Best Practices for Foundations in Molecular Simulations : v0.1

Efrem Braun[1], Justin Gilmer[2], Heather B. Mayes[3], David L. Mobley[4], Jacob I. Monroe[5], Samarjeet Prasad[6], Daniel M. Zuckerman[7]

[1]University of California, Berkeley; [2]Vanderbilt University; [3]University of Michigan, Ann Arbor; [4]University of California, Irvine; [5]University of California, Santa Barbara; [6]National Institutes of Health; [7]Oregon Health and Science University

This LiveCoMS document is maintained online on GitHub at https://github.com/MobleyLab/basic_simulation_training; to provide feedback, suggestions, or help improve it, please visit the GitHub repository and participate via the issue tracker.

This version dated August 1, 2018

Abstract This document provides a starting point for approaching molecular simulations, guiding beginning practitioners to what issues they need to know about before and while starting their first simulations, and why those issues are so critical. This document makes no claims to provide an adequate introduction to the subject on its own. Instead, our goal is to help people know what issues are *critical* before beginning, and to provide references to good resources on those topics. We also provide a checklist of key issues to consider before and while setting up molecular simulations which may serve as a foundation for other best practices documents.

***For correspondence:**
efrem.braun@berkeley.edu (EB); justin.b.gilmer@vanderbilt.edu (JG); hbmayes@umich.edu (HM); dmobley@mobleylab.org (DLM); jimonroe@umail.ucsb.edu (JIM); p.samar.j@gmail.com (SP); zuckermd@ohsu.edu (DMZ)

Todo list

1 Introduction

Molecular simulation techniques play a very important role in our quest to understand and predict the properties, structure, and function of molecular systems, and are a key tool as we seek to enable predictive molecular design. Simulation methods are extremely useful for studying the structure and dynamics of complex systems that are too complicated for pen and paper theory and helping interpret experimental data in terms of molecular motions, as well as (increasingly) for quantitative prediction of properties of use in molecular design and other applications [**? ? ? ? ?**].

The basic idea of any molecular simulation method is straightforward; a particle-based description of the system under investigation is constructed and then the system is propagated by either deterministic or probabilistic rules to generate a trajectory describing its evolution over the course of the simulation [**? ?**]. Relevant properties can be calculated for each "snapshot" (a stored configuration of the system, also called a "frame") and averaged over the the entire trajectory to compute estimates of desired properties.

Depending on how the system is propagated, molecular simulation methods can be divided into two main categories: Molecular Dynamics (MD) and Monte Carlo (MC). With MD methods, the equations of motion are numerically integrated to generate a dynamical trajectory of the system. MD simulations can be used for investigating structural, dynamic, and thermodynamic properties of the system. With MC methods, probabilistic rules are used to generate a new configuration from the present configuration and this process is repeated to generate a sequence of states that can be used to calculate structural and thermodynamic properties but not dynamical properties; indeed, MC simulations lack any concept of time. Thus, the "dynamics" produced by an MC method are not the temporal dynamics of the system, but the ensemble of configurations that reflect those that could be dynamically sampled. This foundational document will focus on the concepts needed to carry out correct MD simulations that utilize good practices. Many, but not all, of the concepts here are also useful for MC simulations and apply there as well. However, there are a sufficient number of key differences, which are outside the scope of this current document.

Either method can be carried out with different underlying physical theories to describe the particle-based model of the system under investigation. If a quantum mechanics (QM) description of matter is used, electrons are explicitly represented in the model and interaction energy is calculated by solving the electronic structure of the molecules in the system with no (or few) empirical parameters, but with various approximations to the physics for tractability. In a molecular mechanics (MM) description, the molecules are represented by particles representing atoms or groups of atoms. Each atom may be assigned an electric charge and a potential energy function with a large number of empirical parameters (fitted to experiment, QM, or other data) used to calculate non-bonded and bonded interactions. Unless otherwise specified, MD simulations employ MM force fields, which calculate the forces that determine the system dynamics. MM simulations are much faster than quantum simulations, making them the methods of choice for vast majority of molecular simulation studies on biomolecular systems in the condensed phase. However, typically, they are of lower-accuracy than QM simulations and cannot simulate bond rearrangements. QM simulations may be too computationally expensive to allow simulations of the time and length scales required to describe the system of interest [**?**]. The size of the system amenable for to QM simulation also depends on what type of method is chosen, from high-level ab initio methods to semi-empirical methods; discussion of these methods are outside the scope of this article, and useful references are separately available [**?**]. Computational resources available are also an important consideration in deciding whether QM simulations are tractable. Roughly, QM simulations might be tractable with hundreds of atoms or fewer, while MD simulations routinely have tens or hundreds of thousands of atoms in the system. Much above that level, coarse-graining methods are used. They reduce resolution and computational cost. Although many of the approaches for atomistic simulations discussed here can apply to coarse-grained simulations, such simulations are not the focus of this paper and we will not discuss how coarse-grained simulations are initially built.

