CHEMISTRY OF MATERIALS

pubs.acs.org/cm

Methods/Protocols

Computational Modeling of Molecular Mechanics for the Experimentally Inclined

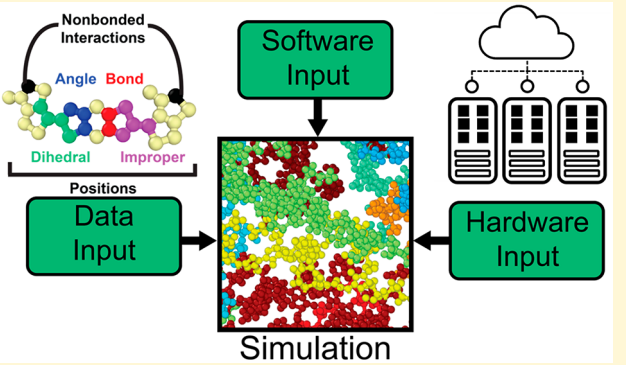
Andrew T. Kleinschmidt, Alexander X. Chen, Tod A. Pascal,* and Darren J. Lipomi*

Cite This: *Chem. Mater.* 2022, 34, 7620−7634

Read Online

ACCESS | Metrics & More | Article Recommendations

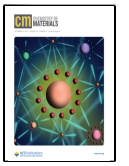
ABSTRACT: Modern computer simulations can provide unique atomic-scale insights into complex material systems, but the method of performing a simulation may seem obscure to the nonspecialist. The goal of this Protocol is to introduce to experimental researchers a description of the tools and methods used in atomic simulations which elucidate the structure, morphology, and dynamics of polymers and nanomaterials. In particular, it focuses on the workflow and logistics of simulations in which the central component is indivisible atoms ("atomistic" as opposed to "quantum" or "continuum" methods). We present methods which describe the positions of atoms, e.g., Monte Carlo (MC) and molecular dynamics (MD) simulations, along with the necessary processes by which simulations are set up, run, and analyzed. However, much of the terminology and workflow outlined in this Protocol is general and thus applies to methods beyond MC and MD as well as other molecular systems (e.g., proteins). This Protocol is separated into three general sections. First, it describes the three types of information that are required for a simulation: a description of the system (i.e., "data file"), instructions for the simulations engine (i.e., "software input file"), and instructions for the hardware, usually supercomputing infrastructure (i.e., "hardware input file"). The data file, generally, describes the initial state of the system as well as a definition of how the atoms within the system interact (usually denoted as a "force field"). Together, these three pieces of information are used to run a simulation, which then produces an output that can be analyzed by the researcher. We hope that this Protocol will provide at least three things for the experimentalist: (1) a frame of reference for the interpretation of computational data, (2) facilitation of collaboration with computational scientists, and (3) the encouragement to perform some computational tasks on their own.



INTRODUCTION

In recent years, improvements in computational hardware (i.e., consequences of Moore's law[1]) and the proliferation of supercomputing centers have greatly increased the accessibility and usefulness of computational research. Coupled with an ever-growing description of the fundamental physics, chemistry, and thermodynamics that govern the behavior of an atomistic system, computational models have become more sophisticated, less retrodictive, and more predictive.[2−6] These factors have increased the importance, ubiquity, and utility of scientific computing in all aspects of research. Nevertheless, the tools used by computational scientists can seem obscure to the experimentalist. Without functional literacy, it is difficult to interpret or judge the merits of a computational study. Similarly, while a researcher might recognize results generated by computational models (e.g., as published in the literature), unfamiliarity with the computational process itself poses a significant barrier to entry for an experimentalist. In this Protocol, we attempt to demystify the process of scientific atomistic computing. Our intended audience is an experimental scientist in the area of materials chemistry. Notably, we do not intend for this work to exhaustively describe the simulation process to a novice computational scientist, for which there is already a wide variety of excellent resources widely available.[7−9] Rather, we focus on the process of what computational scientists do and attempt to give context as to the relative ease or difficulty of certain tasks. We give special attention to organic materials, which reflects the idiosyncrasies of the authors. Most of our discussion is centered around classical atomistic simulations, e.g., molecular dynamics and Monte Carlo simulations, where we do not explicitly consider electronic degrees of freedom. Instead, we describe the energy and forces in the system (i.e., the components and their interactions under study) using smooth analytic potentials (i.e.,

Received: January 28, 2022
Revised: August 12, 2022
Published: August 25, 2022



https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634

force fields). Force fields, in this context, analytically (e.g., mathematically) describe all interactions between components (e.g., atoms or molecules) within a simulation. In other words, a force field mathematically describes what is happening within a system. Simulations using force fields are applicable to a wide variety of molecular materials.[10,11] Likewise, the methods and terminology used in this Protocol are widely applicable to a range of molecular systems, even if specifics differ.

An experimentalist might wonder why they might choose to seek out or perform computational simulations and how computational simulations can be used synergistically with their own research. To summarize briefly, computational simulations are commonly used to predict, verify, and/or elucidate experimental results in a cost-effective and time-effective manner. For example, a chemist might want to synthesize a semiconducting polymer with favorable charge-transport properties and high mechanical compliance for applications in organic electronics. Rather than painstakingly synthesizing several families of polymers, fabricating devices to test their electronic properties, and then characterizing their mechanical properties, the chemist might instead choose to use computational simulations to predict which families of polymers are most likely to have desirable properties. Thus, computational simulations can be used to guide the rational design of new polymeric materials.

Many articles describe the results of simulations. However, it is often difficult for an experimentalist to glean an understanding of the logic and workflow of designing and running a simulation. Understanding the process of running a simulation is critical for both evaluating computational work and applying computational insights in experimental research. The typical workflow of performing a simulation, from start to finish, is outlined in Figure 1. In this process, three types of information

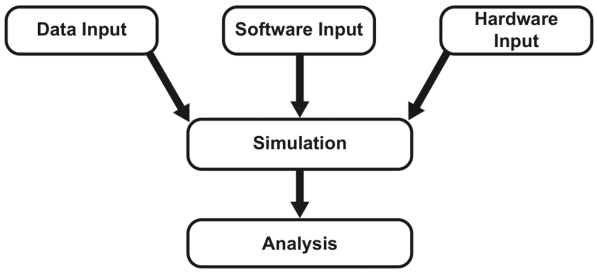


Figure 1. Overview of the types of information required to run a simulation. Researcher begins by creating the data input (or has it embedded within the program input) to inform the program of the initial arrangement and specific interactions of all atoms (Figure 4). Researcher must then create a software input, which uses the data input to begin the simulation on the program and tells the program how to run the simulation (Figure 6a). Hardware input file is also required if specialized hardware is used (i.e., a supercomputer), which informs the hardware how to run the program according to the software input and hardware specifications (Figure 6b). These three tasks converge to carry out the simulation. After the simulation is complete, it is then analyzed by the researcher.

are fed into a program, typically in the form of submission files. These submission files contain instructions designating the initial state of the system (i.e., a description of the components being modeled as well as the interactions between each component), how the software should simulate the system (i.e., what information should be calculated and what method should be used), and how the hardware should execute the program (i.e., a definition of the computational environment). In this Protocol, we arbitrarily refer to these inputs, respectively, with the following nomenclature: "data input," "software input," and "hardware input". The program takes information from the data input to construct the initial configuration of the simulated system. The simulation then propagates the system over time or space. When this simulation is finished, the program will have generated information that can be analyzed in postprocessing (e.g., the equilibrium configuration of the material, thermodynamic properties of the system).

First, the initial starting condition for the simulation (i.e., the data input) must be elaborated. We discuss how to determine and specify an initial condition for the system and what pitfalls must be avoided in doing so. We then describe how the interactions between the different atoms of the system should be included in the model, i.e., the force field.[12] Next, we describe how to create an input for the simulation software (software input). In addition, we make suggestions regarding judicious selection of a simulation software. Likewise, we provide guidance on running and managing the hardware (hardware input). These hardware specifications are often specific to the system being simulated and should be carefully considered to optimize the performance of the program software.

Background. A classical atomistic computer simulation constructs a model in which all of the components (e.g., atoms, ions, and molecules) and interactions (e.g., Coulombic, covalent, van der Waals) in the system (i.e., the interacting components under study) are defined independently. We center our discussion on simulations in which the motions and relative orientations of nuclei are the central focus, namely, molecular dynamics (MD) and Monte Carlo (MC) simulations. While their methodology can differ significantly, the two have similar requirements. Likewise, both MD and MC simulations are often used to supplement experimental studies. Therefore, we use both MD and MC simulations as a framework for introducing how computational simulations are conducted in this Protocol.

Simulations are often purpose made to have one central focus and to disregard other aspects of the chemical system in order to maximize computational efficiency. What a researcher chooses to include or remove from a computational model affects the accuracy of the model, and thus, they must seek a balance between computational efficiency and accuracy. In simulations governed by classical mechanics employing force fields, electronic and continuum behaviors are often simplified or ignored.[10,11,13,14] For example, one group of phenomena that is generally excluded is the energetics and transport of electrons. Thus, formation and breakage of bonds, energetic positions of the molecular orbitals, and electronic currents are typically not explicitly considered.[15−19] Proxies for such phenomena are often used instead if the phenomena are known to have an effect on the motion of the nuclei (e.g., polarizable and reactive models).[20,21] However, these proxies are limited to obtaining the correct atomic configuration, rather than modeling electron behavior (e.g., determination of the band gap of a material). On the opposite extreme, classical atomistic simulations generally lack the necessary length and time scales to directly predict macroscale behavior, such as deformation of a solid sample of macroscopic dimensions. Such behavior is typically the realm of continuum approaches. Therefore, we see that the dominant interactions (i.e., bond

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634

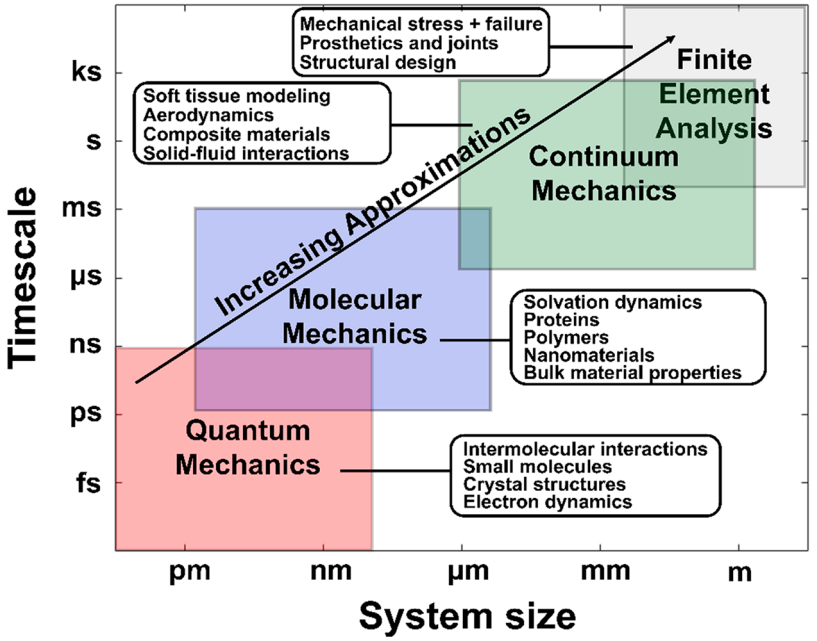


Figure 2. Summary of four common classes of simulations—quantum mechanics, molecular mechanics, continuum mechanics, and finite element analysis—and their typical size and time resolutions. As the time scale and system size increases, a greater number of assumptions and approximations must be made in order to ensure that a simulation runs efficiently. Several examples of common-use cases are given for each class of simulations.

scission, macroscale stretching, and stress concentration) that govern a model in part determine the time and length scales it can handle accurately. The inverse is true as well; a researcher must select a physical model appropriate for the information they want to extract from the simulation.

