\boldsymbol{x_0} \leq F(\boldsymbol{x}$ for all $\boldsymbol{x} \in A$.

The problem of potential development can be cast as an optimization problem. L We start by casting the problem of potential development by adopting the The general multi-objective optimization (MOO) Using the notation of Marley and Arora[25], the general multi-objective optimization problem (MOO) is expressed mathematically as

$$\underset{\boldsymbol{\theta}\in\Theta}{\text{minimize}} \qquad \boldsymbol{L}(\boldsymbol{\theta}) = [L_1(\boldsymbol{\theta}), L_2(\boldsymbol{\theta}), ..., L_N(\boldsymbol{\theta})]^T \tag{3–14}$$

$$\text{subject to} \qquad g_j(\boldsymbol{x}) \leq 0, j = 1, 2, ..., m \tag{3–15}$$

$$h_k(\boldsymbol{x}) = 0, l =, 1, 2, ..., n \tag{3–16}$$

$$\boldsymbol{x} \in \boldsymbol{X} \tag{3–17}$$

where k is the number of objective functions, m is the number of inequality constraints, and e is the number of equality constraints. The vector $\boldsymbol{x} \in \boldsymbol{X} \subseteq \mathbb{R}^n$ is a vector design variables x_i, and X is feasible design space. $\boldsymbol{F}(\boldsymbol{x}) \in \mathbb{R}^k$ are called objectives, cost functions, or criteria. The feasible critereon space Z is defined as $\{\boldsymbol{F}(\boldsymbol{x})|\boldsymbol{x} \in \boldsymbol{X}\}$.

The objective function, $F_1(\boldsymbol{x}) : \mathbb{R}^\kappa \to \mathbb{R}$, For MOOs, the objectives are generally conflicting, preventing simultaneous optimization.

3.3.2 Cost function method

3.3.3 Pareto optimality

If all functions are for minimization, a feasible solution $\boldsymbol{x}_1$ is said to dominate another feasible solution $\boldsymbol{x}_2$, denoted $\boldsymbol{F}(\boldsymbol{x}_1 \succ \boldsymbol{x}_2)$, if and only if $F_i(\boldsymbol{x}_1) \leq F_i(\boldsymbol{x}_2)$ for $i = 1, ..., k$ and $F_j(\boldsymbol{x}_1) < F_j(\boldsymbol{x}_2)$ for at least one objective function j. A solution is said to be Pareto optimal if it is not dominated by any other solution in the solution space.

A Pareto optimal solution cannot be improved with respect to any objective without worsening at least one objective function. The set of all feasible non-dominated solutions in X is referred to as the Pareto optimal set, and for a given Pareto optimal set, the corresponding values in the objective space are called the Pareto Front.

While this is occasionally stated in potential development literature, it is often within the context of the use of global optimization techniques. The purpose of this

section is provide a clear methodological approach to determining the optimal parameters within the context of MOO, and elucidate the problems often encountered in potential development specifically to the choice of optimization techniques often employed in potential development.

Let $V(r_{ij}, \boldsymbol{\theta})$ be an analytical potential, dependent upon the distance, r_{ij}, between atoms i and j; the parameters of the potential are defined by the array $\boldsymbol{\theta} = (\theta_1, ..., \theta_P)$ for P parameters. Then to calculate material properties, the potential is combined with the structures and the necessary simulation conditions, such as temperature, pressure, and volume. Since there is a difference between the predicted material properties which a potential predicts and the actual material properties, it is necessary to introduce notation to differentiate the two. The predicted material properties will be denoted by $\hat{\boldsymbol{q}} = (\hat{q}_1, ..., \hat{q}_M)$, while the actual material property will be denoted by $\boldsymbol{q} = (q_1, ..., q_M)$ for M structural properties. The notation of q comes from verification, validation, and uncertainty quantification literature where the term quantity of interest (QOI) is used.

Then for the purposes potential development, a potential can than be viewed as a function $V : \boldsymbol{\Theta} \rightarrow \hat{\boldsymbol{Q}}$ where the parameters $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ maps to $\hat{\boldsymbol{q}} \in \hat{\boldsymbol{Q}}$. Since $\hat{\boldsymbol{q}}$ is a function of of the potential V, then we denote this relationship $\hat{\boldsymbol{q}}(\boldsymbol{\theta})$ and $\hat{\boldsymbol{Q}}(\boldsymbol{\theta})$.

3.3.4 Prediction Error function

In order to assess the prediction errors of the structure property functions, we denote the $\hat{\boldsymbol{q}}(\boldsymbol{\theta}) = (\hat{q}_1(\boldsymbol{\theta}), \hat{q}_2(\boldsymbol{\theta}), ..., \hat{q}_N(\boldsymbol{\theta}))$ as the predicted material properties

The difference between the prediction values and target values of the QOIs produces a vector of error functions, $\boldsymbol{\epsilon}(\boldsymbol{\theta}) = (\hat{q}_1(\boldsymbol{\theta}) - q_1, \hat{q}_2(\boldsymbol{\theta}) - q_2, ..., \hat{q}_N(\boldsymbol{\theta}) - q_N)$,

3.3.5 Parameters

Let V be an empirical potential parameterized by P number of parameters $\boldsymbol{\theta} = [\theta_1, \theta_2, ...\theta_P]$.

3.3.6 Constraints on parameters

3.3.7 Structure Property Relationships

3.3.8 Constraints on structure property relationships

3.3.9 Parameter Optimization Problem as MOO formulation

3.4 Pareto Front

In multiobjective optimization problems, it is characteristic that no unique solution exists, but a set of mathematically equally good solutions can be identified. These solutions are known as nondominated, efficient, noninferior or Pareto optimal solutions. In MCDM literature, these terms are synomous.

In MCDM literature, the idea of solving a multiobjective optimization problem is understood as helping a human decision maker (DM) in understanding the multiple objectives simultaneously and finding a Pareto optimal solution. Thus, the solution process requires some interaction with the DM in the form of specifying preference information and the final solution is determined by these preferences.

In potential development, the preferences of potential developer likewise influences are particular parameterization, which has results in the development of empirical potentials as somewhat of a black art. In the end, empirical potentials are simplified models which predict structure property relationships.

In classical potential optimization, the identification of an optimal parameterization is determined by the minimization of a cost function which couples multiple objective functions, usually a weighted sum of squares, and different weights are used in an interactive fashion until an acceptable parameterization is determined.

3.5 Surveys of Methods

Chankong and Haimes 1983 Hwang and Masud 1979 Marler and Arora 2004 Miettinen 1999 Sawaragi et al 1985 Steuer 1987 Vincke 1992

We start our review of methods using Hwang and Masud 1979 and Miettinen 199, to classify the different classes of approaches by methological approach rather than technical techniques.

3.5.1 no preference methods

The task is to find some neutral compromise solution without any additional information. This means instead of asking the DM for preference information, some assumption are made about what a reasonable compromise could be like.

3.5.2 *a priori* methods

In *a priori*, the DM first articulates preference information and the solution tries to find a Pareto optimal solution satisfying them as well as possible.

3.5.3 *a posteriori* methods

A representation of a set of Pareto optimal solution is first generated and then the DM is supposed to select the most preferred one among them. This approach gives the DM an overview of different solutions available but if there are more than two objectives in the problem, it may be difficult for the DM to analyze the large amount of information.

3.5.4 Interactive methods

After each iteration, some information is provided to the DM in order to specify preference information. What is noteworthy is that the DM can specify and adjust one's preferences between each iteration and at the same time learn about interdepencies between each iteration and at the same time learn about interdependencies in the problem as well as one's own preferences.

3.6 Solution Methods

MOO solution methods fall under the category of scalarization or non-scalarization methods. Scalarization is the primary method for MOO problems [Miettinen 1999]. Scalarization converts the MOO problem into a paramterized single-objective problem which can be solved using using well-established single-objective optimization methods.

3.6.1 Scalarization Methods

3.6.1.1 Weighting Method

3.6.2 Cost Function

$$C(\boldsymbol{\theta}) = \sum w_i(\hat{q}_i(\boldsymbol{\theta}) - q_i)^2 \tag{3–18}$$

Gass and Saaty 1955 Zadeh 1963

For a interatomic potential being fit with respect to k quantities of interest,

$$\begin{aligned} \underset{\boldsymbol{\theta}}{\text{minimize}} \quad & \sum_{i=1}^{k} w_i \varepsilon_i^2(\boldsymbol{\theta}) \\ \text{subject to} \quad & \boldsymbol{\theta} \in \boldsymbol{\Theta} \end{aligned} \tag{3–19}$$

where $w_i \geq 0$ for $i = 1, ..., k$ Weakly Pareto optimal.

In the development of interatomic potentials, the DM is asked to specify weights in which case the method is used as an *a priori* method.

Algorithms for multiobjective optimization should produce Pareto optimal solutions, and that any Pareto optimal solution can be found. Censor1977 discusses the conditions which the whole Pareto set can be generated by the weighting metho when positive weights are presented. In this respect, the weighting method has a serious shortcoming. Any Pareto optimal solution can be found by altering weights only if the problem is convex. Some Pareto optimal solutions of nonconvex problems cannot be found regardless of how the weights are selected.

The problems of the weighting schemes have ben explored by the classical potential development community. The method may jump from one vertex to anohter vertex leaving intermediate solutions undetected with relatively small changes in the weighting schemes.

Scaling of the objective functions.

The weighting method can be used as an *a posteriori* method where different weight can be used to generate different Pareto optimal solutions, and then the DM selects the most satisfactory solution. Systemic methods of perturbing the weights to obatain

different Pareto optimal solutions are suggested (Chankong and Haimes 1983), but Das and Dennis 1997 illustrates that an evenly distributed set of weights does not necessarily produce an evenly distributed representation of the Pareto optimal set, even when the problem is convex.

When the weighting scheme is used as an *a priori* method, the DM is expected to represent his/her preferences in the form of weights. Roy and Mousseau (1996) suggests that the role of weights in expressing preferences maybe mis leading. Although the relative importance of weights show the relative importance of the objective functions it is not clear what underlies this notion. The relative importance of objective functions is usually understood globally, for the entire decision problem, while many practical applications show that the importannce typically varies for different objective function values, that is, the concept is only meaningful locally. (Podinovsky 1994).

Weights that produce a certain Pareto optimal solution are not necessarily unique, and different weights may produce similar solutions. On the other hand, a small change in weights may cause big differences in the objective function. It is not easy for the potential developer to control the solution process because weights behave in an indirect way.
The solution process then becomes an interactive one where the DM trues to guess such weights that would produce a satisfactory solution, and this is not desirable because the DM cannot be properly suppported which leads to frustation complications in potential development. Instead, in such cases it is advisable to use real interactive methods where the DM can better control the solution process with more intuitive preference information.

The weighting method is also difficult

3.7 Optimization Methods

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}} \quad f(x) \\ &\text{subject to} \quad g_i(x) \leq 0 \quad h_j(x) = 0 \quad x \in X \end{aligned} \tag{3–20}$$

here x is the optimization variable, f is the objective function, g_i are inequality constraints, and h_j are equality constraint functions.

The concept of genetic algorithms were inspired by evolutionist theories explaining the origin of species[26]. In nature, weak and unfit speicies within their environment are faced with extinction by natural selection, while strong ones pass on their genes to future generations through reproduction. In the long run, species carrying the correct combinatioon in their genes become dominant in their population.

In GA terminology, a solution vector $\boldsymbol{x} \in \boldsymbol{X}$ is called an individual or a chromosome. Chromosomes are made of descrete units called genes. Each gene controls on or more features of the chromosome. Normally, a chromosome corresponds to a unique solution $\boldsymbol{x}$ in the solution space. This requires a mapping mechanism between the solution space and chromosome. GA operates with a collection of chromosomes, called a population. As the search evolves, the poulation includes fitter and fitter positions, eventually it converges, meaning that it is dominated by a single solution. Two operators are defined crossover and mutation. In the crossover operator, two parent solutions are combined togehter to form offspring. The mutation operator introduces random changes into the population.

The first multi-objective GA, called vector evaluated GA (or VEGA), was proposed by Schaffer [5]. Afterwards, several multi-objective evolutionary algorithms were developed including Multi-objective Genetic Algorithm (MOGA) [6], Niched Pareto Genetic Algorithm (NPGA) [7], Weight-based Genetic Algorithm (WBGA) [8], Random Weighted Genetic Algorithm (RWGA)[9], Nondominated Sorting Genetic Algorithm (NSGA) [10], Strength Pareto Evolutionary Algorithm (SPEA) [11], improved SPEA (SPEA2) [12], Pareto-Archived Evolution Strategy (PAES) [13], Pareto Envelope-based Selection Algorithm (PESA) [14], Region-based Selection in Evolutionary Multiobjective Optimization (PESA-II) [15], Fast Non-dominated Sorting Genetic Algorithm (NSGA-II) [16], Multi-objective Evolutionary Algorithm (MEA) [17], Micro-GA [18], Rank-Density Based Genetic Algorithm (RDGA) [19], and Dynamic Multi-objective Evolutionary

Algorithm (DMOEA) [20]. Note that although there are many variations of multi-objective GA in the literature, these cited GA are well-known and credible algorithms that have been used in many applications and their performances were tested in several comparative studies.

Vector Evaluated Genetic Algorithm (VEGA). Schaffer proposed VEGA for finding multiple solutions to multiobjective problems. He created VEGA to find and maintain multiple classification rules in a set covering problem. VEGA tried to achieve this goal by selecting a fraction of the next generation using one of the objective functions.