At the heart of classical atomistic simulations is the description of how the particles interact (i.e., the "force field"). The force field describes interparticle relationships in a system and is traditionally defined using analytic functions (e.g., quadratic terms to describe bond stretching). Since the electronic degrees of freedom are ignored, simulations employing force fields are relatively efficient. Nowadays, at various supercomputing facilities, simulations of millions of atoms, sometimes evolved over hundreds of nanoseconds in the case of MD, are routinely performed. The parameters that define the force fields (e.g., bond lengths and associated vibrational frequencies) are obtained in a variety of ways, including from accurate electronic structure and quantum mechanical calculations of model systems. In these cases, the resulting force field can be quite general and predictive. As an alternative, quantum mechanical calculations can be used to directly describe atomic interactions, in so-called ab initio simulation.[22,23] The drawback of such approaches is that the length and time scales that are practically accessible are significantly reduced to thousands of atoms over hundreds of picoseconds of dynamics, respectively. The choice of force fields or quantum mechanics depends on the property being calculated, the size of the system being considered, and the time scale of the process being modeled. This Protocol primarily focuses on MC or MD simulations that require use of force fields, although we will touch upon ab initio simulations where appropriate.

The key characteristic of classical force field-based simulations is that all interactions between components of the system are described using classical, analytical equations. As noted previously, this approach stands in contrast to simulations which explicitly model subatomic components like electrons (e.g., density functional theory, Hartree−Fock) or resolve higher length and/or time scales without explicitly modeling atoms (e.g., finite element analysis, finite-difference time domain). However, force field simulations also differ in terms of atomic resolution. For example, hydrogen atoms are sometimes not explicitly considered but rather grouped with the atom to which they are bonded. Going one step further, multiple similar atoms or functional groups of atoms can be grouped into a "superatom" to simplify the simulation and reduce the computational power required to perform it. This type of simulation, in general, is described by the term "coarse grained." On the other end of the spectrum, force field simulations can be used to approximate the behavior of subatomic particles in polarizable force fields. Here, there are various approaches for modeling the subatomic degrees of freedom (e.g., Drude oscillators, point dipoles, fluctuating charges). Yet, all examples described fall under the same class of simulations ("molecular mechanics"), and thus, the general procedures described herein apply to any of these types of simulations as well.

An experimentalist might be familiar with a wide range of characterization techniques that can be conducted with one instrument common to their field of study. Although the equipment used remains the same, each of these characterization techniques could give distinctive data useful for understanding their system at hand. Simulations are similar in this regard. Although the instrument used for all simulations is a computer, many different classes of simulations exist, each of which focuses on systems of different sizes with different time scales of interest. Likewise, as alluded to above, each class of simulations may contain several types of simulations that are specialized for specific systems. We summarize four common classes of simulations (quantum mechanics, molecular mechanics, continuum mechanics, and finite element analysis) below along with a general range of their resolutions and an

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634

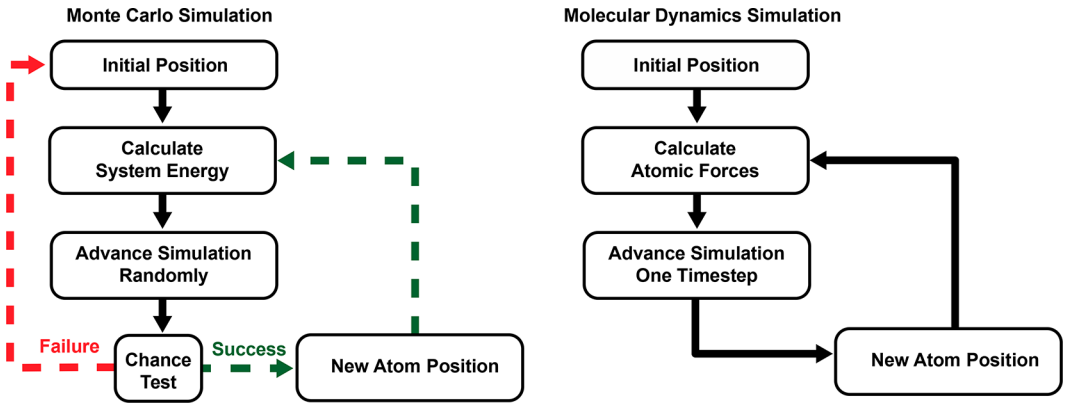


Figure 3. Overview of Monte Carlo (MC) and molecular dynamics (MD) simulations. Both systems progress from an initial, user-defined arrangement of atoms. In MC simulations, the energy of the system is calculated, while in MD simulations, the net force on each atom is calculated. MC simulations are then advanced randomly, given a probabilistic "chance" test. Dotted lines represent possible outcomes: system could fail the test and return to its initial position or succeed and advance to a new atomic configuration. In comparison, MD simulations are advanced deterministically using the force on the atoms to determine where they will end up a very short interval in the future. In both cases, the processes are then repeated (i.e., a new energy of the system is then calculated in a Monte Carlo simulation, while new forces are calculated in a molecular dynamics simulation) until enough of the system is sampled that the thermodynamics can be computed.

incomplete list of examples of systems that would typically fall under their purviews (Figure 2). Not all classes of simulations are hammers, and not all systems of interest are nails; no one class of simulations is "one-size-fits-all", and thus, a researcher must judiciously select a simulation appropriate for their system under study. In general, an increase in the time scale or size of the system of interest requires a greater number of assumptions and approximations. These assumptions are made in order to complete a simulation in a reasonable amount of time as well as balance the trade-off between accuracy of the simulation and efficiency (e.g., time, cost). For example, quantum mechanical simulations make several fundamental assumptions regarding electronic behavior (e.g., objects are restricted to discrete values). In molecular mechanics, simulations generally assume no quantum effects with at least partial separation of atoms. In continuum mechanics, objects within systems are generally assumed to be homogeneous and isotropic (as well as typically under steady-state conditions). A system modeled using finite element analysis (FEA) might assume idealized materials (e.g., no defects) and loading conditions.

Force field-based simulations can give insight into how the atomic configuration of a material changes naturally over time or in response to some external stimulus. A detailed picture of material properties comes at a price in that it is typically computationally demanding. As a necessity, modern-day simulations exploit the computing power of local clusters or, more commonly, supercomputing facilities. A supercomputer uses up to tens of thousands of processors simultaneously and facilitates communication between processors, providing the infrastructure to model complex material systems faster than could be modeled using local hardware (e.g., commercial laptops and desktops).[24] Yet, even on a supercomputer, using conventional MD to evolve the system forward in time, a single nanosecond simulating a reasonably sized system—approximately 100 000 atoms—could take several hours for an unoptimized system. We do note, however, that optimizing the configuration of such a simulation could reduce this estimate by several orders of magnitude.[25−27] As these time scales are substantial even on specialized hardware, we will focus on describing simulations on a supercomputer. However, the general process this Protocol describes, at least for software operation, would be similar for any computational architecture (i.e., both local and cloud resources).

Another feature of force field-based simulations is its relative accuracy, particularly when detailed information about the structure or thermodynamics of a system is desired. However, it is not without significant limitations. An experimental researcher must maintain reasonable expectations for what information can be garnered and how complex the simulations can be made. Among the most common uses for force field-based simulations are those elucidating biological or biochemical systems for research in chemistry or biology (e.g., drug development). For example, for proteins, a force field-based simulation can help determine the shape of the active site, how the active site changes in conformation upon the binding of a ligand (whether to the site itself or an allosteric binding site), or even the overall shape and positioning of the active site. In organic systems, these simulations may give insight into the structure of a polymer chain in solution and determine useful properties, such as its radius of gyration or end-to-end distance. In inorganic systems, these simulations can elucidate the crystallographic structure of materials as well as thermodynamic observables (e.g., heat of formation).

In general, the greater the complexity of a system and the more computational power is used, the more accurate a simulation will be. There are a number of reasons why the complexity and computational cost of a simulation using force fields might increase. For example, simulation times increase when considering processes that (1) evolve over a long period of time, (2) require simulation of large cell sizes (e.g., a larger molecule, cell, or polymer chain), (3) undergo chemical changes, or (4) involve various nonequilibrium processes. The crystallization of many materials can take place over minutes or hours, far outside the accessible time scale of equilibrium atomistic simulation. Another example might be the interaction of multiple large protein complexes. Though critical dynamics might be captured on the hundreds of nanoseconds time scale (at the upper range of practicality), the sheer size of these systems makes direct simulation on experimentally relevant time scales difficult. Likewise, while nonequilibrium processes

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634

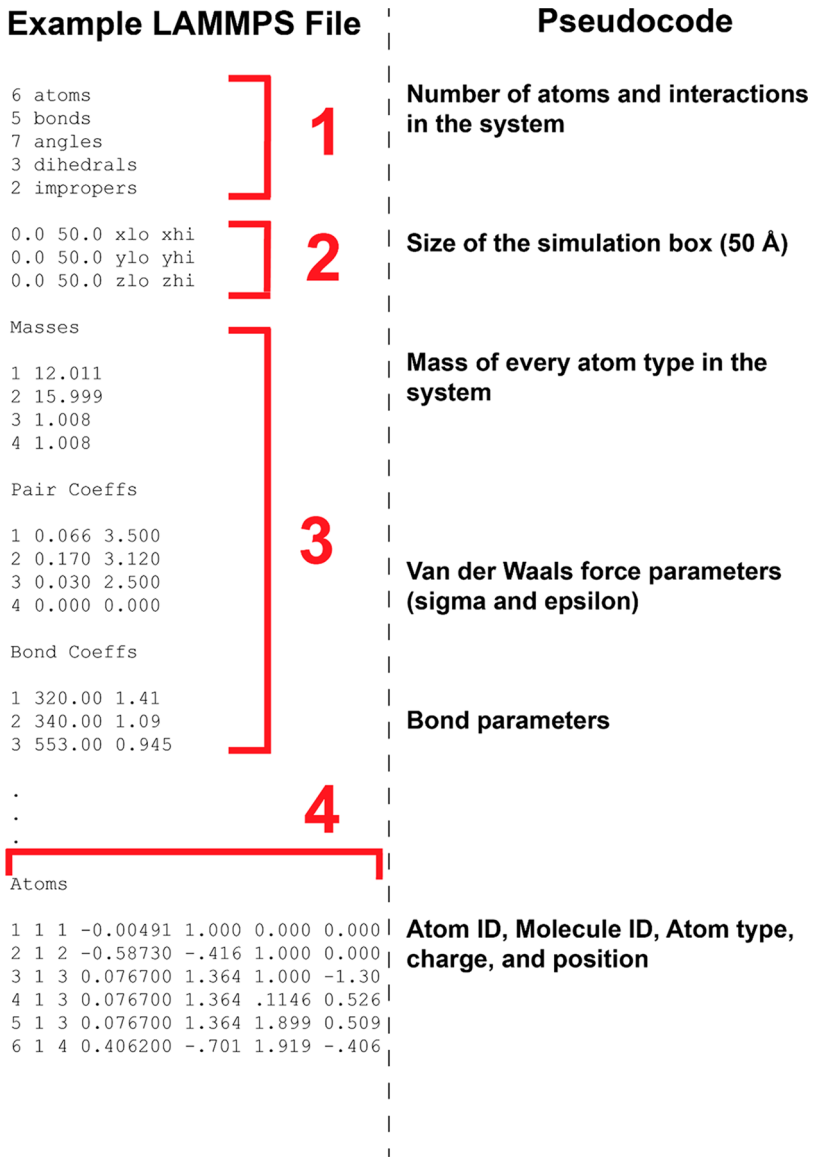


Figure 4. Example of a data file used by the LAMMPS MD simulations engine. This file describes a simple simulation of an isolated methanol molecule in a large simulation box. File is given alongside a pseudocode interpretation of each section of the file. In addition, numbers are assigned to each section of the file, and full text descriptions are given here. (1) Number of atoms and number of explicit interactions (e.g., bonds, angles, dihedrals) in the system. For a simple methanol system, there are 6 atoms, 5 bonds, and so on. (2) Size of the simulation box. This simulation of the methanol molecule will take place in a cubic box with lengths of 50 Å, containing only a single methanol molecule. (3) Descriptions of each atom type, bond type, etc., are given. Atom type 1 describes the carbon atom in methanol, atom type 2 describes the oxygen atom, and atom types 3 and 4 describe the hydrogen atoms bonded to the carbon and oxygen, respectively. "Pair Coeffs" section provides the sigma and epsilon parameters of Lennard–Jones interactions. "Bond Coeffs" provides the spring constant and equilibrium length (in Angstroms) of the bonds, and so on. (4) Specific information about each atom in the system is specified. Each atom is defined with a numerical ID, a molecule ID, an atom type (e.g., the carbon atom is type 1, the oxygen atom is type 2), and initial *xyz* coordinates. With this the initial state of the system is completely described.

can be observed through these simulations, a researcher must take care that they are simulating a sufficient amount of time to allow the simulation to adapt and change realistically. For example, a lack of sufficient time for a simulation of a solid polymeric sample under strain may not allow for the atoms to naturally contort and relax. Due to the system not reaching its equilibrium morphology, artifacts may dominate the system (e.g., nonphysical bond stretching). Additional complexity can be added when (1) describing chemical reactions (e.g., the chemistry of bond-breaking and -forming events) or (2) applying more complex interaction potentials (e.g., reactive force fields or ab initio molecular dynamics approaches). Therefore, the general complexity of evaluating the forces or energy of many systems of interest usually necessitates downscaling of simulation time and size for practicality.

Methods of Interest. The computational schemes we focus on are molecular dynamics (MD) and Monte Carlo (MC) methods (Figure 3).[10,11] These two approaches are similar in many ways. In both cases, the interactions between atoms define the energies of the system and the forces between the atoms. A total energy is the assigned to the system, or a net force is calculated for each atom.

Monte Carlo simulations focus entirely on the energy of the system. Here, the initial configuration is randomly changed (either displacing an atom, rotating a molecule, or inserting or deleting a molecule), and the energy of the new configuration

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634

is calculated. The simulation then has a decision to make: accept this random change in configuration or go back to the original configuration. The decision occurs probabilistically, with the likelihood of keeping the new configuration increasing as the energy difference with the previous configuration decreases.[28] In this way, the system will preferentially—but not always—explore low-energy conformations. Simulations using MC methods thus mostly survey low-energy states while also traversing the energy barriers between low-energy, favorable conformations.

In contrast, a molecular dynamics simulation evolves a system forward in time based on the forces on the atoms. While MC samples configurations in a probabilistic fashion, MD simulations—at least in their purest form—move deterministically.[29] On the basis of again an initial conformation of atoms, a net force is calculated on each atom. The system is then advanced in time by numerical integration: the forces and masses of each atom are used to calculate the acceleration of the atoms according to Newton's second law.[30,31] The accelerations are then integrated over a small increment of time (called a "time step") to generate a new position of the atoms. For numerical stability, the time step is typically related to the fastest motion in the system. For example, in organic materials, this is often a C−H bond vibration due to their very short time scale (on the order of femtoseconds).[32,33] The process is then iterated, and the resulting thermodynamic properties are obtained by statistical averaging.

MC and MD simulations both determine the energetic minima and distribution of energy states of a system at equilibrium. However, they have key differences in how they proceed. MD simulations sample configurations that are close in energy to the initial configuration (relative to the temperature), which means that the chances of capturing very high energy states or overcoming large energy barriers is small. MC simulations explicitly encompass the entire energy landscape, but the time evolution of the system is lost. Due to the reliance of MC on random moves, it can more easily overcome energy barriers in a free energy surface than MD. However, this advantage comes at the cost of mechanistic information, that is, how the system might actually traverse the energy landscape. Which approach is most useful for a particular system depends on the complexity of the system, computational demands, and thermodynamic information that is desired.[34] Notably, the ability to access temporal information about a system is an advantage of MD simulations and facilitates the calculation of several useful properties (e.g., (self) diffusivity). As a result, MD simulations are usually preferred in materials science. However, if the equilibrium morphology of a material is the primary interest, MC techniques are attractive, as they are typically more efficient at utilizing computing resources. This computational efficiency is especially clear in systems defined by rugged or complicated potential energy landscapes. In addition, MC is often superior at evaluating systems in different phases, such as liquid−vapor equilibria and adsorption.[35,36]

PRACTICAL STEPS IN FORCE FIELD-BASED ATOMISTIC SIMULATIONS

1. Step I: Input Information. Three types of information are typically required to initialize a simulation: a description of the initial system, instructions for the software on the simulation tasks, and instructions for the hardware to run the simulation. This information is generally inputted in the form of submission files (e.g., files that are uploaded or read). For simplicity, we refer to these inputs, respectively, as data file, software file, and hardware file.

1.1.1. Creating the Data File and Specifying the Initial System. The data file describes the initial position of all of the atoms and defines how the atoms will interact. We note that the initial state of the system could instead be directly embedded within the program input (e.g., script) rather than being a data file that is uploaded.

A crucial aspect of initializing the system is the initial placement of the atoms. The initial position of the system is specified in the data file. In both MC and MD, the simulation is likely to initially move toward an energetic minimum nearby this initial configuration. If the initial configuration is near an unphysical (i.e., unrealistic) energetic minimum, e.g., extremely high in energy but with similar conformations that are even higher in energy, the simulation will progress within the unphysical minima, and unphysical results will be obtained.[37−40] The initialization of the simulation is also of great importance to the workflow chosen. There are many programs which will simply run a MD or MC simulation after it is initialized.[41−49] Therefore, this initialization is typically the only time the user has direct control over the simulation. Thus, much like an experimentalist must optimize measurement parameters to minimize artifacts due to the measurement itself, judicious selection of the initial conditions will facilitate efficient and meaningful simulations (Figure 4).

A data file may be created simply by "brute force", whereby the coordinates of the atoms are copied from a molecular rendering software (e.g., Avogadro).[50] Such an approach works well if the system is relatively small. However, for systems comprising thousands of atoms, this method is impractical. An additional complication is the need for randomness in the initial position. For example, consider a single polymer chain initially stretched out. The all-trans conformation, in which all dihedral potential energies are minimized, is likely an enthalpic metastable minimum. Thus, this extended conformation is easily built with common molecule drawing software packages. However, an extended polymer chain conformation is exceedingly unlikely to occur in reality, as such a state is highly unfavorable entropically. Therefore, this conformation may represent a poor choice of the initial position, particularly if rotation of the monomers to a more randomized configuration is a slow process that would take a long time to achieve (i.e., conjugated polymers). Instead, it is often optimal to create random configurations of the atoms using Boltzmann weighting. This randomness is a base requirement in many systems of interest. For example, in a liquid, it would be essentially impossible for molecules to be oriented in the same direction. Therefore, the molecules should be rotated to introduce randomness while maintaining a Boltzmann distribution. Similarly, when simulating a glass, not every atom is evenly spaced (as in an ideal crystalline lattice), and so some degree of randomness should be built into the initial state. To generate more realistic initial positions of atoms for more complicated systems, it is often necessary to use a program to impose randomness in the system. The many different ways in which randomness can be introduced and the advantages and disadvantages of each method have been discussed by the thorough work of others.[51,52]

In MD simulations, another important consideration is the accounting of steric repulsive forces at short distances, that is,

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634

for many types of pairwise interactions, the energy of repulsion between atoms increases dramatically when atoms get too close. This "Pauli repulsion force" is frequently modeled in a force field by an exponential function. The exponential nature of the repulsion means that two molecules initialized too close together will move apart very rapidly at the next time step. In situations where the two molecules move farther apart than can be numerically accounted for, the simulation becomes unstable (and can fail or crash). Therefore, when multiple molecules are placed in random positions, molecules should be initialized with adequate initial spacing. Several software packages exist to ensure a reasonable initial configuration.[53,54] Alternatively, in particularly difficult cases, a different style of interaction which does not involve exponential repulsive forces (e.g., soft potentials) may be used at the beginning of a simulation.

Finally, depending on the diversity of starting configurations, it may be useful to generate multiple different starting data files and to simulate each system simultaneously and independently.[55,56] For example, simulations of polymer films below their glass transition will remain relatively frozen in their initial position. Therefore, multiple simulations, averaged together, are needed to obtain accurate results about the overall film morphology and energetics.

1.1.2. Choice of Force Fields. Another critical component of a simulation is the description of the interactions between the components (e.g., atoms, molecules). As noted previously, when describing these interactions, we restrict ourselves to classical MC or MD simulations employing force fields. This force field analytically defines all interactions deemed important to the outcome of the simulation such as the flexibility of a bond or long-range attractivity of an atom. Notably, the force field is an all-inclusive entity: a single force field defines how any two arbitrary atoms several Angstroms apart will be treated as well as how two bonded atoms will behave.

The choice of force field is thus crucial. A researcher can either select an existing force field or attempt to develop their own. The latter is a notoriously difficult proposition for even the most seasoned researcher, and the use of widely available force fields for a variety of systems is highly recommended (Table 1).[44,57−75] Even if adjustments to the force field are necessary, it is typically beneficial to use a preexisting force field as a starting point. These force fields exist for a variety of applications.[42,43,74,76,77] For example, several different force fields exist to model proteins, such as AMBER and CHARMM.[76,78] Whether the force field is bespoke or generic, the force field must be appropriate for the system under consideration. If a key interaction is not contained in an existing force field, a researcher must either add it to the force field manually or generate a custom force field.

Table 1. List of Commonly Used Force Fields

force field	primary use	ref
AMBER	biomolecules, proteins	79−81
CHARMM	biomolecules, proteins	68,76,82,83
OPLS	organic liquids, general organic molecules	84−86
REAX-FF	reactive force field for bond formation and breaking	59,87,88
Drude	polarizable force field	89−91
GROMOS	hydrocarbons, biomolecules	70,92−95
AMOEBA	polarizable force field developed for biomolecules	96−99
MEAM	embedded atom force field for inorganics	100
UFF	general force field to apply to any arbitrary system	58
DREIDING	general force field to apply to any arbitrary system	60,101,102
MARTINI	coarse-grained force field for large simulations to lump together like atoms	103

For modeling organic systems, certain choices for the style of interaction used in the force field are standard (Figure 5). While the available types of atoms (usually differentiated by hybridization, e.g., sp^2, sp^3) and the associated parameters differ, the same general structure is used by common force fields (e.g., CHARMM, OPLS, AMBER).[69,76,78] For example, van der Waals interactions between nonbonded atoms are usually treated as interacting using Lennard−Jones 12−6 potentials. On the other hand, covalent bonds are defined between pairs of atoms in relatively close proximity, where the quantum mechanical electron-pairing force is described using a harmonic spring-like interaction. Similarly, for three atoms bonded in a linear fashion, angles are also defined to enforce a specific local geometry. Another common type of interaction in simulations of organic molecules occurs between four atoms. These interactions are generally called torsions. Here, we differentiate two main types: conventional dihedral and improper dihedral angles. Conventional dihedral angles are simple torsional angles between four atoms connected in a linear fashion (as defined in a basic organic chemistry course). Improper angles occur when three atoms are connected to another central atom and are often used to enforce planarity (e.g., in ammonia).

In contrast to organic materials, the behavior of inorganic materials is described by force fields of fundamentally different forms.[59,61−67] The behavior of inorganic materials is highly dependent on the presence of metallic bonds (and bonds to metal atoms) as opposed to covalent bonds in organic systems. To more accurately describe such systems, these force fields often employ generator functions that encode—or "embed"—the quantum mechanical electron density. Covalent-style interactions (i.e., bonds, angles) are often eschewed entirely in favor of these generators that depend only on distance.