Fitness Sharing encourage the search in unexplored section of a Pareto front by artificially thinning solutions in densely populated area. To achieve this goal, densely populated areas are identified and a penalty method is used to penalize the solutions located in such areas. This approach was recommended by Goldberg and Richardson[27] and used by Fonseca and Fleming[28] to penalize clustered solutions.

$$dz(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{\sum_{k=1}^{K} \left(\frac{z_k(\boldsymbol{x_1}) - z_k(\boldsymbol{x_2})}{z_k^{max} - z_k^{min}} \right)^2} \tag{3–21}$$

based on these distances, calculate a niche count for each solution $\boldsymbol{x} \in \boldsymbol{X}$ as

$$nc(\boldsymbol{x}_1, t) = \sum_{\boldsymbol{x}_2 \in \boldsymbol{X}, r(\boldsymbol{x}_2,t)=r(\boldsymbol{x}_1,t)} \max \left\{ \frac{\sigma_{share} - dz(\boldsymbol{x}_1, \boldsymbol{x}_2)}{\sigma_{share}}, 0 \right\} \tag{3–22}$$

where σ_{share} is the niche size by defining a neighborhood of solutions in the objective space. Solutions in the same neighborhood contribute to each other's nich count. Therefore, a solution in a crowded neighborhood will have a higher niche count, reducing the probability of selecting that solution from being culled from the survivor set.

3.8 Visualization

This problem is dealt with in discussions about visualization and and analysis of the large amounts of data generated from a posteriori approaches to solving these problems. Edgeworth 1881 Koopmans 1951 Kuhn Tucker 1951 Pareto 1896, 1906

3.9 Treatment

Our treatment of the mapping of the empirical potential is treated as a bijective mapping into two measure spaces.

Let us define parameter space with the probability measure space $(\Theta, \mathcal{F}(\Theta), \mathbb{P})$.

Then we define the error space of the structure property relationships with the probability measure space $(\mathcal{E}, \mathcal{F}(\mathcal{E})), \mathbb{Q})$.

To solve forward problems, the parameters of a potential, $\boldsymbol{\theta}$ is known *a priori*, are used in conjunction of a set of atomic arrangements in a simulation cell with periodic boundary conditions to predict n material properties, $\boldsymbol{q} = (q_1, ...q_n)$. These predictions depend not only on the atomic arrangements but also on the parameterization, denoted $\hat{\boldsymbol{q}}(\boldsymbol{\theta}) = (\hat{q}_1(\boldsymbol{\theta}), ...\hat{q}_n(\boldsymbol{\theta}))$. The differences between the predicted values and references values are denoted $\boldsymbol{\epsilon}(\boldsymbol{\theta}) = |\hat{\boldsymbol{q}}(\boldsymbol{\theta}) - \boldsymbol{q}_i|$, where $|\boldsymbol{x}|$ is the elementwise magnitude of the vector $\boldsymbol{x}$.

The problem of parameterization is an inverse problem where an optimal parameterization produces ideal outcomes for the forward problem, i.e. difference between the predicted value and the reference value, $\epsilon_i(\boldsymbol{\theta}) = 0$. Since replication of results is typically not achievable, then the goal of parameterization becomes $\min_{\boldsymbol{\theta}} \epsilon_i(\boldsymbol{\theta})$ for all i. Typically, there does not exist an optimal parameterization, $\boldsymbol{\theta}^*$, which minimizes $\epsilon_i(\boldsymbol{\theta})$ for all $i < n$. Requiring a prioritization of which material properties have a preference for fidelity in predictions.

The typical approach to solving the inverse problem transforms the above problem into a scalar optimization problem amenable to derivative approaches. A cost function C which couples the individual objectives, ϵ_i, along with a set of weights $\boldsymbol{w} = (w_1, ..., w_n)$, to represent preferences, that is

$$C(\boldsymbol{\theta}) = \sum_i^n w_i(\hat{q}_i(\boldsymbol{\theta}) - q_i)^2 = \sum_i^n w_i \epsilon_i^2(\boldsymbol{\theta}) \tag{3–23}$$

It is clear that the selection of $\boldsymbol{w}$ uniquely determines $\boldsymbol{\theta}^*$. However, the values of w_i which will produce an acceptable potential are typically not known *a priori*. When the

initial weighting scheme fails to give an acceptable results, $\boldsymbol{w}$ is changed in an *ad hoc* approach until an acceptable parameterization is achieved.

Since analytical solutions are intractable, numerical solutions are achieved by selecting an initial parameterization, $\boldsymbol{\theta}_0$, and using derivative-based optimization techniques to minimize the cost function. If the $C(\boldsymbol{\theta})$

We generalize the problem of parameter estimation by a casting it more generally into a multi-objective optimization problem:

$$\min_{\boldsymbol{\theta}} \boldsymbol{\epsilon}(\boldsymbol{\theta}) = \begin{bmatrix} \epsilon_1(\boldsymbol{\theta}) \\ \vdots \\ \epsilon_n(\boldsymbol{\theta}) \end{bmatrix} = \begin{bmatrix} \hat{q}_1(\boldsymbol{\theta}) - q_1 \\ \vdots \\ \hat{q}_n(\boldsymbol{\theta}) - q_n \end{bmatrix} \tag{3–24}$$

To remove the dependence on *a priori* performance preferences, it is necessary to define an ensemble of parameterization which are optimal in a sense. Suppose we have two parameterizations, where $\boldsymbol{\theta}_1$ dominates $\boldsymbol{\theta}_2$, denoted $\boldsymbol{\theta}_1 \prec \boldsymbol{\theta}_2$, when $\epsilon_i(\boldsymbol{\theta}_1) \leq \epsilon_i(\boldsymbol{\theta}_2) \forall i \in \{1, ..., n\}$ and $\exists i \in \{1, ...n\}, \epsilon_i(\boldsymbol{\theta}_1) < \epsilon_i(\boldsymbol{\theta}_2)$. We say that $\boldsymbol{\theta}_n$ is Pareto efficient if $\nexists \boldsymbol{\theta}_i \in \Theta, \boldsymbol{\theta}_i \not\prec \boldsymbol{\theta}_1$.

The Pareto set $\Theta^{(p)}$ is the set of all Pareto effcient points, that is the set of nondominated points. While performance requirements have not yet been encoded to determine $\boldsymbol{\theta}^*$, this point must fall in the Pareto set, $\boldsymbol{\theta}^* \in \Theta$. If ϵ_i are competiing, then clearly there are parameterizations which performs well with respect to ϵ_i, but poorly with respect to ϵ_j.

We originally defined Θ as a compact space of the parameters $\boldsymbol{\theta}$ defining the feasible θ-space. In a deterministic approach we would want to identify an algorithm such that we start with feasible set of parameterizations and constrains the sets of parameterizations until it produces a set of parameterizations which produces the Pareto set in ϵ-space, that is a process

$$\Theta = \Theta_0 \supset \Theta_1 \supset \ldots \supset \Theta_k = \Theta^{(p)} \tag{3–25}$$

which produces due to Eq

$$\mathcal{E} = \mathcal{E}_0 \subset \mathcal{E}_1 \supset \ldots \supset \mathcal{E}_k = \mathcal{E}^{(p)} \tag{3–26}$$

for $k < \infty$ iterations. Since $\Theta \subset \mathbb{R}^p$ and $\mathcal{E} \subset \mathbb{R}^n$, we provide the following approach which uses Monte Carlo simulation in an approach which is inspired by Bayesian inference, although this approach does not use a Bayesian updating approach. The goal of this approach is to produce an ensemble of $\boldsymbol{\theta} \in \Theta^{(p)}$ and describe this ensemble as a probability distribution which could be used as a starting point in uncertainty quantification propagation.

We propose the following approach:

$$\Theta_k \rightarrow \hat{Q}_k(\Theta_k) \rightarrow \mathcal{E}_k(\Theta_k) \tag{3–27}$$

$$\mathcal{E}_k(\Theta_k) \rightarrow \mathcal{E}_k^{(p)}(\Theta_k^{(p)}) \tag{3–28}$$

$$\mathcal{E}_k^{(p)}(\Theta_k^{(p)}) \rightarrow \mathcal{E}_k^{(cp)}(\Theta_k^{(cp)}) \tag{3–29}$$

$$\mathcal{E}_k^{(cp)}(\Theta_k^{(cp)}) \rightarrow \Theta_{k+1} \tag{3–30}$$

The notation $\rho(\boldsymbol{\theta})$ refers to the joint probability density function that $\boldsymbol{\theta} \in \Theta^P$. Intuitively, one can think of $\rho(\boldsymbol{\theta})\Delta\boldsymbol{\theta}$ as the probability that a random variable drawn from $\rho(\boldsymbol{\theta})$ will fall within the infinitesimal compact set $[\boldsymbol{\theta}, \boldsymbol{\theta} + \Delta\boldsymbol{\theta}]$

Even if Θ is defined as compact, $\hat{Q}$ may not be bounded. By construction $\hat{q}_i > 0$, however $\hat{q}_i$ may not be bounded from above. There exists $\boldsymbol{\theta} \in \Theta$ which produces pathological members of the Pareto set. Specifically, there is exists $\boldsymbol{\theta} \in \Theta$ such that $\boldsymbol{\epsilon}(\theta) \in \mathcal{E}^(p)$, but produces an $\epsilon_i(\boldsymbol{\theta}) > \epsilon_{i,max}$ for at least one $i \in \{1, ..., n\}$, where $\epsilon_{i,max}$ is an arbitrary performance requirement.

We generalize Eq To estimate $\Theta^{(p)}$, we simplify the drawing of samples from a uniform distribution defined by hyperrectangles which defines ΘThe choice of

3.9.0.1 Kernel Density Estimate

The Kullbach-Leiber divergence[29] measures the divergence between two probability density functions $f(x)$ and $g(x)$,

$$D(f \parallel g) = \int f(x) \log \frac{f(x)}{g(x)} dx \tag{3–31}$$

is commonly used in statistics as a measure of similarity between two density distributions, and has the following properties: (1) self-similarity, $D(f \parallel f) = 0$, (2) self-identification, $D(f \parallel g) = 0$ only if $f = g$, and (3) positivity, $D(f \parallel g) \geq 0$ for all f and g.

The integral in Equation 3–31 can be calculated from Monte Carlo[30], by drawing a sample x_i, from the statstical distribution of f such that $\mathbb{E}\left[\log \frac{f(x)}{g(x)}\right] = D(f \parallel g)$. Using N i.i.d. samples $\{x_i\}_{i=1}^{N}$, we have

$$D_{MC}(f \parallel g) = \frac{1}{N} \sum_{i}^{N} \log \frac{f(x)}{g(x)} \rightarrow D(f \parallel g) \tag{3–32}$$

as $n \rightarrow \infty$. The variance of the estimation error is $\frac{1}{N}\mathrm{Var}_f\left[\log \frac{f}{g}\right]$. To compute $D_{\mathrm{MC}}(f \parallel g)$, we need to generate samples $\{x_i\}_{i=1}^{N}$ from f. Then for $1 \leq i \leq N$, evaluate $f(x_i)$ and $g(x_i)$ to calculate D_{MC}.

3.10 Methodology

To demonstrate the potential of this process to develop a working potential, a Coulumb-Buckingham potential[31] is developed for magnesium oxide (MgO). This pair wise potential for atoms i and j

$$V(r_{ij}; A, \rho, C) = \frac{Z_i Z_j}{4\pi\varepsilon_0 r_{ij}} + A \exp(-\frac{r_{ij}}{\rho}) - \frac{C}{r_{ij}^6} \tag{3–33}$$

where $r_{ij} = \|\boldsymbol{r}_i - \boldsymbol{r}_j\|_2$ is the distance between the atoms i and j, and q_i areq_i describe the charges of the atoms, and A, B, and C, are the parameters of the potential.

The first term of the potential is the electrostatic potential energy, the second term is repulsive due to the Pauli exclusion principle, and the third term is an attractive van der Waals energy.

We use the same relevant assumptions used in Lewis and Catlow[31], the Mg-Mg interactions are assumped to be purely coulombic, the Mg-O is considered to be the Born-Mayer form, $A\exp(-r/\rho)$, where the van der Waals term is ignored.

The charge of the atoms is allowed to deviate from their formal charges, provided that $Z_{Mg} = -Z_O$, to preserve charge neutrality.

3.10.1 Reference Values

3.10.2 Implementation

Implemented in Python using LAMMPS as the molecular dynamics engine as the calculator. Parrellization is done through MPI.

CHAPTER 4
PROBABILITY METHODS

4.1 Probability

The discussion of the probability methods used in this work starts with the probability construction by Kolmogorov, which builds probability as a measure of sets, but adapted to the particular necessities of probability. This section clarifies terminology used throughout as this work, which at some points it is more convenient to talk about continuous probability distribution functions, and at other times using the notation for the probability of sets. What follows is a necessarily brief introduction to probability theory in a more rigorous sense, we refer the reader to classic textbooks of Rudin[32] for a rigorous treatment of measure theory and Chung[33] for a measure theory construction of probability.

Let $(\Omega, \mathcal{F}, \mathbb{P}$ be a measure space with $\mathbb{P}(\Omega) = 1$. Then $(\Omega, \mathcal{F}, \mathbb{P}$ is a probability space, with sample space Ω, event space $\mathcal{F}$, and probabilty measure $\mathbb{P}$.

A random variable X is a variable whose possible values are outcomes of a random phenemon. As a function, a random variable is required to be measurable, which rules out pathological issues.

The underlying foundation of any probability distribution is the sample space, which is the set of all probable outcomes denoted as Ω. The realization of an outcome is denoted $\omega \in \Omega$.

The events for the measure space are defined in such a way that a probability measure can be assigned. This allows to assign probability measures on complex events to characterize groups of outcomes. The collection of all such events is a σ-algebra $\mathcal{F}$ of subsets of Ω. Not every subset of the sample space Ω must be considered an event, the σ-algebra restricts $\mathcal{F}$ to subsets of Ω for which $\mathbb{P}$ can be asssigned.

The probability measure, $\mathbb{P}$, a function which maps $\mathbb{P} : \mathcal{F} \to [0, 1]$. A probability is a real number between zero and one. Within this work we do not disguish the difference

between impossible events which have probability zero, and probability-zero events which are not necessarily impossible. Events of probability one is an event that happens almost surely, with almost total certainty.

The triplet $(\Omega, \mathcal{F}, \mathbb{P})$ is the probability measure space.

4.1.1 Random Variable

A random variable has a probability distribution, which specificies the probability falls in. Specifically, $X : \Omega \to \mathbb{R}$. If a random variable $X : \Omega \to \mathbb{R}$ is defined on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$. Then the probability of an event A occuring is $\{\omega : X(\omega) = A\}$ which is denoted as $\mathbb{P}(X = A)$.

A probability density function for a random variable X has a density f_X, where f_X is a non-negative function:

$$\mathbb{P}[a \leq X \leq b] = \int_a^b f_X(x)dx \tag{4–1}$$

4.1.2 Probability Density Function

For example, the uniform distribution

4.1.3 Probability Density Function

4.1.4 Joint Probability Density Function

4.1.5 Sampling from a Distribution

In this section, we discuss the general concept of sampling using the technique of inverse transform sampling as a method for pseudo random number generation. Inverse sampling samples from a uniform distribution, $u_i = U(0, 1)$, and interprets it as a probability, and then returns the largest number x from the domain of the distribution $\mathbb{P}(X)$ such that $\mathbb{P}(-\infty < X < x) \leq u_i$.

Computationally, this method involves computing the CDF and inverting the function, which is known as the quantile function.i

Note that for a discrete distribution, computing the CDF is not in general too difficult: we simply add up the individual probabilities for the various points of the

distribution. For a continuous distribution, however, we need to integrate the probability density function (PDF) of the distribution, which is impossible to do analytically for most distributions (including the normal distribution).

4.1.6 Expectation

If X is a random variable with a finite number of outcomes, $\{x_1, x_2, ..., x_k\}$ occurs with the probabilities $\{p_1, p_2, ...p_k\}$. Then the expectation of X is defined as

$$\mathbb{E}[X] = \sum_{i=1}^{k} x_i p_i \tag{4–2}$$

If X is a continuous random variable, then

$$\mathbb{E}[X] = \int_{\Omega} X(\omega) d\mathbb{P}(\omega) \tag{4–3}$$

If X admits a density $f(x)$, then the expected value is defined as

$$\mathbb{E}[X] = \int_{\mathbb{R}} x f_X(x) dx \tag{4–4}$$

4.2 Estimation of a Distribution

In the previous section, we have discussed

4.2.1 Parametric Estimation

4.2.2 Non-parametric Estimation

Fusce eget tempus lectus, non porttitor tellus. Aliquam molestie sed urna quis convallis. Aenean nibh eros, aliquam non eros in, tempus lacinia justo. In magna sapien, blandit a faucibus ac, scelerisque nec purus. Praesent fermentum felis nec massa interdum, vel dapibus mi luctus. Cras id fringilla mauris. Ut molestie eros mi, ut hendrerit nulla tempor et. Pellentesque tortor quam, mattis a scelerisque nec, euismod et odio. Mauris rhoncus metus sit amet risus mattis, eu mattis sem interdum.

4.2.3 Kernel Density Estimation

4.2.3.1

Let $(x_1, x_2, ..., x_n)$ be a univariate and identically distributed sample drawn from some distribution with an unknown density f. The goal is to estimate the shape of this function f. The kernel density estimator is

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{4–5}$$

where K is the kernel and $h > 0$ is a smoothing parameter called the badwidth. The kernel function satisfies the condition

$$\int_{-\infty}^{+\infty} K(x)dx = 1 \tag{4–6}$$

4.2.3.2 Choice of kernels

Popular kernels: Epanachnikov, Bi-weight, Triangular, Gaussian, Rectangular.

For the kernel method, we adopt the gaussian kernel ϕ,

$$\phi(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{t^2}{2\sigma^2}\right) \tag{4–7}$$

4.2.3.3 Selection of bandwidth parameters

The chosen bandwith is important because it has a strong influenjce on the boundary of the density curve. The curve boundary has poor smoothness quality when bandwidth takes small value; while as the increasing bandwidth, the smoothness improves, but the fitness of the curves becomes poor. The accuracy of kernel estimation is dependent on suitable bandwidth.

Scott' Method

Silverman Method

Chiu Method

4.3 Bayesian Estimation

4.4 Sampling

Requirements for kernel

Table 4-1. Sample size required to ensure relative mean square error at zero is less than 0.1, when estimating a standard normal density using a normal kernel and the window width that minimize the mean square error loss at zero[1]

Dimenstionality	Required Sample Size
1	4
2	19
3	67
4	223
5	768
6	2790
7	10700
8	43700
9	187000
10	842000

4.5 Monte Carlo Methods

Monte Carlo methods are a broad class of computational algorithms which are dependent upon random sampling to obtain numerical results. The essential aspect of these approaches is to use randomness to solve problems which might be deterministic in principle. Monte Carlo simulations sample from a probability distribution for each variable to produce an arbitrarily large number of possible outcomes, and the results are analyzed to get probabilities of different outcomes occuring.

CHAPTER 5
AN EVOLUTIONARY ALGORITHM FOR GENERATING PARETO EFFICIENT POTENTIALS

Gradient based optimizers are efficient at finding local minima for high dimensional, non-linearly constrained, convex problems; however, gradient methods have problems dealing with noisy or discontinuous functions, and are not designed to handle multi-modal problems or discrete or mixed descrete-continuous design variables. In these cases, there are different options available to the potential developer including: multiple restarts of the gradient based optimizer for different initial conditions, which requires multiple guesses at initial starting parameterizations, $\boldsymbol{\theta}_0$; systematically searching the design space, and using a gradient free minimizer.

When we cast the potential optimization problem from a single objective optimization problem to a multiple objective optimization problem, the problem becomes more difficult. In order to explore the optimal parameterization space, However, in previous literature genetic algorithms are used to optimize potentials.

Our algorthm has the following goals: (1) to identify the strengths and weaknesses of solution of the Pareto optimal solutions, (2) to generate estimates of the Pareto optimal front in a serious of iteratively better approximations, and (3) to describe the candidate parameterizations through the use of a distribution function and use MCMC sampling, but updating the distribution using culling of the Pareto distribution.

5.1 Genetic Algorithm

Genetic algorithms are a popular meta-heuristic that is particularly well-suited for this class of problems. Traditional GA are customized to accomodate multi-objective problems by using specialized fitness functions and introducing methods to promote solution diversity. The use of evolutionary algorithms in the development of classical potentials is not new, and numerous optimization approaches such as gradient-based approaches, genetic algorithms, and neural networks have been developed. Genetic algorithms

- Set $t = 1$. Randomly generate N soluitions to form the first population, P_1. Evaluate the fitness of solutions in P_1
- Crossover
- Mutation
- Fitness assessment
- Selection. Select N solution from Q_t based on their fitness and copy them to P_{t+1}
- If the stopping criterion is satisfied, terminate the search and return to the current population, else set $t = t + 1$ and go to step 2.

5.2 The Problem

Using the construction of an empirical potential as outlined in Chapter 2, we define an empirical interatomic potential, $\hat{V}$, which approximates the potential energy surface, V. If $\boldsymbol{R}$ is an atomic configuration, then both the EIP and the PES maps the configurational space onto energies, $\hat{V} : \boldsymbol{R} \to \mathbb{R}$ and $V : \boldsymbol{R} \to \mathbb{R}$. We can then write

$$\hat{V})(\boldsymbol{R}) = V(\boldsymbol{R}) + \epsilon(\boldsymbol{R}) \tag{5–1}$$

where $\epsilon(\boldsymbol{R})$ is a difference equation required for equality balance.

$$\epsilon(\boldsymbol{R}) = \hat{V}(\boldsymbol{R}) = V(\boldsymbol{R}) \tag{5–2}$$

Since $\epsilon \to 0$ as the EIP becomes a better approximation to the PES, we will use ϵ to define loss functions and performance filters.

If $\hat{V}$ is an analytical parameterized by P parameters, $\boldsymbol{\theta} = [\theta_1, ..., \theta_P] \in \mathbb{R}^P$

5.3 A Probability Approach

In a deterministic approach, a parameterization, θ, is a member of the domain Θ, and we use numerical routines to identify the optimal parameterization $\theta^* \in \Theta$.

Here we take a probabilistic approach to potential development. Here Θ is a random variable, and $\theta \in \Theta$ is a specific realization of that random variable.

For the purpose of generality, we use $\hat{V}$ to predict a set of material values $\hat{q}$ from which we have target values $\boldsymbol{q} \in \mathbb{R}$

We replace the notion of θ being a deterministic value, and instead $\theta \in \Theta$

5.4 The Fitting Database

5.5 An Iterative Procedure

5.6 Incorporation of Prior Knowledge

5.7 Sampling and Filtering

5.8 Kullbeck-Leiber Divergence

The Kullback-Leiber divergence[29], $D_{KL}(\rho_1||\rho_2)$, measures how one probability measures how one probability distribution, ρ_1, diverges from a second probability distribution function, ρ_2. For continuous random variables, D_{KL}, is defined as the integral

$$D_{KL}(\rho_1||\rho_2) = \int \rho_1(x)\frac{\rho_1(x)}{\rho_2(x)}dx \tag{5–3}$$

$D_{KL} > 0$, with $D_{KL}(\rho_1||\rho_2)$, when $\rho_1 = \rho_2$ almost everywhere. For our application, our distributions are KDEs, so the evaluation of the integral can be done by Monte Carlo integration. As the number of iterations, i, increases, the Kullbach-Leiber divergence $D_{KL}(\rho_{i-1}||\rho_i)$ convergences asymtotically to zero. Since early iterations are more likely to cause changes in the set approximating $\boldsymbol{\Theta}^*$ than later simulations, changes in D_{KL} will initially be large. However, it becomes increasingly more difficult to identify further Pareto-optimal solutions later in the simulation. As a result, the distribution becomes more stationary. However, since this integral is evaluated by Monte Carlo estimation, the KDE estimate of the distributions from the sample population will have small divergences betweem, $\rho_i(x)$ and $\rho_{i-1}(x)$.

CHAPTER 6
POTENTIAL DEVELOPMENT SOFTWARE

6.1 Background

The simulation of atoms involving hundreds of atoms are commonplace, due to the success of density functional theory[8, 9] and the availability of many software packages avail for the calculations such as VASP[34–36], ABINIT[37–40], and Quantum Espresso[41]. These electronic-structure calculations are high-fidelity calculations, which accuracy improving as the description of the exchange correlation energy functional has improved from local density approximation(LDA) to PBE to hybrid methods. Thesg electronic-structure models allow for simulations of hundreds of atoms which when combined with workflow management software, such as AFLOW[42] and `pymatgen`[43] has given rise to high-throughput computational efforts, which leverage these energy calculators.

As an alternate to electronic structure methods, the use of empirical potential that describe the effects of the valence electron interactions without explicitly describing the electrons themselves. The simplief descriptions of interatomic interations allow for larger system sizes and longer simulations timeframes than can be accomplished with *ab initio* techniques. However, these approaches are accompanied by a loss of accuracy compared to electronic structure methods.

In this chapter, a software toolkit for the reproducible, algorithmic development of interatomic potentials for atomic-level simulations using autonomous machine-learning techniques is described. The *Python Potential Optimization Software Package*(`pyospack`) is open-access software for the automation of potential development workflows, which leverages the richness of machine learning codes of the python language with ubiquitous molecular dynamics software code, LAMMPS[44], and the lattice dynamics code, GULP[45].

Nevertheless, the process of potential development largely remains non-transparent and subjective, involving the repeated intervention of a skilled potential developer.[46, 47] As a result, the final parameterization depends on the many choices made by the potential developer; this means that the process by which a potential is developed is generally neither fully documented, nor reproducible. Moreover, there is currently no objective method for evaluating the suitability of the function form of the atomic potential or determining if the final parameterization selected yields the best possible fit to the fitting database.

6.2 Introduction

Classical atomistic simulation methods, of which molecular dynamics (MD) simulation[48–51] is the most common, are a vital tool in the analysis of solid state and materials systems. The description of the interactions of the atoms is encoded in the interatomic potential, many of which have been developed to describe specific materials systems. The embedded atom method (EAM)[52–55] and Finnis and Sinclair[56] potentials, among others, were developed and continue to be developed for metals. Bond order potentials (BOP) such as those of Brenner[57, 58] and Tersoff[59], and the three-body Stillinger and Weber[60] potential are widely used to describe covalently-bonded materials. For ionically bonded materials, the electrostatic interactions are typically described by Coulomb potentials, with various formalisms for the short ranged interactions, the Buckingham potential being the most widely used.[61, 62] The continuing evolution of these formalisms, the development of more sophisticated potential formulations such as ReaxFF[63, 64] and COMB [65, 66], and the increasing accuracy of density functional theory (DFT) calculations, which typically constitute at least part of the fitting database, have allowed the materials fidelity of these potentials to increase markedly. Nevertheless, the process of potential development largely remains non-transparent and subjective, involving the repeated intervention of a skilled potential developer.[46, 47, 67] As a result, the final parameterization depends on the many choices made by the potential developer;

this means that the process by which a potential is developed is generally neither fully documented, nor reproducible. Moreover, there is currently no objective method for evaluating the suitability of the function form of the atomic potential or determining if the final parameterization selected yields the best possible fit to the fitting database.