While a force field is a convenient construct for the approximation of necessary intermolecular forces, which forces qualify as "important" in a system remain at the discretion of the researcher. For example, when describing weak van der Waals interactions, a researcher may consider London dispersion interactions between pairs of atoms. Alternatively, they might consider more complicated functions that incorporate contributions from many-body (e.g., three or four neighbor) interactions.[59,65−67] However, more complex force fields require more computational power, and thus, researchers must decide whether the additional accuracy is worth the additional computational cost. Alternatively, force fields may be eschewed entirely in favor of a quantum mechanical calculation.[22,23] Quantum mechanical calculations provide significantly greater accuracy when determining the energy (and thus forces) of a system. In addition, specific approximations for all relevant types of interactions (i.e., bonds, angles) within a system would no longer be needed. Rather, the relevant behaviors of the system will emerge naturally from the solution of the Hamiltonian. The trade-off is, once again, that these ab initio calculations are substantially more computationally intensive to perform. Thus, to counter this increased computational cost, the number of atoms in the

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634

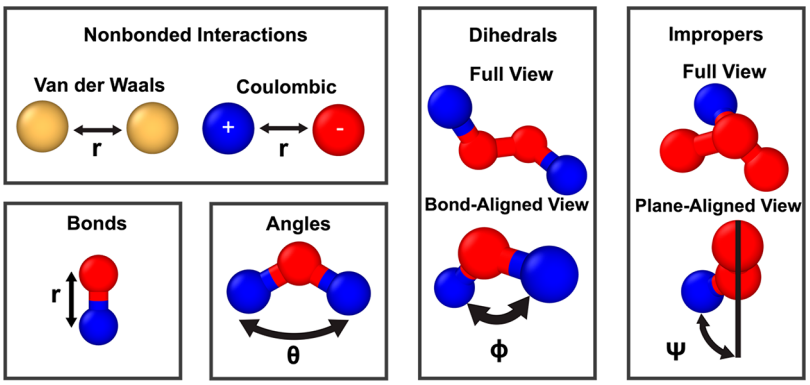


Figure 5. Types of interactions in a standard force field (e.g., force field for proteins or hydrocarbons). Generic forces between almost all atoms in the system, particularly van der Waals interactions, are shown in yellow. For other interactions, atoms undergoing forces are defined in blue, while other atoms in the system are shown in red. Nonbonded atoms experience electrostatic forces typically modeled using dispersion and Coulombic terms. Dispersion term typically accounts for the Pauli repulsion force at short distances and London dispersion forces at intermediate distances. These forces are usually described using a Lennard−Jones potential. For systems where atoms have partial atomic charges, the atoms experience Coulombic interactions. In addition, atoms that are chemically bonded to one another experience other forces and interact based on their distance apart (usually with a harmonic function). Three atoms that are bonded in a linear fashion interact with a potential based on the angle with respect to the central atom. Four atoms that are bonded in a linear fashion interact with a potential based on their orientation about the central bond, called a dihedral interaction. Improper angles are torsions created when one atom is bonded to three other atoms and a specific orientation (e.g., planarity) is required (full view). Potential is calculated based on the angle between one atom and the plane defined by the other three atoms (plane-aligned view).

system or length of time simulated generally must be decreased.

1.1.3. Creating the Software and Hardware Input Files. The next step is to run the simulation using specialized MC or MD software programs. For research-quality simulations, a researcher typically uses a supercomputer, a collection of high-quality hardware which runs a simulation with many computational resources (advanced processors and memory architectures). To maximize the efficient use of any such system, the software and hardware both require sets of instructions. We will discuss these input files together, though they take quite different forms. Generally speaking, the software input file is a set of instructions for actually running the simulations. It defines the thermodynamic state functions of interest (i.e., the temperature and pressure of interest in the model). The hardware input is a set of instructions that dictates what physical resources (e.g., processors and memory) are used to run the simulation (Figure 5).

A wealth of crucial information is contained in a software input file: what force field to use, how much time to simulate, which conditions to implement (e.g., boundary conditions, constraints such as temperature or pressure), which changes affect the simulation, and what is outputted from the program (Figure 6a). The parametrization of a force field can range in difficulty depending on the complexity of the system and the availability of parametrized potentials of interest. Generally speaking, the parameters of interactions are difficult to generate, so it is recommended to use an existing force field if possible. In the simplest case, the required functional forms are already supported in the MC or MD software, so using an existing force field can be relatively straightforward. For example, if the pairwise van der Waals interactions of the force field are in a Lennard−Jones style, they can be implemented using specific keywords in the software input file of most simulation packages (Table 2).

The next decision a researcher must make regards what external conditions to simulate. A natural choice is to attempt to mimic the relevant experimental conditions. Simulations can be run with constant pressure (P) or volume (V), constant temperature (T) or energy (E), and a constant number of particles (n) or chemical potential (μ). These choices define the statistical mechanical ensemble of the simulation. A summary of common ensembles is shown in Table 3.

The ensemble corresponds to specific constraints imposed on the system, and the relevant thermodynamic properties of the system (e.g., pressure, temperature, heat capacity) all have unique definitions in each ensemble. An ensemble is judiciously chosen such that it is relevant to the true experimental system being considered. For example, to understand the physical properties of a gas in a closed container, a researcher might run a simulation at constant temperature, T, volume, V, and number of particles, N (i.e., the canonical ensemble). The choice of the ensemble is specified in the software input file.

The final consideration is how the system may be modified during its simulation. A wide variety of changes may happen to the system, e.g., it could be heated, cooled, stretched, or compressed, and the simulation should reflect these changes by including appropriate keywords in the software input file. However, it is crucial that these modifications be made with the time scale of simulation in mind. Evolution in structure caused by, for example, heating or stretching actually takes place over the course of milliseconds (at least). Except in very special cases, such time scales are inaccessible to atomistic simulations employing common force fields. To compensate, a researcher might be inclined to heat or cool the system very rapidly. However, this invariably leads to unwanted physics, e.g., superheating or supercooling, that does not realistically depict the processes involved (e.g., bond rotation to accommodate strain instead of bond stretching, diffusion of materials under heat). Therefore, obtaining values from any nonequilibrium process requires judicious care.

The hardware input file instructs the supercomputer which tasks to perform and how to perform them (i.e., how many hardware CPUs or GPUs to use, Figure 6b). Simulations will generally run faster if more supercomputing resources are used,

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634

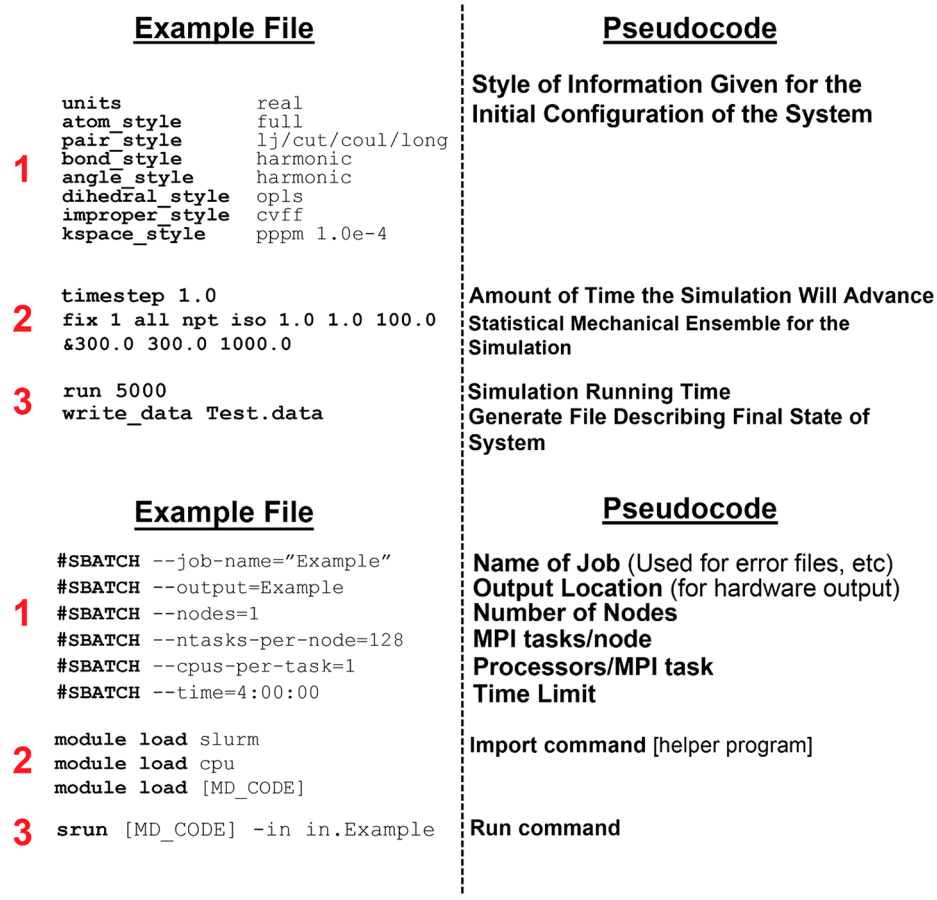


Figure 6. LAMMPS input to the simulation (a) software and (b) hardware input files. (a) Software input file is split up into three parts: (1) description of the system, (2) description of how to propagate the system, and (3) description of the stopping condition. Part 1 lists the relevant units used to represent the structures as well as all types relevant interactions (Lennard–Jones and Coulombic interactions for the above example). Part 2 describes the time step used for the calculation and the thermodynamic ensemble used for the calculation (defined in terms of constraints). Finally, part 3 specifies when to stop the calculation and what internal postprocessing should be done. (b) Hardware input file with information separated into three parts. Various supercomputing centers use different schedulers. Here, we provide example input using the SLURM scheduler.[104] In part 1, we define the basic parameters of the job, such as the name of the job, the place to put hardware output information, and how long to run the job for. In part 2, the hardware is instructed to update environment variables in order to access ("load") the necessary software for the simulation. In part 3, the supercomputer is instructed to run the simulation.

Table 2. List of Commonly Used MD and MC Codes

software	primary use	ref
LAMMPS	general MD, highly modifiable	105
NAMD	general MD, highly optimized	106
GROMACS	MD designed for biomolecules	107
CHARMM	MD designed for biomolecules	108
AMBER	MD designed for biomolecules	109,110
TINKER	general MD and MC	111
ABALONE	general MD and MC, primarily for biomolecules	112,113
BOSS	general MC	114
CP2K	ab initio MD	26
Q	MD implementing quantum behavior	115
DESMOND	software for MD	116
Materials Studio	software for MD and MC	117
Hoomd Blue	software for MD and MC	118,119
MCMD	software for MD and MC	120

Table 3. Summary of Common Statistical Mechanical Ensembles for Computational Simulations

ensemble name	variables (const.)	example
canonical	*NVT*	gas molecules in a closed container
microcanonical	*NVE*	gas molecules in a closed and insulated container
grand canonical	μVT	gas molecules in an open container
isobaric–isothermal (Gibbs)	*NPT*	gas molecules in a closed container sealed by a moving piston
isoenthalpic–isobaric (enthalpy)	*NPH*	gas molecules in a closed and insulated container sealed by a moving piston

but the relationship between the number of supercomputing resources used and the required simulation time is typically sublinear.[121,122] In other words, if twice the amount of the supercomputer resources are used, the simulation will run faster but not quite twice as fast. This sublinear relationship occurs because simulations running on multiple processors typically break the tasks up into parallel duties. This parallelization requires that the processors communicate with one another, thus increasing the workload. As more processors are added, the speedup often becomes increasingly sublinear as the amount of computational time dedicated to communication between processors increases. Another factor is that some processes cannot be parallelized. Because supercomputer facilities are shared resources, a scheduler (e.g., the program

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620–7634

which prioritizes different users of a supercomputing center) is typically used to allocate supercomputer resources. Much like a chemist might wait for the completion of a reaction or a materials scientist might wait for communal characterization tools to become available, simulation requests are generally queued until the requested resources (e.g., processors, nodes) become available. For this reason, requesting more resources on a supercomputer often means a longer queue time. Therefore, the "real" time required for a simulation might even increase if too many resources are requested, despite the speedup of the simulation itself.