The key driver of this software is to implement a systematic methodology to fit interatomic potentials that allows the objective evaluation of the quality of the parameterization, and the stability of the functional form, and that is both completely algorithmic and reproducible. Current parameterization processes generally involve the minimization of a single scalar cost function, typically a weight sum of some measure of the error (typically, absolute difference of square difference, as discussed in chapter 3) of the predicted value, $\hat{q}_i\boldsymbol{\theta}$ and the reference value of the specific material property $\hat{q}_i$. This approach has a number of challenges:

- It is necessary to begin the optimization process from a single initial guess of a vector of parameters, $\boldsymbol{\theta}_0$, which we evolve through scalar optimization process usually dependent upon the calculation of gradients and Hessians of the the cost function with respect to each parameter, until it converges to the optimal parameterization $\boldsymbol{\theta}^*$, which maybe frustrated by issues of convexity and ill-conditioned behavior of the optimization problem;
- In the presence of a solution space, that has many local minima, the final potential is dependent upon the selection upon the choice of initial conditions; and
- Most importantly, the weights chosen for the fitting by the developer purport to represent the priorities of the potential develop in terms of the size of acceptable erros in the predicted values, when no such direct correspondence between the weights and errors in the final potential predictions.

6.3 Fitting Process

Instead of looking at gradient approaches to optimization, The challenge for potential fitting is to identify an optimal parameterization, $\boldsymbol{\theta}^*$, that yields the smallest errors in predicted properties. We cast this as a multi-objective optimization problem, as commonly used in the field optimal criteria decision making. In attacking this MOO problem, we need an appropriate descriptor for the set of possible solutions before the developers

preferences are expressed; we adopt the concept of Pareto optimality. A Pareto optimal solution is a point in parameter space at which the reduction in error in the predicted value of any materials property can take place only through an increase in the error in one or more other materials property.

A parameterization $\boldsymbol{\theta}_2$ dominates $\boldsymbol{\theta}_1$, when $|\epsilon_i(\boldsymbol{\theta}_1)| \leq |\epsilon_i(\boldsymbol{\theta}_2|$ for all i; the parameter vector $\boldsymbol{\theta}_i$ is Pareto optimal if it is not dominated by any other $\boldsymbol{\theta}_j$. The set of parameter vectors, $\boldsymbol{\Theta}^{(p)}$, contains all the parameterizations for multi-objective optimization problem.

6.4 Implementation

Software for potential software development is by necessity a complicated piece of software which draws expertise from different functional expertise. In addition to the faithful execution of potential formalisms, software for potential development must be able to simulate a wide range of materials properties, implement optimization routine, and conduct post-fitting analysis.

Our optimization process, identifies a set of parameterization, which produces Pareto optimal results, but it is still necessary for the potential developer to identify which parameterization is most suitable for this application.

In one use case, the potential developer will look to approximate the parameter set which produces Pareto optimal predictions. Since parameter space is explored by sampling from a probability distribution, this effort is embarassingly parallelizable and we wish to scale this effort over the large number of processors available in a high performance cluster (HPC) environment likely available to potential development. The typical use case for this software package involves a potential developer using the software program by modifying the examples distributed with this program for their particular needs.

On the other hand, once the optimization process is complete, the potential developer needs to explore the tradeoffs in performance between different QOIs, explore relationships between the different parameters, and select potentials of interest. Now, the potential developer uses the same software library which aids them in interactive, exploratory data

analysis, likely requiring *ad hoc* visualization. In this use case, `pypospack` enables users to quickly create their own analysis and visualization routine, by leverage the same core packages to meet their specific needs, typically on a workstation.

As a result, our potential development software cannot be delivered as a monolithic software application, but must support a workflow which allows for interactive, exploratory data analysis. It needs to be largely platform agnostic, capable of running on HPC clusters as well as personal workstations.

6.5 Implementation

6.5.1 Underlying Technologies

Pypospack was written in python to leverage the strength of python as a high-level language for writing scientific applications. Python has a liberal open source license which makes the distribution of the application without license issues. Since Python is available on many platform, there are few issues with portability between platform. Python flexible syntax allows potential developers to either use the pypospack library to define a potential develop process as a simple procedural script, or allow the developers to encapulate implementation details away from the end users. We look at the success of the materials project, where the materials properties can be predicted by solving fundamental laws of physics using quantum mechanical appoximation such as density functional theory (DFT). This virtual testing of materials was employed to design and optimize materials *in silico*.

`pypospack` is written in python. The python language has a clean syntax yet has sophisticated constructs which allows is indifferent to either procedural or object-oriented programming styles, as the situation dicates. Software development for this program was developed using procedural code, which was later encapsulated to classs object to abstract the implementation details into base objects.

Additionally, `pypospack` utilizes popular, open-source packages from the python community. Moreover, *pypospack* takes a different approach to the development of

potentials by identifying a set of Pareto optimal potentials through an evolutionary stochastic optimization algorithm.

Python is popular programming language that is popular for scientific applications, due to the maturity and stability of fundamental numerical libraries, quality of documentation, and availability of well-supported distrbutions, such as Anaconda[68], makes Python accessible and convenient for a broad audience. Additionally, matplotlib[69] integrated with IPython[] provides an interactive research and development environment with data visualization suitable for most users. As a result, is an appealing choice for algorithmic development and exploratory data analysis[70]

NumPy[71] adds an array language, similar in syntax to MATLAB, and similar in power to Fortran, in which operations are performed in compiled code. Scipy[72] builds on top of NumPy to provide functionality for optimization, numerical integration, and a statistics package for creating random variates, and a linear algebra package which provides extended interfaces to BLAS[73] and LAPACK[74]. When possible, the data produced by `pyopspack` is exposed through Pandas[75] to simplify data management and data analysis tasks using well known syntax. Scikit-learn[76] integrating a wide range of machine learning algorithms for medium-scale supervised and unsupervised problems. This package focuses on bringing machine learning to non-specialists using a general-purpose high-level language. MPI for Python(mpi4py([77, 78] provides bindings of the Message Passing Interface (MPI)[79] standard for the Python programming language and allows the exploitation of multiple processors in an HPC environment. This is important for ease of installation and portability, as providing libraries around Fortran code can prove challenging on various platforms.

To run simulations, `pyopspack` spawns a new process, through the Python `subprocess` module, and obtain the exit codes, which is returned from the child process and is interpreted by `pyopspack` in the event of external codes returning a success or failure. Through this facility, input files are made for the external executable, which is run under

a child subprocess until an exit code is detected, when the output files of the energy calculator are then parsed for pertinent information. Currently, `pypospack` supports LAMMPS and GULP as external energy calculators

Software for potential software development is by necessity a complicated piece of software which draws expertise from many disciplines for software development. The goal of pypospack is not to deliver a monolithic software application, but provide a software with a flexible software architectural library from which potential developers can quickly create their own potential optimziation software, by leveraging an object-orient software framework based upon a series of core packages, which deliver specific functionality to `pypospack`.

6.5.2 Potential

Configuration space consists of the position of each atom, $\boldsymbol{r} = \{r_1, r_2, r_3\}$, embedded in a periodic volume used to model the infinite bulk defined by the collective basis vectors, $\mathbb{R}^3$, $H = (\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3)$. For the purposes of notational compactness, we will refer to a configuration of atoms as R, and treat the potential energy surface as a function, $U : \boldsymbol{R} \rightarrow \mathbb{R}$.

An empirical potential, $\hat{V}$, approximates the potential energy surface with a set of equations referred to collectively as the functional form. These set of equations are often chosen to capture the relevant physics or chemistry of a system, such as the Buckingham potential for ionic solids, the embedded atom method for metals, and many body potentials for covalently bonded materials. To specialize an empirical potential for material system a set of parameters, $\boldsymbol{\theta} = (\theta_1, ..., \theta_P) \in \mathbb{R}^P$, needs to be selected. Since $\hat{V} : \{\boldsymbol{R}, \boldsymbol{\theta}\} \rightarrow \mathbb{R}$, we can think of $\hat{V}$ as a function that is not only dependent upon the configuration of the atoms $\boldsymbol{R}$, but also upon the parameterization $\boldsymbol{\theta}$.

Classical interatomic potential reduce the quantum-mechanical interactions of electrons and nuclei to an effective interaction between a collection of atoms described by

an analytical set of functions. This greatly reduces the computational effort in molecular dynamics (MD) simulations.

Potentials are typically obtained by determining the potential paramters which optmize a set of reference data, which typically includes experimental values such as lattice contants, cohesive energies, elastic constants, and are supplemented with *ab-initio* obtained data such as defect formation energies.

In this section, the potential development software Python Potential Operational Software Package (*pypospack*), is presented which is a software library which automates the development of analytical interatomic potentials using evolutionary stochastic optimization techniques, described in the previous chapter.

Like other potential development software packages, *pypospack* was developed to seprate the process of parameter optimization from the selection of the analytical form of the potental. *pypospack* separates itself from other packages by designed using object-oriented methods which enables users of this software to quickly integrate new potentials, material properties, or even optimization techniques.

The `potential` package contains class objects which inherent from the appropriate abstract base class and overide the required methods for implementation. Currently, the `Potential` abstract class only requires the implementation of the `_init_parameter_names`, `lammps_potential_section_to_string`, `gulp_potential_section_to_string`, and `evaluate` methods. Each of these methods accepts a `parameter` ordered dictionary object, with the parameter name and it's respective value for key-value pairs. Depending upon the use of the potential, not all methods need to be overridden. For example, if simulations are only to be run in LAMMPS or GULP, then the only the `lammps_potential_section_to_string` and `gulp_potential_section_to_string` need to be implemented respectively.

For the implementation of the EAM potential, the potential formalism consists of three functions, which are specified in the formalism.

Table 6-1 lists the currently implemented potentials

Table 6-1. Current implemented potentials in pypospack

Class Name	Parent Class
Potential	
PairPotential	Potential
BuckinghamPotential	PairPotential
MorsePotential	PairPotential
BornMayerPotential	PairPotential
LennardJonesPotential	PairPotential
GeneralizedLennardJonesPotential	PairPotential
ThreeBodyPotential	Potential
TersoffPotential	ThreeBodyPotential
StillingerWebberPotential	ThreeBodyPotential
EamPotential	Potential
EamDensityFunction	Potential
EamEmbeddingFunction	Potential
BjsEmbeddingFunction	EamEmbeddingFunction
UniversalEmbeddingFunction	EamEmbeddingFunction
FinnisSinclairEmbeddingFunction	EamEmbeddingFunction
EamEmbeddingEquationOfState	EamEmbeddingFunction
RoseEquationOfStateEmbeddingFunction	EamEmbeddingEquationOfState

6.5.3 Execution Framework

The execution framework involves the management of the `Qoi` objects and the `Task` each parameter set. `Task` defines a simulation task, which will spawn a subprocess using the Python `subprocess` module to execute external code.

6.5.4 Simulation Tasks

At the heart of the `pypospack` conceptualization of the parameterization workflow is the decomposition of the calcuation of material properties into individual simulations.

All tasks inherit from the the abstract class `task.Task`, with an intermediate abstract class for the implementation of simulation tasks for LAMMPS(`AbstractLammpsSimulation`), and GULP(`task.gulp.GulpSimulation`. LAMMPS and GULP implementation of tasks are determined separately.

6.5.5 Quantities of Interest

6.6 Sampling Framework

6.6.1 Sampling From Parametric Distributions

6.6.2 Sampling from Non-parametric Distributions

6.6.3 Iterative Sampling

6.6.4 Input and Output

6.6.5 Configuration File

Rather than using a traditional input file, the object `PyposmatConfigurationFile` is provided, which uses the native serialization facilities to save the state of the object to a file and reinstantiate the object from a file. The `PyposmatConfigurationFile` uses a nested `OrderedDict`, which is hashtable of key-value pairs. This gives the potential developer the ability to script the creation of the configuration file in a way which is not possible with a flat file configuration. The YAML data serialization language[**?**] which is human-readable structured aroundthree primitives: mappings (such as mappings and dictionaries), sequences (arrays and lists), and scalars (strings and numbers). This data format allows portability between programming languages since it is language agnostic, in ways that the `pickle` Python object serialization does not. The branches of the YAML tree correspond to the configuration of different objects in `pypospack` which will be covered later within this chapter.

6.6.6 Structure Files

In a similar vein, `pypopsack` adopts the POSCAR file format of VASP as the standard serialization object. However, since different software packages have different structure representation. Thiss issue is implemented with inheritance. Every structure required to do a materials property calcualtion requires a structure file. Since the fitting database is often calculated from a DFT code, `pypospack` adopts the POSCAR file format from VASP as the serialization file format.

The lattice vectors which bound the simulation box, $(\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3)$, are represented mathematically as $a_0 H$, where $\boldsymbol{a}_i \in \mathbb{R}^3$, $a_0 = \|\boldsymbol{a}_1\|$, and the matrix $H \in \mathbb{R}^{3\times3}$

`SimulationCell`

6.6.7 Structure Database

6.6.8 Potential Definition

Structure files use the To that end, the configuration file is a YAML("YAML Ain't Markup Langage) is a human-readable, data-serialization language.

6.7 Parallelization

One of the key focuses of `pypospack` is to keep pace with the evolution of high-end supercomputers with an effort to developing the software to execute on not only on desktop workstations but high performance cluster computing.

The first iteration of this software uses a simple parallelization scheme taking advantage of MPI.

6.8 Software Architecture

pypospack is a object-oriented framework, written in Python3 and targeted to a packages curated and maintained to the Anaconda software distribution.

6.8.1 Data

6.8.2 Atomic Structure Files

pypospack uses a a custom class SimulationCell which describes atomic positions as a representative unit volume, bounded by the three lattice vectors, $\boldsymbol{a}_1$, $\boldsymbol{a}_2$, and $\boldsymbol{a}_3$, where $\boldsymbol{a}_i = [a_{i1}, a_{i2}, a_{i3}]$. These are stored locally by an H-matrix representation,

$$H = [] = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \tag{6–1}$$

6.8.3 Configuration File

One of the goals of *pypospack* is to make it easy for any user to get started quickly

In particular, the software is written to alleviate the potential developer from understanding the complete implementation of all the algorithms for sampling, potential development, calculation of

pypospack

6.8.4 Material System Representation

Currently, *pypospack* uses the POSCAR file format utilized by VASP as the primary input definition of atomic configuration files to represent different types of structures.

In addition, *pypospack* has a compatibiity layer with the atomistic simulation envioronment (ASE). This provides capability with a wide variety of structure files.

6.8.5 Parallelization

Since *pypospack* uses Monte Carlo sampling as the workhorse for estimation, the computational effort can be parallelized over the parameter space being searched. Concurrency is implemented by assigning each processor a different random seed and number of simulations is divded equally amongst the processors. To avoid issues with file access, each rank is given it's own directory and the results are processed by the rank 0 processor at the end of each iteration.

Currency is done though *mpi4py* which provides a python interface to the standard Message Passing Interface (MPI), since the implementation details are taken care of by *mpi4py*, *pypospack* supports the broad range of MPI implementations.

6.8.6 Energy Evaluations

The energy evaluation of *pypospack* is done through an integration layer which has been implemented for both GULP and LAMMPS. The choice to use an external energy calculator rather than implementing one internally was chosen to minimize The success of these approaches is largely driven that epistemic uncertainty associated with DFT calculations are largely driven by the unknown functional form of the exchange correlation functional.

Contrast this to molecular dynamics, where $V(\boldsymbol{r}_{ij})$ has uncertainty in both the functional form and parameterization. Before attacking problems with functional form,

Let us first consider, three fairly simple systems which are widely studied, where the fundamental physics are significantly different. Metal systems are described by

6.8.7 Statistical Sampling

Statistical Sampling

6.8.8 Machine Learning Algorithms

SciKitLearn

In silico approaches to molecular dynamics simulation is largely constrained by long development times required to develop a potential. Molecular dyanamics simulations are largely dominated

Within the ICME approach, different levels of theory lead to different methods of simulation which span the quantum mechanical level, currently dominated by *ab initio* techniques such as Density Functional Theory (DFT) to the engineering scale. Between these two scales there are atomistic level Molecular Dynamics/Monte Carlo techniques and meso scale simulations.

Pypospack is written as a library to help bridge the *ab initio* techniques to high si

Error estimation may be monitored through the time-evolution of the ensemble average. However, this is a necessary, but not sufficient condition for convergence, because plateaus of the ensemble average often conceal anomalous overlap of the density of stats characterizing the initial and final states. The latter should be the key criterion to ascertain the local convergence of the simulation for those degrees of freedom that are effectively sampled.

Statistical errors in FEP calculations may be estimated by means of a first-order expansion of free energy, which involves an estimation of the sampling ratio of the latter of the calculation (Straatsma, 1986)

$$A = U - TS + \sum \mu_i N_i \tag{6–2}$$

$$G = U + PV - TS + \sum \mu_i N_i = H - TS + \sum \mu_i N_i \tag{6–3}$$

This software is a collection of software libaries which can be scripted together to evaluate software.

Weight free approaches to developing potentials, such as genetic algorithms, exist and use the concept of Pareto efficiency as metric in which to efficiency of a potential.

Pypospack helps to resolve the problems for potential developers by eliminating the requirements for developing code for MPI and complex queueing systems. Instead functionality is developed by exposing high-level APIs to potential developers, while the implementation by lower level APIs is done by the software itself.

This allows potential developers to focus on the creation of a testing database, to define reference structure property relationships. Pypospack then uses an evolutionary algorithm based upon evolving a probability distribution which describes the density of Pareto optimal points in the parameter set.

A series of reusable analytical tools are also developed for the purposes of ad hoc data analysis, and automated data analysis, which aids the potential developer to select potentials ex-post, and then to test these ensembles of potentials against other structure property relationships which are more expensive.

Since the description of the solution of potentials is represented as a probability distribution, the package can be adopted to Bayesian inference techniques which will enable UQ on predictions which represent the propagation of uncertainty to potential development.

Finally, this package also provides some tools to begin to tackle problems of model form uncertainty, transferribility, etc.

This project required the development of software for the development of Pareto frontier.

6.9 Representation of Atomic Structures

6.10 Quantities of Interest

Let us start our discussion of the calculation of point defects by starting reviewing the notion of Kröger-Vink notation[80]

6.11 Tasks

6.12 Implementation of the OpenKIM API

In order to support the largest number of classical potentials, software was written in python using object oriented techniques so that new types of simulations, new potentials, new quantities of interest, and new simulation software.

6.13 Possible Scalability Issues

At the current time, pypospack built upon the scientific python stack. Evaluation of interatomic potentials. This software subprocesses either serial version of LAMMPS or GULP to calculate properties of interest. Parallelization is batched processed across iterations, which each processor rank being given a unique directory space to prevent IO conflicts.

CHAPTER 7
APPLICATIONS TO IONIC SYSTEMS

Lewis Catlow potential[81]

Henkelman *et al*[82]

Two potentials unpublished Buckingham potentials were developed by Ball and Grimes (BG1 and BG2), but were used in the work of Henkelman *et al*[82].

7.1 Potential Formalism

With every potential formalism, the transformation of atomic positions to suitable descriptors. Directly available descriptors of atomic positions are not invariant with respect to translation and rotation of the system. The energy of a configuration is the summation of the pair-wise contributions between the atoms, and the potential is a function is a function of the interatomic distance between the two atoms.

$$\begin{aligned}\hat{V}_(\boldsymbol{R}) &= \sum_{i<j} V_{ij}(\boldsymbol{r}_i, \boldsymbol{r}_j) \\ &= \sum_{i<j} V_{ij}(\boldsymbol{r_i} - \boldsymbol{r_j}) \\ &= \sum_{i<j} V_{ij}(r_{ij})\end{aligned}$$

Lewis and Catlow[81] parameterized a wide range of oxide systems with a pairwise potential, between two atoms i and j. In this formalism as described by Catlow[83], the Buckingham potential[84] is combined with a Coulombic interaction,

$$V_{ij}(r_{ij}) = \frac{Z_i Z_j}{r_{ij}} + A_{ij}\exp(-r_{ij}/\rho_{ij}) - \frac{C_{ij}}{r_{ij}^6} \tag{7–1}$$

The first term is a Coulombic interaction dependent on the the point charges, Z_i and Z_j. The remaining compoentss are components of the Buckingham Potential which combines the repulsion between two ions due to the Pauli exclusion principle combined a term for a weak van der Waals interaction.

7.2 Incorporation of Prior Knowledge

In a binary oxide, there are the three pairwise interactions in which to consider: the anion-anion interaction, the cation-cation interaction, and the cation-cation interaction. Lewis and Catlow make simplifying assumptions, which are replicated in the development of a potential for MgO in this chapter.

Lewis and Catlow make simplifying assumptions[81, 83**?**], which are replicated in the development of a potential for MgO in this chapter.

1. The material is charge neutral
2. Cations only act with each other through Coulomb interaction.
3. The anion-cation is considered to be the Born-Mayer potential.
4. The anion-anion interaction

In the Lewis Catlow (LC) parameterization, Additionally, the cations only interact with each other though the Coulomb interaction.

The cation-cation interaction is considered to be completely coulombic

The anion-cation interaction is considered to be the Born-Mayer potential,

$$V_{+-} = A_{+=} \exp(-r_{ij}), \tag{7–2}$$

where the attractive r^{-6} term is lost.

7.3 Target Properties and Prior Knowledge

We choose the target materials properties for MgO to be the lattice parameter, a_0, and elastic properties (c_{11},c_{12},c_{44}, and shear and bulk, B and G) of the ground-state rock-salt (NaCl) structure, the (100) surface energy, the anion and cation Frenkel defect energies, and the Schottky defect formation energy. Although B and G are fully determined by other elastic constants, it is useful to include them in the fitting database as they are the most physically accessible and measurable the specific sums and differences in elastic properties that correspond to criteria for the stability of the crystal ($B > 0$, $G > 0$, $c_{44} > 0$).

Reference values for all of the targeted properties were calculated with VASP[34–36] using the Perdew, Burke, and Ernzerhof generalized-gradient approximation (PBE-GGA) of the exchange-correlation functional [48], [49]. The lattice parameter and elastic properties were calculated from an 8-ion conventional cubic unit cell using a 6x6x6 k-point mesh. Surface energies were calculated using a 1x1x10 supercell, with a slab thickness consisting of half of the height of the cell, and a k-point mesh of 9x9x1. Defect formation energies were calculated from 3x3x3 supercells, using a k-point mesh of 3x3x3. The plane wave cutoff energy was set at 800 eV. The resulting values of these quantities are shown in Table 1 and constitute the QOIs for the parametrization process.

CHAPTER 8
APPLICATIONS TO NICKEL EMBEDDED ATOM POTENTIALS

The aim of this chapter is to apply the earlier described methods and tools involved in developing potentials for metallic systems. The defining difference between metals and nonmetals is the lack of an energy gap between ground and excited states. Metallic systems are normally characterized as a system existing within a sea of delocalized electrons.

8.1 Insights from Quantum Mechancial Techniques

In the development of the Finnis-Sinclair method, the second-moment tight-binding(3)

8.2 Embedded Atom Model

For metallic systems, the most common potential forms are those of the Finnis-Sinclar method[56] and the embedded atom model (EAM) [52, 53]. Both of these approaches have similar formalisms which combine a pair-potential function and an energy functional dependent upon the electron density contributions from an atoms neighbors with the total energy of the system V being described as

$$V = \sum_{i<j} \phi_{s_i s_j}(r_{ij}) + \sum_i F_{s_i}(\bar{\rho}_i) \tag{8–1}$$

The first term is the summation over all neighbors of a pair potential of a pair potential energy(ϕ_{ij}) between two atoms, i and j, with the chemical species, s_i and s_j. The second term is the embedding energy (F_i) necessary to place the atom i at its position in an electron gas density ($\bar{\phi}_i$) influenced by all the neighbors of the atom. In EAM, the host electron density is given by the sum of the contributions ρ_i is represented by the sum of the contributions $\rho_{s_j}(r)$ from all neighboring atoms j.

$$\bar{\rho}_i = \sum_{j \neq i} \rho_{s_j}(r_{ij}) \tag{8–2}$$

In the Finnis-Sinclair potential, $F_{s_i}(\bar{\rho}) = -\sqrt{\bar{\rho}_i}$

8.3 Approaches to EAM Potential Development

The original formulation of the EAM represented the reepulsion between the atomic cores (nuclei and inner electron shells), represented by a power law or an Born-Mayer type exponential function. In these cases, the pair potential, ϕ, is purely repulsive, while the embedding energy F was attractive. Later EAM pair potentials used Morse Functions[10,19,20], which has a distinct energy well, while the embedding function in these cases serves as a corrective term.

In either case, the electron density function and the pair potential are given as analytic functions of the radial separation of two atoms r_{ij} with fitted parameters. The embedding function is sometimes specified as having a specific functional form, or determined by fitting the embedding function to an equation of state such as Rose *et al*[21].

For Nickel, Johnson and Oh[85], Voter-Chen[19,23], Angelo *et al*[20] have used parametric functional forms, where Ercolessi and Adams[24] and Mishin *25* implemented cubic-spline approaches, which do not specify an analytical functional form.

In early potential development, the parameters are adjusted to match properties such cohesive energy, lattice parameters, elastic properties, and vacancy formation energies. With additional computational power, stacking fault energies, phase order differences, and surface energies to fit to important relevant environments thought to be important to produce transferrable potentials.

8.4 Development of an EAM potential

Even when a potential has an analytical functional form. EAM potentials are specified in a functional form which is dependent a specific code format which precalculates evaluation of the $\phi(r_{ij})$, $\rho(r_{ij})$, and $F(\bar{\rho})$. Specifics of this calculation are provided in Appendex **??**.

8.5 Cutoff function

8.6 Generalized Stacking Fault in FCC

The generalized stacking fault in FCC metals describe the the slip of {111} planes of the face centered cubic cell in the ¡112¿ direction, which represents the energy In the early works of Frenkel and Mackenzie describe this motion as a function of the macroscopically measured shear modulus, the Burgers vector and the interplanar spacing of the {111} planes.

Rice[3] unstable stacking faults and stable stacking faults

Vitek[86, 87] develops the notion of the generalized stacking fault, which cannot be measured experimental except at a single point knows as the intrinsic stacking fault γ_{ISF}

Calculation of unstable stacking faults[88–90] was done with EAM potentials, while DFT has been used for the calcualtion of the GSF curve [14,15,16]

Zimmerman takes the approach of creating a simulation of the FCC crystal oriented in the ¡111¿ direction, with the basal plane formed by the ¡112¿ and the ¡111¿ direction. For the ease of creating the stacking fault, the ¡112¿ direction is chosen on the x-axis, and the ¡110¿ is chosen for the y-axis, and the z-axis on the ¡111¿. To create the stacking fault, the lower half remains fixed, while the upperhalf is displaced in the ¡112¿ direction in small increments. Zimmerman uses 6000 atoms consisting of 30 111 planes. After lateral displacement, the atoms are allowed to relax laterally (in the ¡111¿ direction) except for three 111 planes at the top and bottom. This method is used in [11-13]

To create a surrogate model, we use the atomic simulation enviornment (ASE)

8.6.1 Density Functional Theory

8.6.2 Molecular Dynamics

First generate a simulation cell of a representative fcc bulk with $[1\,1\,2]$ in the x direction, $[\bar{1}\,1\,0]$ in the y direction, and $[\bar{1}\,\bar{1}\,1]$ in the z-direction, which was adapted from the script of Spear[91].