As a final note on both software and hardware, the MC or MD software must be compiled in a specific way to optimize performance on the supercomputer. Sometimes, a supercomputer will have MC or MD software already available and optimized. However, if this is not the case or if a researcher wishes to use specialized packages or functionalities in the software, they must compile the software themselves. In this case, we recommend simply reaching out to the technical support of the supercomputer for advice on how to compile the software. If specialized compilation is required or resources are not available to assist with compilation, any documentation for the package should be sought out and followed. If no such documentation is easily available, reaching out to the creators of the package is typically a fruitful endeavor. Finally, a variety of packages exist for accelerating the performance of the software on the supercomputer. Each package typically has information about how they speed up different types of simulations in the user manual, and this should be consulted for best results.

1.2. Step 2: Running the Simulation. Once the simulation begins, little input is typically required of the researcher. As such, the choice of which software to use is most important. Researchers must balance competing interests of time and customizability. Scientists often look at novel systems that are not yet well understood, and these novel systems may require modification of existing software to handle interactions of interest. Some systems will be better suited to handle modifications, although at times the modifications will come at the expense of computational efficiency. However, this is general advice, and every situation is different. For a software package to provide the desired performance, a researcher must understand the particular needs and goals of their simulation.

Until now we have focused on simple molecular dynamics or Monte Carlo simulations, which are somewhat limited in their usefulness. Conventional MD simulations are often run on the order of nanoseconds. Monte Carlo simulations are similarly run for enough time steps to explore local minima but not to explore all accessible energy states. A variety of newer methods have been developed for modified versions of MC or MD simulations to explore time and space more rapidly, allowing the simulation to garner more data without a significant increase in computational time.[37,123−131] These methods—termed "advanced sampling" methods—are useful for collecting more complete sets of data. For example, techniques such as metadynamics and parallel tempering can help explore energy surfaces with rugged free energy surfaces.[132−135] These techniques can increase the speed for simulations to randomly sample a rare event or transition, which would ordinarily require a long simulation time. That said, the researcher must ensure that an advanced sampling method of interest is compatible with or easily incorporated into their simulation program or else undertake the (sometimes laborious) process of converting to another code.

1.3. Step 3: Analyzing the Simulation Data. Having successfully run the simulation, all that remains is to collect and organize the data into an understandable format. The raw output of an MC simulation that is returned to the user is the position of every atom in every conformation. For an MD simulation, the equivalent output is the position of every atom at every time step. The resulting data often requires postprocessing. For example, the physical properties of a system are often derived from distributions or averages over the trajectories of the atoms during the course of a simulation. Likewise, if dynamical information (e.g., characterizing transition pathways between conformations of a molecule) is needed, data visualization methods can be used to map trajectories of a component (e.g., atom, molecule) within the system. Some simulation codes have built-in analysis capabilities, allowing the user insight into the system as the simulation proceeds. Alternatively, data analysis must be performed afterward with commercial, open source, or custom-made software. The code used for analysis can range greatly in complexity. For example, a fairly straightforward script can be used to calculate the average morphology of a polymer by means of the radius of gyration, while more complicated calculations can be done to determine far more complex material properties. For example, if the system in question is being stretched, a program can be written to calculate the stress tensor on the system as a whole or atom by atom. For a polymer system, the crystallinity can be derived from knowledge of every atomic position. In a protein, changes to the shape of the active site or measurement of ligand binding affinity can be calculated. For inorganic systems, thermodynamic quantities or information about the crystal structure can be extracted from a simulation.

1.4. Example: Small-Molecule Drug Discovery. To place concepts described in this Protocol into context for experimentalists, we provide an example describing how these processes translate to real-world applications. We focus on a use case that has yielded experimental, tangible results: the use of computation simulations for guiding the design of small molecules for drug discovery. To contextualize these concepts, we focus on how such simulations are designed and performed by computational scientists.

Nearly every major pharmaceutical company now incorporates computational tools into their drug discovery workflow or pipeline. Atomistic simulations, and thus models, are routinely used in this drug discovery process. One particular computational method is of particular ubiquity: free energy perturbation (FEP) simulations, which can be used (among many applications) to predict how strongly a ligand binds to the active site of a protein. While this Protocol describes the FEP process at a high level, a vast library of studies provide significantly greater detail regarding procedures and applications.[136−140]

When orchestrating a FEP simulation, the mechanics of the simulation must be first considered. In this example, a FEP simulation is used to describe the binding affinity of a small molecule to a protein active site. Thus, a force field that accurately describes the interactions typical to the binding site and small molecule is necessary. For example, a force field that does not accurately handle hydrogen bonding will render a simulation unhelpful when modeling highly hydrophilic ligands or binding sites. Alternatively, a binding site that coordinates to

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634

a metal will likely require use of a special force field in order to accurately describe the interactions of metals and their d orbitals.

To obtain accurate information about the system of interest, a data file that considers all relevant components must be constructed, the two most important components of which are the small molecule and the protein binding site of interest. However, additional chemical moieties are also likely required to be included in the system. For example, the full protein might be included as well if other regions of the protein will likely affect how the shape of the binding site changes and adapts when bound by a small molecule. In addition, because the environment will affect interactions between the protein and the small molecule, water must be included as well. Thus, a computational scientist might choose to construct a simulation in which the system is defined as the protein and the small molecule surrounded by water molecules (at a specific density). In this data file, interaction constants (e.g., sigma and epsilon for Lennard−Jones interactions, spring constants for bonded interactions) will also be assigned based on the force field selected. If an automated (e.g., in-house) method for assigning force field parameters to a biochemical simulation exists, that will be implemented at this stage. If no such automation exists, the process of assigning force field parameters becomes far more difficult. The computational scientist will either need to manually assign the parameters or (much more likely) develop their own process for assigning the parameters of the chosen force field.

Next, the parameters of the simulation must be defined in the software input file. In this FEP simulation example, two key parameters are of significant interest: (1) the amount of simulation time (i.e., time scale) the simulation is performed over and (2) any restraints that allow for a more rapid binding of the small molecule to the active site. These two parameters must be judiciously chosen based on the specifics of the system; here, it is sufficient to know that this information is determined by descriptors in the software input.

Finally, the hardware input must be specified, and several considerations must be made regarding how the simulation will be run. For example, the availability of hardware is a key constraint when considering how to run the simulation. Access to a local cluster, supercomputer, or similar cloud computing resources typically allows the researcher to default to that hardware of choice. In addition, access to specialized hardware resources (e.g., GPUs) must be taken into account at this stage. Likewise, the software being utilized should be capable of taking advantage of the hardware resources available in order to maximize efficiency. With these three inputs completed, an FEP simulation is ready to be run.

CONCLUSIONS

We have now walked through the entire process of performing a simulation for material systems, from the types of information required as inputs, to running the simulation, to the analysis of the output. This Protocol has focused on force field-based, classical atomistic simulations, but the workflow presented is similar for many other types of simulations. As advancements in computing technology have greatly improved the accessibility of computational simulations for research purposes, the interwoven relationship between experimental and computational research has become increasingly synergistic. Computational scientists require the collaboration of experimentalists to validate and refine their models. Experimentalists require microscopic insights into material structure and morphology that can be accessed from such simulations, that is, in the last several decades, the boundaries between experimentalists and computational scientists have blurred, and it has become increasingly common for researchers to be familiar with both. This synergy is beneficial to modern data science approaches (e.g., machine learning) that are of interest to chemical, biological, and materials research. Yet, practical improvements in advanced statistical models require further advancements in both experimental characterization and computational models. In addition, the interdisciplinary nature of research has greatly expanded. It is now commonplace for one research group to have research interests spanning several fields of study and for individual researchers to be involved in multiple projects that differ greatly in focus. As stated, the general workflow described in this Protocol is widely applicable, and our overarching goal was to offer experimentalists useful foundational knowledge for developing highly translatable computational skills for any material system.

Should experimentalists desire to deepen their knowledge of the methods described in this Protocol or to begin incorporating these methods into their own work, we have included a list of further resources.[11,22,141−144] Much of our discussion in this Protocol has been focused on the decision-making process, largely centered around the trade-offs between simulation accuracy, simulation time, and supercomputer resources. In doing so, we hope to help experimentalists translate the intuition they already have to those familiar to computational studies, as it is not difficult to draw comparisons between the two. To give an example, if a researcher wished to control an experiment to the greatest level of accuracy possible, an experimentalist would degas every solvent, recrystallize every reagent, control the physical environment to immeasurable precision, and take measurements over the course of hours or days. Yet, this is typically not done because the trade-off between time spent and experimental outcome can be prohibitively unfavorable. In a similar vein, we introduced several general relationships between computational resources available and the simulated system of interest that affect the trade-offs we describe in this Protocol. In doing so, we hope to help experimentalists develop a high-level understanding of the advantages, disadvantages, capabilities, and constraints of computational work. Likewise, we hope that our explanation of the simulation process will be useful for experimentalists interested in incorporating computational data in their own work or for those whose work overlaps with that of computational scientists or, if this is not possible, to at least spare our computational colleagues from the question why is your simulation taking so long?

AUTHOR INFORMATION

Corresponding Authors

Tod A. Pascal − *Department of NanoEngineering, University of California, San Diego, La Jolla, California 92093-0448, United States;* orcid.org/0000-0003-2096-1143; Email: tpascal@eng.ucsd.edu

Darren J. Lipomi − *Department of NanoEngineering, University of California, San Diego, La Jolla, California 92093-0448, United States;* orcid.org/0000-0002-5808-7765; Email: dlipomi@eng.ucsd.edu

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634

Authors

Andrew T. Kleinschmidt − *Department of NanoEngineering, University of California, San Diego, La Jolla, California 92093-0448, United States;* orcid.org/0000-0002-6036-6878

Alexander X. Chen − *Department of NanoEngineering, University of California, San Diego, La Jolla, California 92093-0448, United States;* orcid.org/0000-0003-1919-6755

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.chemmater.2c00292

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported by the Air Force Office of Scientific Research (AFOSR) grant no. FA9550-19-1-0278 to D.J.L. This work was partially supported by the NSF through the University of California San Diego Materials Research Science and Engineering Center (MRSEC, grant DMR-2011924) funding to D.J.L. and T.P. A.K. acknowledges additional support from the Achievement Reward for College Scientists (ARCS) Foundation.

■ REFERENCES

(1) Smaller, Faster, Cheaper, Over: The Future of Computer Chips. *The New York Times*; https://www.nytimes.com/2015/09/27/technology/smaller-faster-cheaper-over-the-future-of-computer-chips.html (accessed Jun 3, 2022).

(2) de Pablo, J. J.; Jackson, N. E.; Webb, M. A.; Chen, L.-Q.; Moore, J. E.; Morgan, D.; Jacobs, R.; Pollock, T.; Schlom, D. G.; Toberer, E. S.; et al. New Frontiers for the Materials Genome Initiative. *npj Comput. Mater.* **2019**, *5* (1), 41.

(3) Ehrenberger, T.; Cantley, L.; Yaffe, M. Computational Prediction of Protein-Protein Interactions. *Methods Mol. Biol.* **2015**, *1278*, 57−75.

(4) Gebhardt, J.; Kiesel, M.; Riniker, S.; Hansen, N. Combining Molecular Dynamics and Machine Learning to Predict Self-Solvation Free Energies and Limiting Activity Coefficients. *J. Chem. Inf. Model.* **2020**, *60* (11), 5319−5330.

(5) Summers, A. Z.; Gilmer, J. B.; Iacovella, C. R.; Cummings, P. T.; McCabe, C. MoSDeF, a Python Framework Enabling Large-Scale Computational Screening of Soft Matter: Application to Chemistry-Property Relationships in Lubricating Monolayer Films. *J. Chem. Theory Comput.* **2020**, *16* (3), 1779−1793.

(6) Shih, C. Y.; Wu, C.; Shugaev, M. V.; Zhigilei, L. V. Atomistic Modeling of Nanoparticle Generation in Short Pulse Laser Ablation of Thin Metal Films in Water. *J. Colloid Interface Sci.* **2017**, *489*, 3−17.

(7) Braun, E.; Gilmer, J.; Mayes, H. B.; Mobley, D. L.; Monroe, J. I.; Prasad, S.; Zuckerman, D. M. Best Practices for Foundations in Molecular Simulations [Article v1.0]. *Living J. Comput. Mol. Sci.* **2019**, *1* (1), 5957−5957.