CHAPTER 9
APPLICATIONS TO COVALENTLY BONDED MATERIALS

9.1 Potential Formalism

The Stillinger-Weber potential[60] is a combination of a two-body (ϕ_2) and three-body (ϕ_3) terms, which is a function of the interatomic distances (r_{ij},r_{ik}) from a central atom i and the angle (θ_{ijk}).

$$E = \sum_{i<j} \varepsilon\phi_2(r_{ij}) + \sum_{j<k} \varepsilon\phi_3(r_{ij}, r_{ik}, \theta_{ijk}) \tag{9–1}$$

where

$$\phi_2(r_{ij}) = A_{ij}\left[B_{ij}\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{p_{ij}} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{q_{ij}}\right]\exp\left(\frac{\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}}\right) \tag{9–2}$$

$$\phi_3(r_{ij}, r_{ik}, \theta_{ijk}) = \lambda_{ijk}\epsilon_{ijk}\left[\cos(\theta_{ijk}) - \cos(\theta_{0,ijk})\right]\exp\left(\frac{\gamma_{ij}\sigma_{ij}}{r_{ij} - a_{ij}\sigma_{ij}}\right)\exp\left(\frac{\gamma_{ik}\sigma_{ik}}{r_{ik} - a_{ik}\sigma_{ik}}\right) \tag{9–3}$$

The A, B, p, and q parameters only apply to the two-body interactions. The *lambda*, $theta_0$ parameters are used only for three-body interactions. The ε,σ, and a parameters shared between the terms.

In order to compare the performance of the potentials developing using this software package, we use the original parameterization of Stillinger and Weber (SW)[60], Vink *et al*(VMWM)[92], and Pizzagalli *et al*[2]

9.2 Methodology

Here we repeat the process outlines in Chapter.

For the development of a new potential, we use a subset of the reference values from Pizzagalli *et al* [2].

Table 9-1. Table of parameters for the reference potentials, and the lower and upper bounds used to define the uniform distribution

Parameter	SW	VBWM	PG
ϵ	2.1686	1.64833	1.04190
σ	2.0951	2.0951	2.128117
a	1.80	1.80	1.80
λ	21.0	31.5	31.0
γ	1.20	1.20	1.1
A	7.049556277	7.049556277	19.0
B	0.602224558	0.6022245584	0.65
p	4.0	4.0	3.5
q	0.0	0.0	0.0

Table 9-2. Fitting database for Si, reference values taken from Pizzagalli *et al*[2]

Property	Units	Value
E_c	eV	-4.63
a_0	Å	5.43
C_{11}	GPa	166
C_{12}	GPa	64
C_{44}	GPa	80
B	GPa	99
E_v	eV	3.6

9.3 Results and Discussion

9.3.1 Analysis of Prediction Performance

9.3.1.1 Univariate QOI analysis

To analyze the performance of our ensemble of potentials, an univariate inspection of the probability density functions for each of the QOIs can provide insight on the performance of the potential formalism. Figure 9-1 shows the evolution of the predicted values of the bulk modulus for Silicon, B. The typical evolution on how these parameter optimization process evolves an ensemble of potentials. These curves are the probability density function, estimate using a kernel density estimate with the Silverman bandwidth estimation. In early iterations, the uncertainty reduction in predictions is quite large, but in later iterations the estimates for the Pareto optimal ensemble improves, the probability density function increases around the target value.

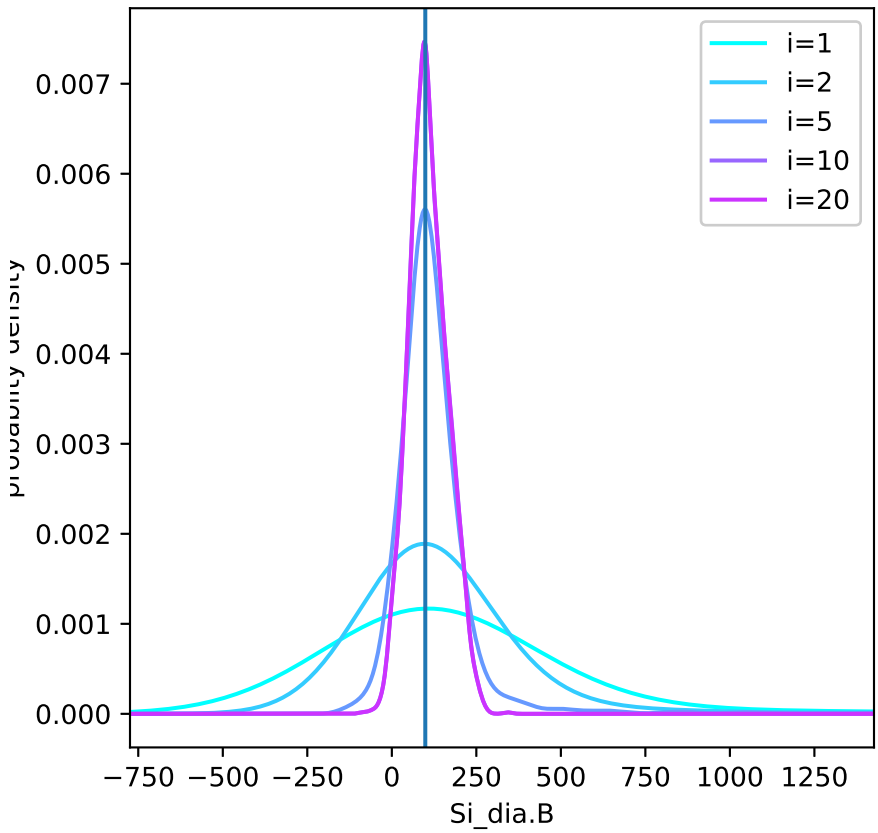

Figure 9-1. Evolution of the prediction of the bulk modulus, B, for Silicon.

In a scalar optimization approach, the existance of local minima make the identification of the optimal parameterization as local minima becomes a basin of attraction in a gradient descent approach. Even when global optimization techniques are used, the potential developer is not aware if the existence of local minima and cannot evaluate the region of these parameterizations, would be an area of interest. In contrast, the Pareto optimization approach does not look for a single optimal parameterization but a ensemble of candidate parameters, which product Pareto optimal predctions. Instead, these regions become basins of attractions for candidate potentials, which manifests itself in regions of elevated probability. for identifying candidate parameters is the presence of multiple

regions of parameterization becomes apparent upon visual inspection of the distribution of QOI predctions.

In Figure 9-2, the probability density plot for the predictions of the lattice parameter, a_0 shows a bimodal distribution; the candidate potentials have two peaks which indicate an elevated probability distribution due a high concentration of predictions in those regions. Simular to the evolution of predictions for the bulk modulus, initial iterations have a broad proability distribution which indicates a high level of uncertainty, while the later iterations concentrate the predictions. In this situation, one peak centered around the target value of $a_0 = 5.43$ Å. The second peak becomes more pronounced in later iterations indicate a second population of potentials which predict a lattice parameter of 5.0 Å. While potential developers are interested in getting the lattice parameters correct, the second peaks contains potentials which are not dominated by the potentials in the first peak, and must have better predictions in other material properties.

Prediction for the cohesive energy E_c of the system is the material property which shows an atypical distribution; Figure 9-3 shows the evolution of those predictions. In early iterations ($i = 1$ and $i = 2$), the probability density functiosn resemble a normal distribution with peaks approximately -4.0 eV/Å, which is significantly higher than the target value of -4.63 eV/Å. The candidate parameterizations continue to concentrate their probability density into a smaller region, but does so in an unexpected way. The tail end probabilities continue to reduce markedly, but by the fifth iteration ($i = 5$), a probability density function takes a significantly different shape. Instead of having a peak, there is a region of elevated probability that is relatively constant over the range $-4.5\mathring{A} < E_c < -3.6\mathring{A}$. At the final iteration ($i = 20$), a clear mode develops at $E_c = -4.3$ Åwith a shoulder at $E_c = -3.6$ Å. The distribution of potentials conveys important information to the potential developer; the majority of the potentials predicting a cohesive energy larger than the target values means that choosing to get the cohesive

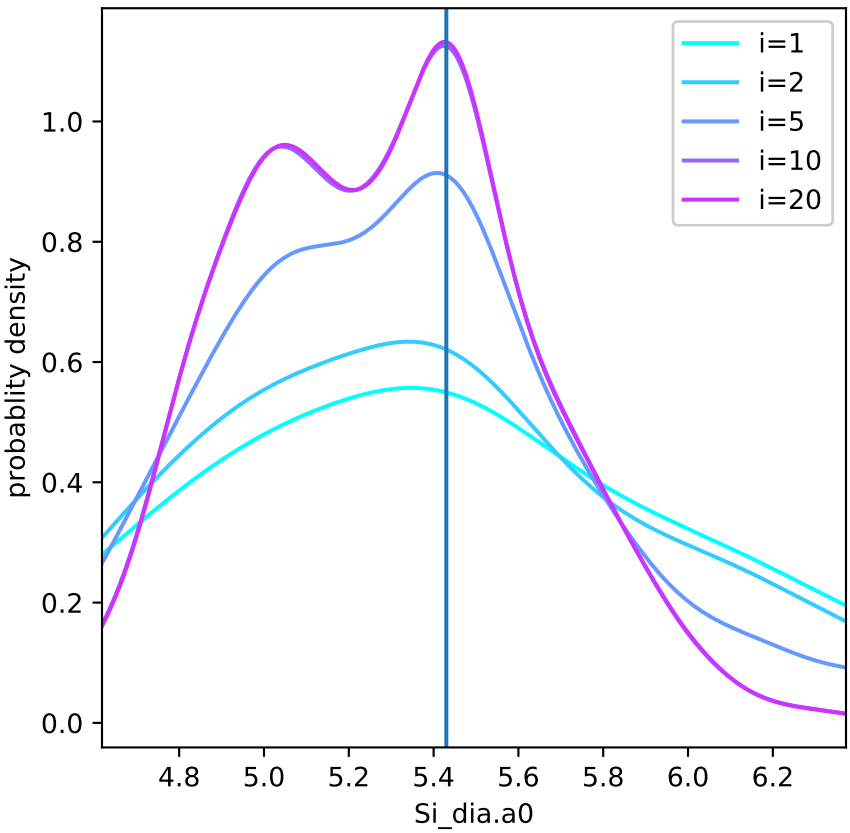

Figure 9-2. Evolution of the the prediction of the lattice parameter, a_0, for Si.

energy corrent likely means markedly lower fidelity in predicting the other target material properties.

9.3.1.2 Bivariate QOI objective analysis

In moving towards the selection of a potential in a multi-objective space, it is useful examine the problem through a series of bi-objective plots.

9.3.1.3 Multivariate QOI objective analysis

9.3.2 Analysis of parameter space

9.3.3 Selection of Potentials

Both the Akaike information criterion (AIC)[93] and the Bayesian information criterion (BIC) [94] are estimators of the relative quality of statistical models for a given

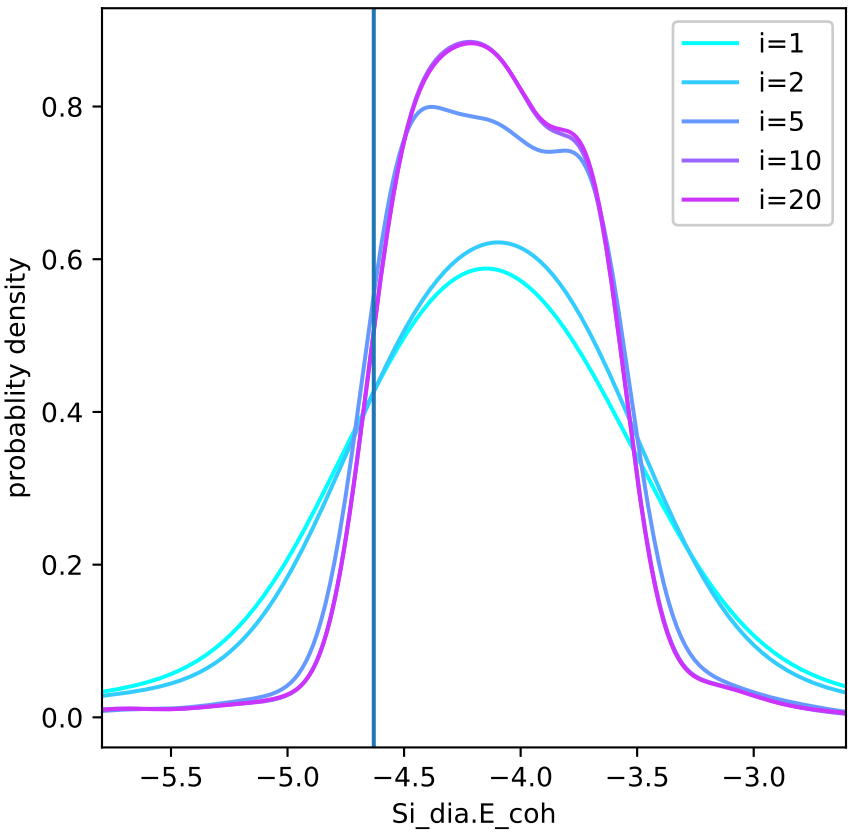

Figure 9-3. Evolution of the prediction of the cohesive energy, E_c for Si.

set of data. Both of these criteria are penalized for using a more complicated model. The AIC or BIC of a model is written in the form $[2\log(\hat{L} + kp]$, where $\hat{L}$ is a likelihood function, p is the number of parameters in the model, and k is 2 for AIC and $\log(n)$ for BIC. Given a collection of models for data, bot h the AIC and BIC estimates the quality of each model relative to the other models, providing a mechanism for model selection. ddhe Akaike information criterion (AIC) [93] is an estimator of the relative quality of statistical models for a given set of data. Given a collections of models for the data, AIC estimates the quality of each model relative to other models, providing a mechanism for model selection.

9.3.4 Validation of Potentials

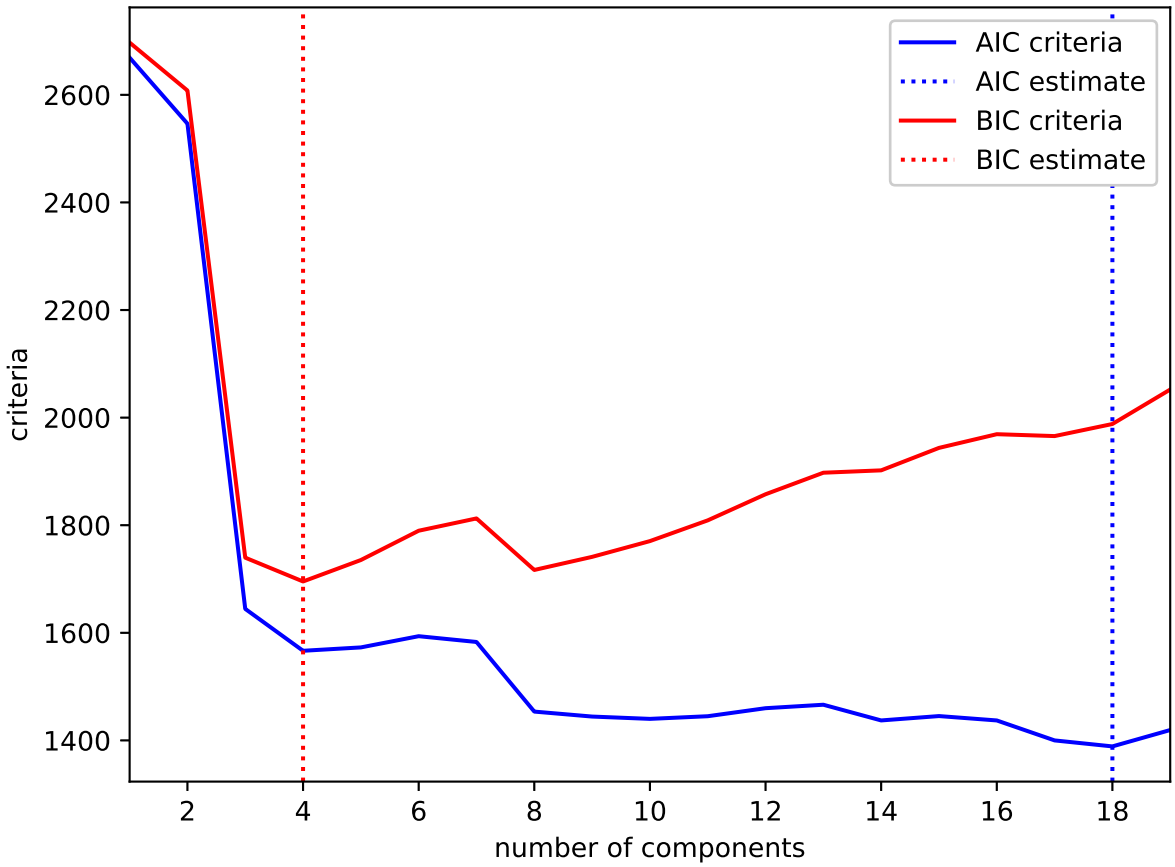

Figure 9-4. Identification of the number of

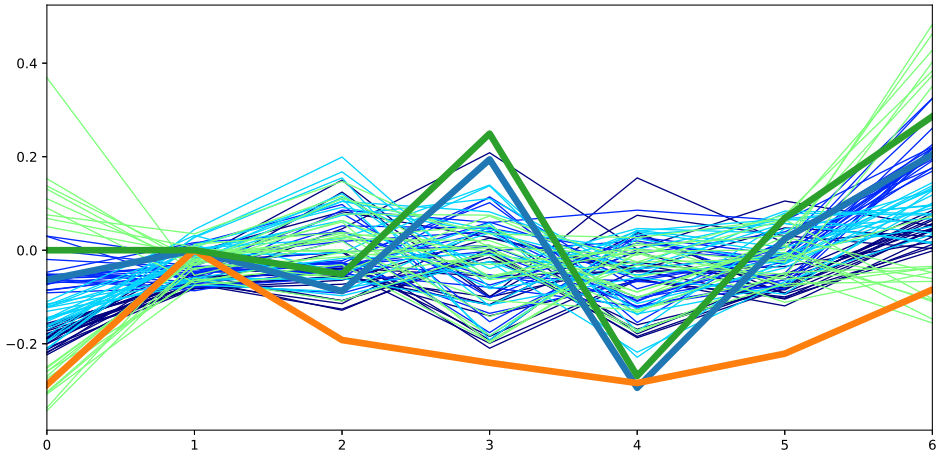

Figure 9-5. Parallel plot

REFERENCES

[1] B. W. Silverman, "Density estimation for statistics and data analysis," 1986.

[2] L. Pizzagalli, J. Godet, J. Guénolé, S. Brochard, E. Holmstrom, K. Nordlund, and T. Albaret, "A new parametrization of the stillinger–weber potential for an improved description of defects and plasticity of silicon," *J. Phys. Condens. Matter*, vol. 25, no. 5, p. 055801, jan 2013. [Online]. Available: https://doi.org/10.1088%2F0953-8984%2F25%2F5%2F055801

[3] D. R. Hartree, "The wave mechanics of an atom with a non-coulomb central field. part i. theory and methods," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 24, no. 1, p. 89110, 1928.

[4] J. C. Slater, "The self consistent field and the structure of atoms," *Phys. Rev.*, vol. 32, pp. 339–348, Sep 1928. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.32.339

[5] J. A. Gaunt, "A theory of hartree's atomic fields," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 24, no. 2, p. 328342, 1928.

[6] J. C. Slater, "Note on hartree's method," *Phys. Rev.*, vol. 35, pp. 210–211, Jan 1930. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.35.210.2

[7] V. Fock, "selfconsistent field" mit austausch für natrium," *Zeitschrift für Physik*, vol. 62, no. 11, pp. 795–805, Nov 1930. [Online]. Available: https://doi.org/10.1007/BF01330439

[8] P. Hohenberg and W. Kohn, "Inhomogeneous electron gas," *Phys. Rev.*, vol. 136, pp. B864–B871, Nov 1964. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.136.B864

[9] W. Kohn and L. J. Sham, "Self-consistent equations including exchange and correlation effects," *Phys. Rev.*, vol. 140, pp. A1133–A1138, Nov 1965. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.140.A1133

[10] D. M. Ceperley and B. J. Alder, "Ground state of the electron gas by a stochastic method," *Phys. Rev. Lett.*, vol. 45, pp. 566–569, Aug 1980. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.45.566

[11] S. H. Vosko, L. Wilk, and M. Nusair, "Accurate spin-dependent electron liquid correlation energies for local spin density calculations: a critical analysis," *Canadian Journal of Physics*, vol. 58, no. 8, pp. 1200–1211, 1980. [Online]. Available: https://doi.org/10.1139/p80-159

[12] J. P. Perdew and A. Zunger, "Self-interaction correction to density-functional approximations for many-electron systems," *Phys. Rev. B*, vol. 23, pp. 5048–5079, May 1981. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.23.5048

[13] J. P. Perdew and Y. Wang, "Accurate and simple analytic representation of the electron-gas correlation energy," *Phys. Rev. B*, vol. 45, pp. 13 244–13 249, Jun 1992. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.45.13244

[14] D. C. Langreth and M. J. Mehl, "Beyond the local-density approximation in calculations of ground-state electronic properties," *Phys. Rev. B*, vol. 28, pp. 1809–1834, Aug 1983. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.28.1809

[15] A. D. Becke, "Density-functional exchange-energy approximation with correct asymptotic behavior," *Phys. Rev. A*, vol. 38, pp. 3098–3100, Sep 1988. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.38.3098

[16] R. LeSar, *Introduction to computational materials science: fundamentals to applications*. Cambridge University Press, 2013.

[17] J. E. Lennard-Jones, "On the determination of molecular fields.i. from the variation of the viscosity of a gas with temperature," *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 106, no. 738, pp. 441–462, 1924. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1924.0082

[18] F. Ercolessi and J. B. Adams, "Interatomic potentials from first-principles calculations: The force-matching method," *Europhysics Letters (EPL)*, vol. 26, no. 8, pp. 583–588, jun 1994. [Online]. Available: https://doi.org/10.1209%2F0295-5075%2F26%2F8%2F005

[19] M. C. Payne, I. J. Robertson, D. Thomson, and V. Heine, "Ab initio databases for fitting and testing interatomic potentials," *Philosophical Magazine B*, vol. 73, no. 1, pp. 191–199, 1996. [Online]. Available: https://doi.org/10.1080/13642819608239124

[20] W. Karush, "Minima of functions of several variables with inequalities as side constraints," Master's thesis, University of Chicago, 1939.

[21] H. W. Kuhn and A. W. Tucker, "Nonlinear programming," in *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*. Berkeley, Calif.: University of California Press, 1951, pp. 481–492.

[22] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.

[23] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Reliability Engineering & System Safety*, vol. 91, no. 9, pp. 992 – 1007, 2006, special Issue - Genetic Algorithms and Reliability. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0951832005002012

[24] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[25] R. Marler and J. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, Apr 2004. [Online]. Available: https://doi.org/10.1007/s00158-003-0368-6

[26] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. Ann Arbor: University of Michigan Press, 1975.

[27] D. E. Goldberg, J. Richardson *et al.*, "Genetic algorithms with sharing for multimodal function optimization," in *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 41–49.

[28] C. M. Fonseca and P. J. Fleming, "Multiobjective genetic algorithms," in *Genetic algorithms for control systems engineering, IEE colloquium on*. IET, 1993, pp. 6–1.

[29] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 03 1951. [Online]. Available: https://doi.org/10.1214/aoms/1177729694

[30] J. R. Hershey and P. A. Olsen, "Approximating the kullback leibler divergence between gaussian mixture models," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 4, April 2007, pp. IV–317–IV–320.

[31] G. V. Lewis and C. R. A. Catlow, "Potential models for ionic oxides," *Journal of Physics C: Solid State Physics*, vol. 18, no. 6, p. 1149, 1985. [Online]. Available: http://stacks.iop.org/0022-3719/18/i=6/a=010

[32] W. Rudin, *Real and complex analysis*. McGraw-Hill, Inc., 1987.

[33] K. L. Chung, *A course in probability theory*. Academic press, 2001.

[34] G. Kresse and J. Hafner, "Ab initio molecular dynamics for liquid metals," *Phys. Rev. B*, vol. 47, pp. 558–561, Jan 1993. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.47.558

[35] G. Kresse and J. Furthmüller, "Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set," *Phys. Rev. B*, vol. 54, pp. 11 169–11 186, Oct 1996. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.54.11169

[36] ——, "Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set," *Computational Materials Science*, vol. 6, no. 1, pp. 15 – 50, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0927025696000080

[37] X. Gonze, J.-M. Beuken, R. Caracas, F. Detraux, M. Fuchs, G.-M. Rignanese, L. Sindic, M. Verstraete, G. Zerah, F. Jollet, M. Torrent, A. Roy, M. Mikami, P. Ghosez, J.-Y. Raty, and D. Allan, "First-principles computation of material properties: The ABINIT

software project," *Computational Materials Science*, vol. 25, no. 3, pp. 478–492, November 2002. [Online]. Available: https://doi.org/10.1016/s0927-0256(02)00325-7

[38] X. Gonze, G.-M. Rignanese, M. Verstraete, J.-M. Beuken, Y. Pouillon, R. Caracas, F. Jollet, M. Torrent, G. Zerah, M. Mikami, P. Ghosez, M. Veithen, J.-Y. Raty, V. Olevano, F. Bruneval, L. Reining, R. Godby, G. Onida, and D. H. D.C. Allan, "A brief introduction to the ABINIT software package," *Zeitschrift fr Kristallographie - Crystalline Materials*, vol. 220, no. , pp. 558–562, January 2005. [Online]. Available: http://dx.doi.org/10.1524/zkri.220.5.558.65066

[39] X. Gonze, B. Amadon, P.-M. Anglade, J.-M. Beuken, F. Bottin, P. Boulanger, F. Bruneval, D. Caliste, R. Caracas, M. Ct, T. Deutsch, L. Genovese, P. Ghosez, M. Giantomassi, S. Goedecker, D. Hamann, P. Hermet, F. Jollet, G. Jomard, S. Leroux, M. Mancini, S. Mazevet, M. Oliveira, G. Onida, Y. Pouillon, T. Rangel, G.-M. Rignanese, D. Sangalli, R. Shaltaf, M. Torrent, M. Verstraete, G. Zerah, and J. Zwanziger, "ABINIT: First-principles approach to material and nanosystem properties," *Comput. Phys. Commun.*, vol. 180, no. 12, pp. 2582–2615, December 2009. [Online]. Available: https://doi.org/10.1016/j.cpc.2009.07.007