(8) The Sherrill Group: Notes; http://vergil.chemistry.gatech.edu/notes/index.html (accessed May 28, 2022).

(9) Video Lectures | Atomistic Computer Modeling of Materials (SMA 5107) | Materials Science and Engineering | MIT OpenCourseWare; https://ocw.mit.edu/courses/3-320-atomistic-computer-modeling-of-materials-sma-5107-spring-2005/video_galleries/video-lectures/ (accessed May 28, 2022).

(10) Hollingsworth, S. A.; Dror, R. O. Molecular Dynamics Simulation for All. *Neuron* **2018**, *99* (6), 1129−1143.

(11) Harrison, R. L. Introduction to Monte Carlo Simulation. *AIP Conf. Proc.* **2009**, *1204*, 17−21.

(12) Noé, F.; Tkatchenko, A.; Müller, K.-R.; Clementi, C. Machine Learning for Molecular Simulation. *Annu. Rev. Phys. Chem.* **2020**, *71*, 361−390.

(13) Argaman, N.; Makov, G. Density Functional Theory: An Introduction. *Am. J. Phys.* **2000**, *68* (1), 69−79.

(14) Wang, K. F.; Wang, B. L.; Kitamura, T. A Review on the Application of Modified Continuum Models in Modeling and Simulation of Nanostructures. *Acta Mech. Sin.* **2016**, *32* (1), 83−100.

(15) Ducharne, B.; Juuti, J.; Bai, Y. A Simulation Model for Narrow Band Gap Ferroelectric Materials. *Adv. Theory Simulations* **2020**, *3* (9), 2000052.

(16) Nasir, S. N. F. M.; Ullah, H.; Ebadi, M.; Tahir, A. A.; Sagu, J. S.; Mat Teridi, M. A. New Insights into Se/BiVO4 Heterostructure for Photoelectrochemical Water Splitting: A Combined Experimental and DFT Study. *J. Phys. Chem. C* **2017**, *121* (11), 6218−6228.

(17) Li, H.; Henkelman, G. Dehydrogenation Selectivity of Ethanol on Close-Packed Transition Metal Surfaces: A Computational Study of Monometallic, Pd/Au, and Rh/Au Catalysts. *J. Phys. Chem. C* **2017**, *121* (49), 27504−27510.

(18) Falek, W.; Benali-Cherif, R.; Golea, L.; Samai, S.; Benali-Cherif, N.; Bendeif, E. E.; Daoud, I. A Structural Comparative Study of Charge Transfer Compounds: Synthesis, Crystal Structure, IR, Raman-Spectroscopy, DFT Computation and Hirshfeld Surface Analysis. *J. Mol. Struct.* **2019**, *1192*, 132−144.

(19) Michaels, W.; Zhao, Y.; Qin, J. Atomistic Modeling of PEDOT:PSS Complexes I: DFT Benchmarking. *Macromolecules* **2021**, *54* (8), 3634−3646.

(20) Naserifar, S.; Brooks, D. J.; Goddard, W. A.; Cvicek, V. Polarizable Charge Equilibration Model for Predicting Accurate Electrostatic Interactions in Molecules and Solids. *J. Chem. Phys.* **2017**, *146* (12), 124117.

(21) Lamoureux, G.; Roux, B. Modeling Induced Polarization with Classical Drude Oscillators: Theory and Molecular Dynamics Simulation Algorithm. *J. Chem. Phys.* **2003**, *119* (6), 3025.

(22) Pastore, G.; Smargiassi, E.; Buda, F. Theory of *Ab Initio* Molecular-Dynamics Calculations. *Phys. Rev. A* **1991**, *44* (10), 6334.

(23) Iftimie, R.; Minary, P.; Tuckerman, M. E. Ab Initio Molecular Dynamics: Concepts, Recent Developments, and Future Trends. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102* (19), 6654−6659.

(24) Mann, A. Core Concept: Nascent Exascale Supercomputers Offer Promise, Present Challenges. *Proc. Natl. Acad. Sci. U. S. A.* **2020**, *117* (37), 22623−22625.

(25) Nguyen, T. D. GPU-Accelerated Tersoff Potentials for Massively Parallel Molecular Dynamics Simulations. *Comput. Phys. Commun.* **2017**, *212*, 113−122.

(26) Kühne, T. D.; Iannuzzi, M.; Del Ben, M.; Rybkin, V. V.; Seewald, P.; Stein, F.; Laino, T.; Khaliullin, R. Z.; Schütt, O.; Schiffmann, F.; et al. CP2K: An Electronic Structure and Molecular Dynamics Software Package - Quickstep: Efficient and Accurate Electronic Structure Calculations. *J. Chem. Phys.* **2020**, *152* (19), 194103.

(27) Smirnov, G. S.; Stegailov, V. V. Efficiency of Classical Molecular Dynamics Algorithms on Supercomputers. *Math. Model. Comput. Simulations 2016 86* **2016**, *8* (6), 734−743.

(28) Rosenthal, J. S. Parallel Computing and Monte Carlo Algorithms. *Far East J. Theor. Stat.* **2000**, *4*, 207−236.

(29) Pastor, R. W.; Brooks, B. R.; Szabo, A. An Analysis of the Accuracy of Langevin and Molecular Dynamics Algorithms. *Mol. Phys.* **1988**, *65* (6), 1409−1419.

(30) Rodger, P. M. On the Accuracy Of Some Common Molecular Dynamics Algorithms. *Mol. Simul.* **1989**, *3* (5−6), 263−269.

(31) Grubmüller, H.; Heller; Windemuth, A.; Schulten, K. Generalized Verlet Algorithm for Efficient Molecular Dynamics Simulations with Long-Range Interactions. *Mol. Simul.* **1991**, *6* (1-3), 121−142.

(32) Winger, M.; Trzesniak, D.; Baron, R.; van Gunsteren, W. F. On Using a Too Large Integration Time Step in Molecular Dynamics Simulations of Coarse-Grained Molecular Models. *Phys. Chem. Chem. Phys.* **2009**, *11* (12), 1934−1941.

(33) Fincham, D. Choice of Timestep in Molecular Dynamics Simulation. *Comput. Phys. Commun.* **1986**, *40* (2−3), 263−269.
(34) Hafner, J. Atomic-Scale Computational Materials Science. *Acta Mater.* **2000**, *48* (1), 71−92.
(35) Panagiotopoulos, A. Z. Direct Determination of Phase Coexistence Properties of Fluids by Monte Carlo Simulation in a New Ensemble. *Mol. Phys.* **1987**, *61* (4), 813−826.
(36) Panagiotopoulos, A. Z.; Quirke, N.; Stapleton, M.; Tildesley, D. J. Phase Equilibria by Simulation in the Gibbs Ensemble. *Mol. Phys.* **1988**, *63* (4), 527−545.
(37) Comer, J.; Phillips, J. C.; Schulten, K.; Chipot, C. Multiple-Replica Strategies for Free-Energy Calculations in NAMD: Multiple-Walker Adaptive Biasing Force and Walker Selection Rules. *J. Chem. Theory Comput.* **2014**, *10* (12), 5276−5285.
(38) Dehez, F.; Tarek, M.; Chipot, C. Energetics of Ion Transport in a Peptide Nanotube. *J. Phys. Chem. B* **2007**, *111* (36), 10633−10635.
(39) Romo, T. D.; Grossfield, A. Unknown Unknowns: The Challenge of Systematic and Statistical Error in Molecular Dynamics Simulations. *Biophys. J.* **2014**, *106* (8), 1553.
(40) Nijmeijer, M. J. P.; Bruin, C.; Bakker, A. F.; van Leeuwen, J. M. J. Wetting and Drying of an Inert Wall by a Fluid in a Molecular-Dynamics Simulation. *Phys. Rev. A* **1990**, *42* (10), 6052.
(41) Plimpton, S. Fast Parallel Algorithms for Short-Range Molecular Dynamics. *J. Comput. Phys.* **1995**, *117* (1), 1−19.
(42) Case, D.; Cheatham, T.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. The Amber Biomolecular Simulation Programs. *J. Comput. Chem.* **2005**, *26* (16), 1668−1688.
(43) Brooks, B.; Brooks, C.; Mackerell, A.; Nilsson, L.; Petrella, R.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; et al. CHARMM: The Biomolecular Simulation Program. *J. Comput. Chem.* **2009**, *30* (10), 1545−1614.
(44) Christen, M.; Hünenberger, P.; Bakowies, D.; Baron, R.; Bürgi, R.; Geerke, D.; Heinz, T.; Kastenholz, M.; Kräutler, V.; Oostenbrink, C.; et al. The GROMOS Software for Biomolecular Simulation: GROMOS05. *J. Comput. Chem.* **2005**, *26* (16), 1719−1751.
(45) Abraham, M. J.; Murtola, T.; Schulz, R.; Pall, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High Performance Molecular Simulations through Multi-Level Parallelism from Laptops to Supercomputers. *SoftwareX* **2015**, *1−2*, 19−25.
(46) Chow, E.; Rendleman, C. A.; Bowers, K. J.; Dror, R. O.; Hughes, D. H.; Gullingsrud, J.; Sacerdoti, F. D.; Shaw, D. E. *Desmond Performance on a Cluster of Multicore Processors Hardware and Operating Environment Benchmark Systems and Simulation Parameters*; 2008.
(47) Watts, K. S.; Dalal, P.; Tebben, A. J.; Cheney, D. L.; Shelley, J. C. Macrocycle Conformational Sampling with MacroModel. *J. Chem. Inf. Model.* **2014**, *54* (10), 2680−2696.
(48) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kalé, L.; Schulten, K. Scalable Molecular Dynamics with NAMD. *J. Comput. Chem.* **2005**, *26* (16), 1781−1802.
(49) Lagardere, L.; Jolly, L.-H.; Lipparini, F.; Aviat, F.; Stamm, B.; Jing, Z. F.; Harger, M.; Torabifard, H.; Cisneros, G. A.; Schnieders, M. J.; et al. Tinker-HP: A Massively Parallel Molecular Dynamics Package for Multiscale Simulations of Large Complex Systems with Advanced Point Dipole Polarizable Force Fields. *Chem. Sci.* **2018**, *9* (4), 956−972.
(50) Hanwell, M. D.; Curtis, D. E.; Lonie, D. C.; Vandermeersch, T.; Zurek, E.; Hutchison, G. R. Avogadro: An Advanced Semantic Chemical Editor, Visualization, and Analysis Platform. *J. Cheminform.* **2012**, *4* (8), 17.
(51) Alder, B. J.; Wainwright, T. E. Studies in Molecular Dynamics. I. General Method. *J. Chem. Phys.* **1959**, *31* (2), 459.
(52) Martinez, L.; Andrade, R.; Birgin, E. G.; Martínez, J. M. PACKMOL: A Package for Building Initial Configurations for Molecular Dynamics Simulations. *J. Comput. Chem.* **2009**, *30* (13), 2157−2164.
(53) Martínez, L.; Andrade, R.; Birgin, E. G.; Martínez, J. M. PACKMOL: A Package for Building Initial Configurations for Molecular Dynamics Simulations. *J. Comput. Chem.* **2009**, *30* (13), 2157−2164.
(54) Meunier, M.; Robertson, S. Materials Studio 20th Anniversary. *Mol. Simul.* **2021**, *47* (7), 537−539.
(55) Yang, S.; Banavali, N. K.; Roux, B. Mapping the Conformational Transition in Src Activation by Cumulating the Information from Multiple Molecular Dynamics Trajectories. *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106* (10), 3776−3781.
(56) Mirjalili, V.; Noyes, K.; Feig, M. Physics-Based Protein Structure Refinement through Multiple Molecular Dynamics Trajectories and Structure Averaging. *Proteins: Struct., Funct., Bioinf.* **2014**, *82*, 196−207.
(57) Cornell, W. D.; Cieplak, P.; Bayly, C. I.; Gould, I. R.; Merz, K. M.; Ferguson, D. M.; Spellmeyer, D. C.; Fox, T.; Caldwell, J. W.; Kollman, P. A. A Second Generation Force Field for the Simulation of Proteins, Nucleic Acids, and Organic Molecules. *J. Am. Chem. Soc.* **1995**, *117* (19), 5179−5197.
(58) Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; Goddard, W. A., III; Skiff, W. M. UFF, a Full Periodic Table Force Field for Molecular Mechanics and Molecular Dynamics Simulations. *J. Am. Chem. Soc.* **1992**, *114* (25), 10024−10035.
(59) van Duin, A. C. T.; Dasgupta, S.; Lorant, F.; Goddard, W. A., III ReaxFF: A Reactive Force Field for Hydrocarbons. *J. Phys. Chem. A* **2001**, *105* (41), 9396−9409.
(60) Mayo, S. L.; Olafson, B. D.; Goddard, W. A. DREIDING: A Generic Force Field for Molecular Simulations. *J. Phys. Chem.* **1990**, *94* (26), 8897−8909.
(61) Shi, S.; Yan, L.; Yang, Y.; Fisher-Shaulsky, J.; Thacher, T. An Extensible and Systematic Force Field, ESFF, for Molecular Modeling of Organic, Inorganic, and Organometallic Systems. *J. Comput. Chem.* **2003**, *24* (9), 1059−1076.
(62) Comba, P.; Remenyi, R. Inorganic and Bioinorganic Molecular Mechanics Modeling—the Problem of the Force Field Parameterization. *Coord. Chem. Rev.* **2003**, *238−239*, 9−20.
(63) Daw, M. S.; Foiles, S. M.; Baskes, M. I. The Embedded-Atom Method: A Review of Theory and Applications. *Mater. Sci. Reports* **1993**, *9* (7−8), 251−310.
(64) Lee, B.-J.; Baskes, M. I. Second Nearest-Neighbor Modified Embedded-Atom-Method Potential. *Phys. Rev. B* **2000**, *62* (13), 8564.
(65) Wei, J.; Zhou, W.; Li, S.; Shen, P.; Ren, S.; Hu, A.; Zhou, W. Modified Embedded Atom Method Potential for Modeling the Thermodynamic Properties of High Thermal Conductivity Beryllium Oxide. *ACS Omega* **2019**, *4* (4), 6339−6346.
(66) Stuart, S. J.; Tutein, A. B.; Harrison, J. A. A Reactive Potential for Hydrocarbons with Intermolecular Interactions. *J. Chem. Phys.* **2000**, *112* (14), 6472.
(67) Tersoff, J. Modeling Solid-State Chemistry: Interatomic Potentials for Multicomponent Systems. *Phys. Rev. B* **1989**, *39* (8), 5566.
(68) MacKerell, A. D., Jr.; Bashford, D.; Bellott, M.; Dunbrack, R. L., Jr.; Evanseck, J. D.; Field, M. J.; Fischer, S.; Gao, J.; Guo, H.; Ha, S.; et al. All-Atom Empirical Potential for Molecular Modeling and Dynamics Studies of Proteins †. *J. Phys. Chem. B* **1998**, *102* (18), 3586−3616.
(69) Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J. Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *J. Am. Chem. Soc.* **1996**, *118* (45), 11225−11236.
(70) Schmid, N.; Eichenberger, A. P.; Choutko, A.; Riniker, S.; Winger, M.; Mark, A. E.; van Gunsteren, W. F. Definition and Testing of the GROMOS Force-Field Versions 54A7 and 54B7. *Eur. Biophys. J.* **2011**, *40* (7), 843−856.
(71) Lins, R. D.; Hünenberger, P. H. A New GROMOS Force Field for Hexopyranose-Based Carbohydrates. *J. Comput. Chem.* **2005**, *26* (13), 1400−1412.
(72) Kaminski, G. A.; Friesner, R. A.; Tirado-Rives, J.; Jorgensen, W. L. Evaluation and Reparametrization of the OPLS-AA Force Field for