[40] X. Gonze, F. Jollet, F. Abreu Araujo, D. Adams, B. Amadon, T. Applencourt, C. Audouze, J.-M. Beuken, J. Bieder, A. Bokhanchuk, E. Bousquet, F. Bruneval, D. Caliste, M. Ct, F. Dahm, F. Da Pieve, M. Delaveau, M. Di Gennaro, B. Dorado, C. Espejo, G. Geneste, L. Genovese, A. Gerossier, M. Giantomassi, Y. Gillet, D. Hamann, L. He, G. Jomard, J. Laflamme Janssen, S. Le Roux, A. Levitt, A. Lherbier, F. Liu, I. Lukaevi, A. Martin, C. Martins, M. Oliveira, S. Ponc, Y. Pouillon, T. Rangel, G.-M. Rignanese, A. Romero, B. Rousseau, O. Rubel, A. Shukri, M. Stankovski, M. Torrent, M. Van Setten, B. Van Troeye, M. Verstraete, D. Waroquiers, J. Wiktor, B. Xu, A. Zhou, and J. Zwanziger, "Recent developments in the ABINIT software package," *Comput. Phys. Commun.*, vol. 205, pp. 106–131, August 2016. [Online]. Available: https://doi.org/10.1016/j.cpc.2016.04.003

[41] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. D. Corso, S. de Gironcoli, S. Fabris, G. Fratesi, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch, "QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials," *Journal of Physics: Condensed Matter*, vol. 21, no. 39, p. 395502, sep 2009. [Online]. Available: https://doi.org/10.1088%2F0953-8984%2F21%2F39%2F395502

[42] S. Curtarolo, W. Setyawan, G. L. Hart, M. Jahnatek, R. V. Chepulskii, R. H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M. J. Mehl, H. T. Stokes, D. O. Demchenko, and D. Morgan, "Aflow: An automatic framework for high-throughput materials discovery," *Computational Materials Science*, vol. 58, pp. 218 – 226, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0927025612000717

[43] S. P. Ong, W. D. Richards, A. Jain, G. Hautier, M. Kocher, S. Cholia, D. Gunter, V. L. Chevrier, K. A. Persson, and G. Ceder, "Python materials genomics (pymatgen): A robust, open-source python library for materials analysis," *Computational Materials Science*, vol. 68, pp. 314 – 319, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0927025612006295

[44] S. Plimpton, "Fast parallel algorithms for short-range molecular dynamics," *Journal of Computational Physics*, vol. 117, no. 1, pp. 1 – 19, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002199918571039X

[45] J. D. Gale and A. L. Rohl, "The general utility lattice program (gulp)," *Molecular Simulation*, vol. 29, no. 5, pp. 291–341, 2003. [Online]. Available: https://doi.org/10.1080/0892702031000104887

[46] J. A. Martinez, D. E. Yilmaz, T. Liang, S. B. Sinnott, and S. R. Phillpot, "Fitting empirical potentials: Challenges and methodologies," *Current Opinion in Solid State and Materials Science*, vol. 17, no. 6, pp. 263 – 270, 2013, frontiers in Methods for Materials Simulations. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359028613000727

[47] J. A. Martinez, A. Chernatynskiy, D. E. Yilmaz, T. Liang, S. B. Sinnott, and S. R. Phillpot, "Potential optimization software for materials (posmat)," *Computer Physics Communications*, vol. 203, pp. 201 – 211, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0010465516300042

[48] M. P. Allen and D. J. Tildesley, *Computer simulation of liquids*. Oxford, United Kingdom: Oxford university press, 1987.

[49] J. M. Haile, *Molecular dynamics simulation: Elementary Methods*. New York, NY, USA: Wiley-interscience, 1997.

[50] R. LeSar, *Introduction to Computational Materials Science*. Cambridge, United Kingdom: Cambridge University Press, 2013.

[51] D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Applications*. San Diego, CA, USA: Academic Press, 2002.

[52] M. S. Daw and M. I. Baskes, "Semiempirical, quantum mechanical calculation of hydrogen embrittlement in metals," *Phys. Rev. Lett.*, vol. 50, pp. 1285–1288, Apr 1983. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.50.1285

[53] ——, "Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals," *Phys. Rev. B*, vol. 29, pp. 6443–6453, Jun 1984. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.29.6443

[54] M. S. Daw, S. M. Foiles, and M. I. Baskes, "The embedded-atom method: a review of theory and applications," *Materials Science Reports*,

vol. 9, no. 7, pp. 251 – 310, 1993. [Online]. Available: http://www.sciencedirect.com/science/article/pii/092023079390001U

[55] S. M. Foiles and M. I. Baskes, "Contributions of the embedded-atom method to materials science and engineering," *MRS Bulletin*, vol. 37, no. 5, p. 485491, 2012.

[56] M. W. Finnis and J. E. Sinclair, "A simple empirical n-body potential for transition metals," *Philosophical Magazine A*, vol. 50, no. 1, pp. 45–55, 1984. [Online]. Available: https://doi.org/10.1080/01418618408244210

[57] D. W. Brenner, "Relationship between the embedded-atom method and tersoff potentials," *Phys. Rev. Lett.*, vol. 63, pp. 1022–1022, Aug 1989. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.63.1022

[58] D. W. Brenner, O. A. Shenderova, J. A. Harrison, S. J. Stuart, B. Ni, and S. B. Sinnott, "A second-generation reactive empirical bond order (REBO) potential energy expression for hydrocarbons," *Journal of Physics: Condensed Matter*, vol. 14, no. 4, pp. 783–802, jan 2002. [Online]. Available: https://doi.org/10.1088%2F0953-8984%2F14%2F4%2F312

[59] J. Tersoff, "New empirical approach for the structure and energy of covalent systems," *Phys. Rev. B*, vol. 37, pp. 6991–7000, Apr 1988. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.37.6991

[60] F. H. Stillinger and T. A. Weber, "Computer simulation of local order in condensed phases of silicon," *Phys. Rev. B*, vol. 31, pp. 5262–5271, Apr 1985. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevB.31.5262

[61] G. V. Lewis and C. R. A. Catlow, "Potential models for ionic oxides," *Journal of Physics C: Solid State Physics*, vol. 18, no. 6, pp. 1149–1161, feb 1985. [Online]. Available: https://doi.org/10.1088%2F0022-3719%2F18%2F6%2F010

[62] J. D. Gale, "Empirical potential derivation for ionic materials," *Philosophical Magazine B*, vol. 73, no. 1, pp. 3–19, 1996. [Online]. Available: https://doi.org/10.1080/13642819608239107

[63] A. C. T. van Duin, S. Dasgupta, F. Lorant, and W. A. Goddard, "Reaxff: a reactive force field for hydrocarbons," *The Journal of Physical Chemistry A*, vol. 105, no. 41, pp. 9396–9409, 2001. [Online]. Available: https://doi.org/10.1021/jp004368u

[64] T. P. Senftle, S. Hong, M. M. Islam, S. B. Kylasa, Y. Zheng, Y. K. Shin, C. Junkermeier, R. Engel-Herbert, M. J. Janik, H. M. Aktulga, T. Verstraelen, A. Grama, and A. C. T. van Duin, "The reaxff reactive force-field: development, applications and future directions," *Npj Computational Materials*, vol. 2, 03 2016. [Online]. Available: https://doi.org/10.1038/npjcompumats.2015.11

[65] T. Liang, T.-R. Shan, Y.-T. Cheng, B. D. Devine, M. Noordhoek, Y. Li, Z. Lu, S. R. Phillpot, and S. B. Sinnott, "Classical atomistic

simulations of surfaces and heterogeneous interfaces with the charge-optimized many body (comb) potentials," *Materials Science and Engineering: R: Reports*, vol. 74, no. 9, pp. 255 – 279, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0927796X13000612

[66] T. Liang, Y. K. Shin, Y.-T. Cheng, D. E. Yilmaz, K. G. Vishnu, O. Verners, C. Zou, S. R. Phillpot, S. B. Sinnott, and A. C. van Duin, "Reactive potentials for advanced atomistic simulations," *Annual Review of Materials Research*, vol. 43, no. 1, pp. 109–129, 2013. [Online]. Available: https://doi.org/10.1146/annurev-matsci-071312-121610

[67] D. Brenner, "The art and science of an analytic potential," *physica status solidi (b)*, vol. 217, no. 1, pp. 23–40, 2000. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291521-3951%28200001%29217%3A1%3C23%3A%3AAID-PSSB23%3E3.0.CO%3B2-N

[68] Continuum Analytics, "Anaconda software distribution," *Computer software. Vers.*, pp. 3.0–3.6, 2019. [Online]. Available: https://anaconda.com

[69] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science Engineering*, vol. 9, no. 3, pp. 90–95, May 2007.

[70] P. F. Dubois, "Guest editor's introduction: Python: Batteries included," *Computing in Science Engineering*, vol. 9, no. 3, pp. 7–9, May 2007.

[71] S. v. d. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: a structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.

[72] E. Jones, T. Oliphant, P. Peterson *et al.*, "SciPy: Open source scientific tools for Python," 2001–. [Online]. Available: http://www.scipy.org/

[73] B. L. A. S. T. B. Forum, "Blasic linear algebra subprograms technical (blast) forum standard," Tech. Rep., 2002. [Online]. Available: http://www.netlib.org/blas/blast-forum/blas-report.pdf

[74] E. Anderson, Z. Bai, J. Dongarra, A. Greenbaum, A. McKenney, J. Du Croz, S. Hammarling, J. Demmel, C. Bischof, and D. Sorensen, "Lapack: A portable linear algebra library for high-performance computers," in *Proceedings of the 1990 ACM/IEEE conference on Supercomputing*. IEEE Computer Society Press, 1990, pp. 2–11.

[75] W. McKinney *et al.*, "Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, vol. 445. Austin, TX, 2010, pp. 51–56.

[76] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[77] L. Dalcín, R. Paz, and M. Storti, "Mpi for python," *Journal of Parallel and Distributed Computing*, vol. 65, no. 9, pp. 1108 – 1115, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731505000560

[78] L. Dalcín, R. Paz, M. Storti, and J. D'Elía, "Mpi for python: Performance improvements and mpi-2 extensions," *Journal of Parallel and Distributed Computing*, vol. 68, no. 5, pp. 655 – 662, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0743731507001712

[79] Message Passing Interface Forum, "Mpi: A message-passing interface standard, version 3.1," Tech. Rep., 2015. [Online]. Available: http://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf

[80] F. Kröger and H. Vink, "Relations between the concentrations of imperfections in crystalline solids," ser. Solid State Physics, F. Seitz and D. Turnbull, Eds. Academic Press, 1956, vol. 3, pp. 307 – 435. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0081194708601356

[81] G. Lewis and C. Catlow, "Potential models for ionic oxides," *Journal of Physics C: Solid State Physics*, vol. 18, no. 6, p. 1149, 1985.

[82] G. Henkelman, B. P. Uberuaga, D. J. Harris, J. H. Harding, and N. L. Allan, "Mgo addimer diffusion on mgo (100): a comparison of ab initio and empirical models," *Physical Review B*, vol. 72, no. 11, p. 115437, 2005.

[83] C. R. A. Catlow and J. S. Anderson, "Point defect and electronic properties of uranium dioxide," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 353, no. 1675, pp. 533–561, 1977. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1977.0049

[84] R. A. Buckingham and J. E. Lennard-Jones, "The classical equation of state of gaseous helium, neon and argon," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 168, no. 933, pp. 264–283, 1938. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1938.0173

[85] R. Johnson and D. Oh, "Analytic embedded atom method model for bcc metals," *Journal of Materials Research*, vol. 4, no. 5, pp. 1195–1201, 1989.

[86] V. Vitek, "Thermally activated motion of screw dislocations in bcc metals," *physica status solidi (b)*, vol. 18, no. 2, pp. 687–701, 1966.

[87] ——, "Intrinsic stacking faults in body-centred cubic crystals," *Philosophical Magazine*, vol. 18, no. 154, pp. 773–786, 1968.

[88] Y. Sun, J. R. Rice, and L. Truskinovsky, "Dislocation nucleation versus cleavage in ni 3 ai and ni," *MRS Online Proceedings Library Archive*, vol. 213, 1990.

[89] Y. Sun, G. E. Beltz, and J. R. Rice, "Estimates from atomic models of tension-shear coupling in dislocation nucleation from a crack tip," *Materials Science and Engineering: A*, vol. 170, no. 1-2, pp. 67–85, 1993.

[90] D. Farkas, S. Zhou, C. Vailhe, B. Mutasa, and J. Panova, "Embedded atom calculations of unstable stacking fault energies and surface energies in intermetallics," *Journal of materials research*, vol. 12, no. 1, pp. 93–99, 1997.

[91] P. Spear. Lammps stacking fault energy. [Online]. Available: https://icme.hpc.msstate.edu/mediawiki/index.php/LAMMPS_Stacking_Fault_Energy

[92] R. Vink, G. Barkema, W. van der Weg, and N. Mousseau, "Fitting the stillingerweber potential to amorphous silicon," *J. Non-Cryst. Solids*, vol. 282, no. 2, pp. 248 – 255, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0022309301003428

[93]

[94]

BIOGRAPHICAL SKETCH

Eugene Ragasa was born in Stockton, California, where he attended Saint Mary's High School. He graduated from the United States Military Academy at West Point, New York where he studied mathematics and economics. Upon graduation, he served as an infantry officer in the United States Army for several years before moving to New York City. While living in New York City, he worked as a computer consultant serving a variety of industries such as telecommunications, e-commerce, and finance. He attended Columbia University, New York where he earned a Master's degree in Mathematical Finance, while teaching mathematics at Xavier High School in New York City. He then worked as a quantitative analyst and trader for a variety of trading firms. Additionally, he has been a small business owner at various times owning a computer consultancy business and a trading firm that made markets in a variety of exchange traded products.

After careers in technology and finance, he moved back to his hometown with an interest in applying his skills to science and engineers. He earned a Master's degree in Mechanical Engineering from the University of the Pacific. With an interest in computational simulations, he joined the research group of Simon Phillpot to study atomistic simulations, which he currently does do this day.