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634

Proteins via Comparison with Accurate Quantum Chemical Calculations on Peptides. *J. Phys. Chem. B* **2001**, *105* (28), 6474−6487.
(73) Siu, S. W. I.; Pluhackova, K.; Böckmann, R. A. Optimization of the OPLS-AA Force Field for Long Hydrocarbons. *J. Chem. Theory Comput.* **2012**, *8* (4), 1459−1470.
(74) Dickson, C. J.; Rosso, L.; Betz, R. M.; Walker, R. C.; Gould, I. R. GAFFlipid: A General Amber Force Field for the Accurate Molecular Dynamics Simulation of Phospholipid. *Soft Matter* **2012**, *8* (37), 9617−9627.
(75) Seabra, G. d. M.; Walker, R. C.; Elstner, M.; Case, D. A.; Roitberg, A. E. Implementation of the SCC-DFTB Method for Hybrid QM/MM Simulations within the Amber Molecular Dynamics Package†. *J. Phys. Chem. A* **2007**, *111* (26), 5655−5664.
(76) Vanommeslaeghe, K.; Hatcher, E.; Acharya, C.; Kundu, S.; Zhong, S.; Shim, J.; Darian, E.; Guvench, O.; Lopes, P.; Vorobyov, I.; et al. CHARMM General Force Field (CGenFF): A Force Field for Drug-like Molecules Compatible with the CHARMM All-Atom Additive Biological Force Fields. *J. Comput. Chem.* **2010**, *31* (4), 671−690.
(77) Naserifar, S.; Brooks, D.; Goddard, W.; Cvicek, V. Polarizable Charge Equilibration Model for Predicting Accurate Electrostatic Interactions in Molecules and Solids. *J. Chem. Phys.* **2017**, *146* (12), 124117.
(78) Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and Testing of a General Amber Force Field. *J. Comput. Chem.* **2004**, *25* (9), 1157−1174.
(79) Ponder, J. W.; Case, D. A. FORCE FIELDS FOR PROTEIN SIMULATIONS. *Adv. Protein Chem.* **2003**, *66* (1), 27−85.
(80) Tian, C.; Kasavajhala, K.; Belfon, K. A. A.; Raguette, L.; Huang, H.; Migues, A. N.; Bickel, J.; Wang, Y.; Pincay, J.; Wu, Q.; et al. Ff19SB: Amino-Acid-Specific Protein Backbone Parameters Trained against Quantum Mechanics Energy Surfaces in Solution. *J. Chem. Theory Comput.* **2020**, *16* (1), 528−552.
(81) Cheatham, T. E.; Case, D. A. Twenty-Five Years of Nucleic Acid Simulations. *Biopolymers* **2013**, *99* (12), 969−977.
(82) Mackerell, A. D.; Banavali, N. K. All-Atom Empirical Force Field for Nucleic Acids: II. Application to Molecular Dynamics Simulations of DNA and RNA in Solution MACKERELL AND BANAVALI. *J. Comput. Chem.* **2000**, *21* (2), 105−120.
(83) Feller, S. E.; Gawrisch, K.; MacKerell, A. D. Polyunsaturated Fatty Acids in Lipid Bilayers: Intrinsic and Environmental Contributions to Their Unique Physical Properties. *J. Am. Chem. Soc.* **2002**, *124* (2), 318−326.
(84) Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J. Development and Testing of the OPLS All-Atom Force Field on Conformational Energetics and Properties of Organic Liquids. *J. Am. Chem. Soc.* **1996**, *118* (45), 11225−11236.
(85) Sambasivarao, S. V.; Acevedo, O. Development of OPLS-AA Force Field Parameters for 68 Unique Ionic Liquids. *J. Chem. Theory Comput.* **2009**, *5* (4), 1038−1050.
(86) Roos, K.; Wu, C.; Damm, W.; Reboul, M.; Stevenson, J. M.; Lu, C.; Dahlgren, M. K.; Mondal, S.; Chen, W.; Wang, L.; et al. OPLS3e: Extending Force Field Coverage for Drug-Like Small Molecules. *J. Chem. Theory Comput.* **2019**, *15* (3), 1863−1874.
(87) Nielson, K. D.; Van Duin, A. C. T.; Oxgaard, J.; Deng, W. Q.; Goddard, W. A. Development of the ReaxFF Reactive Force Field for Describing Transition Metal Catalyzed Reactions, with Application to the Initial Stages of the Catalytic Formation of Carbon Nanotubes. *J. Phys. Chem. A* **2005**, *109* (3), 493−499.
(88) Senftle, T. P.; Hong, S.; Islam, M. M.; Kylasa, S. B.; Zheng, Y.; Shin, Y. K.; Junkermeier, C.; Engel-Herbert, R.; Janik, M. J.; Aktulga, H. M.; et al. The ReaxFF Reactive Force-Field: Development, Applications and Future Directions. *npj Comput. Mater.* **2016**, *2* (1), 1−14.
(89) Lopes, P. E. M.; Huang, J.; Shim, J.; Luo, Y.; Li, H.; Roux, B.; Mackerell, A. D. Polarizable Force Field for Peptides and Proteins Based on the Classical Drude Oscillator. *J. Chem. Theory Comput.* **2013**, *9* (12), 5430−5449.
(90) Jiang, W.; Hardy, D. J.; Phillips, J. C.; MacKerell, A. D.; Schulten, K.; Roux, B. High-Performance Scalable Molecular Dynamics Simulations of a Polarizable Force Field Based on Classical Drude Oscillators in NAMD. *J. Phys. Chem. Lett.* **2011**, *2* (2), 87−92.
(91) Lemkul, J. A.; Huang, J.; Roux, B.; Mackerell, A. D. An Empirical Polarizable Force Field Based on the Classical Drude Oscillator Model: Development History and Recent Applications. *Chem. Rev.* **2016**, *116* (9), 4983−5013.
(92) Christen, M.; Hünenberger, P. H.; Bakowies, D.; Baron, R.; Bürgi, R.; Geerke, D. P.; Heinz, T. N.; Kastenholz, M. A.; Kräutler, V.; Oostenbrink, C.; et al. The GROMOS Software for Biomolecular Simulation: GROMOS05. *J. Comput. Chem.* **2005**, *26* (16), 1719−1751.
(93) Kunz, A. P. E.; Allison, J. R.; Geerke, D. P.; Horta, B. A. C.; Hünenberger, P. H.; Riniker, S.; Schmid, N.; Van Gunsteren, W. F. New Functionalities in the GROMOS Biomolecular Simulation Software. *J. Comput. Chem.* **2012**, *33* (3), 340−353.
(94) Riniker, S.; Christ, C. D.; Hansen, H. S.; Hünenberger, P. H.; Oostenbrink, C.; Steiner, D.; Van Gunsteren, W. F. Calculation of Relative Free Energies for Ligand-Protein Binding, Solvation, and Conformational Transitions Using the GROMOS Software. *J. Phys. Chem. B* **2011**, *115* (46), 13570−13577.
(95) Leontyev, I. V.; Stuchebrukhov, A. A. Polarizable Mean-Field Model of Water for Biological Simulations with Amber and Charmm Force Fields. *J. Chem. Theory Comput.* **2012**, *8* (9), 3207.
(96) Ren, P.; Ponder, J. W. Consistent Treatment of Inter- and Intramolecular Polarization in Molecular Mechanics Calculations. *J. Comput. Chem.* **2002**, *23* (16), 1497−1506.
(97) Ponder, J. W.; Wu, C.; Ren, P.; Pande, V. S.; Chodera, J. D.; Schnieders, M. J.; Haque, I.; Mobley, D. L.; Lambrecht, D. S.; DiStasio, R. A.; et al. Current Status of the AMOEBA Polarizable Force Field. *J. Phys. Chem. B* **2010**, *114* (8), 2549−2564.
(98) Zhang, C.; Lu, C.; Jing, Z.; Wu, C.; Piquemal, J.-P.; Ponder, J. W.; Ren, P. AMOEBA Polarizable Atomic Multipole Force Field for Nucleic Acids. *J. Chem. Theory Comput.* **2018**, *14* (4), 2084−2108.
(99) Rathnayake, P. V. G. M.; Bernardi, S.; Widmer-Cooper, A. Evaluation of the AMOEBA Force Field for Simulating Metal Halide Perovskites in the Solid State and in Solution. *J. Chem. Phys.* **2020**, *152* (2), 024117.
(100) Narayanan, B.; Sasikumar, K.; Mei, Z. G.; Kinaci, A.; Sen, F. G.; Davis, M. J.; Gray, S. K.; Chan, M. K. Y.; Sankaranarayanan, S. K. R. S. Development of a Modified Embedded Atom Force Field for Zirconium Nitride Using Multi-Objective Evolutionary Optimization. *J. Phys. Chem. C* **2016**, *120* (31), 17475−17483.
(101) Boyd, P. G.; Moosavi, S. M.; Witman, M.; Smit, B. Force-Field Prediction of Materials Properties in Metal-Organic Frameworks. *J. Phys. Chem. Lett.* **2017**, *8* (2), 357−363.
(102) Sasaki, K.; Yamashita, T. Modification and Validation of the DREIDING Force Field for Molecular Liquid Simulations (DREIDING-UT). *J. Chem. Inf. Model.* **2021**, *61* (3), 1172−1179.
(103) Marrink, S. J.; Risselada, H. J.; Yefimov, S.; Tieleman, D. P.; De Vries, A. H. The MARTINI Force Field: Coarse Grained Model for Biomolecular Simulations. *J. Phys. Chem. B* **2007**, *111* (27), 7812−7824.
(104) Yoo, A. B.; Jette, M. A.; Grondona, M. SLURM: Simple Linux Utility for Resource Management. *Job Scheduling Strategies for Parallel Processing; Lecture Notes in Computer Science; Springer Berlin Heidelberg: Berlin, Heidelberg* **2003**, *2862*, 44−60.
(105) Thompson, A. P.; Aktulga, H. M.; Berger, R.; Bolintineanu, D. S.; Brown, W. M.; Crozier, P. S.; in 't Veld, P. J.; Kohlmeyer, A.; Moore, S. G.; Nguyen, T. D.; et al. LAMMPS - a Flexible Simulation Tool for Particle-Based Materials Modeling at the Atomic, Meso, and Continuum Scales. *Comput. Phys. Commun.* **2022**, *271*, 108171.
(106) Phillips, J. C.; Hardy, D. J.; Maia, J. D. C.; Stone, J. E.; Ribeiro, J. V.; Bernardi, R. C.; Buch, R.; Fiorin, G.; Hénin, J.; Jiang, W.; et al. Scalable Molecular Dynamics on CPU and GPU Architectures with NAMD. *J. Chem. Phys.* **2020**, *153* (4), 044130.

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634

(107) Berendsen, H. J. C.; van der Spoel, D.; van Drunen, R. GROMACS: A Message-Passing Parallel Molecular Dynamics Implementation. *Comput. Phys. Commun.* **1995**, *91* (1−3), 43−56.
(108) Brooks, B. R.; Brooks, C. L.; Mackerell, A. D.; Nilsson, L.; Petrella, R. J.; Roux, B.; Won, Y.; Archontis, G.; Bartels, C.; Boresch, S.; et al. CHARMM: The Biomolecular Simulation Program. *J. Comput. Chem.* **2009**, *30* (10), 1545.
(109) Salomon-Ferrer, R.; Case, D. A.; Walker, R. C. An Overview of the Amber Biomolecular Simulation Package. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2013**, *3* (2), 198−210.
(110) Case, D. A.; Cheatham, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The Amber Biomolecular Simulation Programs. *J. Comput. Chem.* **2005**, *26* (16), 1668−1688.
(111) Rackers, J. A.; Wang, Z.; Lu, C.; Laury, M. L.; Lagardère, L.; Schnieders, M. J.; Piquemal, J. P.; Ren, P.; Ponder, J. W. Tinker 8: Software Tools for Molecular Design. *J. Chem. Theory Comput.* **2018**, *14* (10), 5273−5289.
(112) Toukan, K.; Rahman, A. Molecular-Dynamics Study of Atomic Motions in Water. *Phys. Rev. B* **1985**, *31* (5), 2643.
(113) Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. Semianalytical Treatment of Solvation for Molecular Mechanics and Dynamics. *J. Am. Chem. Soc.* **1990**, *112* (16), 6127−6129.
(114) Jorgensen, W. L.; Tirado-Rives, J. Molecular Modeling of Organic and Biomolecular Systems Using BOSS and MCPRO. *J. Comput. Chem.* **2005**, *26* (16), 1689−1700.
(115) Marelius, J.; Kolmodin, K.; Feierberg, I.; Åqvist, J. Q: A Molecular Dynamics Program for Free Energy Calculations and Empirical Valence Bond Simulations in Biomolecular Systems. *J. Mol. Graph. Model.* **1998**, *16* (4−6), 213−225.
(116) Bowers, K. J.; Chow, E.; Xu, H.; Dror, R. O.; Eastwood, M. P.; Gregersen, B. A.; Klepeis, J. L.; Kolossvary, I.; Moraes, M. A.; Sacerdoti, F. D. Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters. *SC '06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*; IEEE, 2006; p 43;
(117) *Materials Studio*; Dassault Systèmes BIOVIA: San Diego, CA, 2018.
(118) Anderson, J. A.; Eric Irrgang, M.; Glotzer, S. C. Scalable Metropolis Monte Carlo for Simulation of Hard Shapes. *Comput. Phys. Commun.* **2016**, *204*, 21−30.
(119) Anderson, J. A.; Lorenz, C. D.; Travesset, A. General Purpose Molecular Dynamics Simulations Fully Implemented on Graphics Processing Units. *J. Comput. Phys.* **2008**, *227* (10), 5342−5359.
(120) Franz, D. M.; Belof, J. L.; McLaughlin, K.; Cioce, C. R.; Tudor, B.; Hogan, A.; Laratelli, L.; Mulcair, M.; Mostrom, M.; Navas, A.; et al. MPMC and MCMD: Free High-Performance Simulation Software for Atomistic Systems. *Adv. Theory Simulations* **2019**, *2* (11), 1900113.
(121) Jung, J.; Nishima, W.; Daniels, M.; Bascom, G.; Kobayashi, C.; Adedoyin, A.; Wall, M.; Lappala, A.; Phillips, D.; Fischer, W.; et al. Scaling Molecular Dynamics beyond 100,000 Processor Cores for Large-Scale Biophysical Simulations. *J. Comput. Chem.* **2019**, *40* (21), 1919−1930.
(122) Klein, M. L.; Shinoda, W. Large-Scale Molecular Dynamics Simulations of Self-Assembling Systems. *Science (80-.)* **2008**, *321* (5890), 798−800.
(123) Laio, A.; Parrinello, M. Escaping Free-Energy Minima. *Proc. Natl. Acad. Sci. U. S. A.* **2002**, *99* (20), 12562.
(124) Barducci, A.; Bussi, G.; Parrinello, M. Well-Tempered Metadynamics: A Smoothly-Converging and Tunable Free-Energy Method. *Phys. Rev. Lett.* **2008**, *100* (2), 020603−1−020603−4.
(125) Torrie, G. M.; Valleau, J. P. Nonphysical Sampling Distributions in Monte Carlo Free-Energy Estimation: Umbrella Sampling. *J. Comput. Phys.* **1977**, *23* (2), 187−199.
(126) Shen, J.; Fujita, K.; Matsumoto, T.; Hongo, C.; Misaki, M.; Ishida, K.; Mori, A.; Nishino, T. Mechanical, Thermal, and Electrical Properties of Flexible Polythiophene with Disiloxane Side Chains. *Macromol. Chem. Phys.* **2017**, *218* (19), 1700197.
(127) Zwanzig, R. W. High-Temperature Equation of State by a Perturbation Method. I. Nonpolar Gases. *J. Chem. Phys.* **1954**, *22* (8), 1420−1426.
(128) Bolhuis, P. G.; Chandler, D.; Dellago, C.; Geissler, P. L. TRANSITION PATH SAMPLING: Throwing Ropes Over Rough Mountain Passes, in the Dark. *Annu. Rev. Phys. Chem.* **2002**, *53*, 291−318.
(129) Voter, A. F. Hyperdynamics: Accelerated Molecular Dynamics of Infrequent Events. *Phys. Rev. Lett.* **1997**, *78* (20), 3908−3911.
(130) Qi, R.; Wei, G.; Ma, B.; Nussinov, R. Replica Exchange Molecular Dynamics: A Practical Application Protocol with Solutions to Common Problems and a Peptide Aggregation and Self-Assembly Example. *Methods Mol. Biol.* **2018**, *1777*, 101.
(131) Minary, P.; Tuckerman, M. E.; Martyna, G. J. Long Time Molecular Dynamics for Enhanced Conformational Sampling in Biomolecular Systems. *Phys. Rev. Lett.* **2004**, *93* (15), 150201.
(132) Barducci, A.; Bussi, G.; Parrinello, M. Well-Tempered Metadynamics: A Smoothly Converging and Tunable Free-Energy Method. *Phys. Rev. Lett.* **2008**, *100*, 020603.
(133) Laio, A.; Gervasio, F. L. Metadynamics: A Method to Simulate Rare Events and Reconstruct the Free Energy in Biophysics, Chemistry and Material Science. *Rep. Prog. Phys.* **2008**, *71* (12), 126601.
(134) Deighan, M.; Bonomi, M.; Pfaendtner, J. Efficient Simulation of Explicitly Solvated Proteins in the Well-Tempered Ensemble. *J. Chem. Theory Comput.* **2012**, *8* (7), 2189−2192.
(135) Fu, Y.; Bernasconi, L.; Liu, P. Ab Initio Molecular Dynamics Simulations of the SN1/SN2Mechanistic Continuum in Glycosylation Reactions. *J. Am. Chem. Soc.* **2021**, *143* (3), 1577−1589.
(136) Fratev, F.; Sirimulla, S. An Improved Free Energy Perturbation FEP+ Sampling Protocol for Flexible Ligand-Binding Domains. *Sci. Rep.* **2019**, *9* (1), 16829.
(137) Moraca, F.; Negri, A.; De Oliveira, C.; Abel, R. Application of Free Energy Perturbation (FEP+) to Understanding Ligand Selectivity: A Case Study to Assess Selectivity between Pairs of Phosphodiesterases (PDE's). *J. Chem. Inf. Model.* **2019**, *59*, 2729−2740.
(138) Wang, L.; Chambers, J.; Abel, R. Protein-Ligand Binding Free Energy Calculations with FEP. *Methods Mol. Biol.* **2019**, *2022*, 201−232.
(139) Shih, A. Y.; Hack, M.; Mirzadegan, T. Impact of Protein Preparation on Resulting Accuracy of FEP Calculations. *J. Chem. Inf. Model.* **2020**, *60* (11), 5287−5289.
(140) Jespers, W.; Isaksen, G. V.; Andberg, T. A. H.; Vasile, S.; Van Veen, A.; Åqvist, J.; Brandsdal, B. O.; Gutiérrez-De-Terán, H. QresFEP: An Automated Protocol for Free Energy Calculations of Protein Mutations in Q. *J. Chem. Theory Comput.* **2019**, *15* (10), 5461−5473.
(141) Hollingsworth, S. A.; Dror, R. O. Molecular Dynamics Simulation for All. *Neuron* **2018**, *99* (6), 1129.
(142) Hospital, A.; Goñi, J. R.; Orozco, M.; Gelpí, J. L. Molecular Dynamics Simulations: Advances and Applications. *Adv. Appl. Bioinf. Chem.* **2015**, *8* (1), 37−47.
(143) Introduction to Monte Carlo Methods in Chemistry; https://people.chem.ucsb.edu/kahn/kalju/MonteCarlo_4.html (accessed Jan 13, 2022).
(144) Cuppen, H. M.; Karssemeijer, L. J.; Lamberts, T. The Kinetic Monte Carlo Method as a Way To Solve the Master Equation for Interstellar Grain Chemistry. *Chem. Rev.* **2013**, *113* (12), 8840−8871.

https://doi.org/10.1021/acs.chemmater.2c00292
Chem. Mater. 2022, 34, 7620−7634