Speed is a particular concern when describing condensed phase systems, as we are often interested in the properties of molecules (even biomacromolecules) in solution, meaning that systems will consist of thousands to hundreds of thousands or millions of atoms. While system size alone does not require a classical description, if we are interested in calculations of thermodynamic properties like free energy at finite (often laboratory) temperatures, these include substantial entropic contributions (as further discussed below) meaning that fluctuations and correlations of motions within the system affect computed properties, meaning that simulations must not just sample single optimal states but instead must sample the correct distribution of states – requiring simulations of some length. Furthermore, many systems of interest, such as polymers (biological and otherwise) have slow motions that must be captured for accurate calculation of properties. For example, for proteins, relevant timescales span from nanoseconds to seconds or more, and even rearrangements of buried amino acid sidechains can in some cases take microseconds or more, with larger conformational changes and protein folding taking even longer [**? ?**]. Re-

cent hardware innovations have made microsecond-length simulations for biological systems of 50-100,000 atoms relatively routine, and herculean efforts have pushed the longest simulations out past the millisecond range. However, the field would like to reach even longer timescales, meaning that switching to a more detailed energy model is only done with some trepidation because slower energy evaluations mean less time available for sampling. Thus the need for speed limits the use of quantum mechanical descriptions.

Thus, for the rest of this document we will restrict ourselves to classical MD.

One other important note is that, within classical molecular simulations, bond breaking and forming is generally not allowed (with notable exceptions such as reactive force fields), meaning that the overall topology or chemistry of a system will remain constant as a function of time. That is, the particles comprising the system move around, but the chemical identity of each molecule in the system remains a constant over the course of the simulation (with only partial exceptions, such as the case of constant pH simulations [**?**]). This also means that the notion of pH in molecular simulations primarily refers to the selection of *fixed* protonation states for the components of the system.

Here, we first discuss the scope of this document, then go over some of the fundamental concepts or science topics which provide the underpinnings of molecular simulations, giving references for further reading. Then, we introduce a variety of basic simulation concepts and terminology, with additional links to further reading. Our goal is not to cover all topics, but to provide some guidance as to the critical issues which must be considered.

2 Scope of this document

There are several excellent textbooks on classical simulation methods; some we have found particularly helpful are Allen and Tildesley's "Computer Simulations of Liquids" [**?**], Leach's "Molecular Modelling" [**?**], and Frenkel and Smit's "Understanding Molecular Simulations" [**?**], though there are many other sources. Tuckerman's "Statistical Mechanics: Theory and Molecular Simulation" [**?**] may be helpful to a more advanced audience.

In principle, anyone with adequate prior knowledge should be able to pick up one of these books and learn the required skills to perform molecular simulations, perhaps with help from a good statistical mechanics and thermodynamics book or two. In practice, due to the interdisciplinary and somewhat technical nature of this field, many newcomers may find it difficult and time consuming to understand all the methodological issues involved in a simulation study. The goal of this document is to introduce a new practitioner to some key basic concepts and bare minimum scientific knowledge required for correct execution of these methods. We also provide a basic set of "best practices" that can be used to avoid common errors, missteps and confusion in elementary molecular simulations work. This document is not meant as a full introduction to the area; rather, it is intended to help guide further study, and to provide a foundation for other more specialized best-practices documents focusing on particular simulation areas.

Modern implementations of classical simulations also rely on a large body of knowledge from the fields of computer science and numerical methods, which will not be covered in detail here.

3 Science topics

A variety of fields provide the foundation for our simulation methods and analysis of the data produced by these methods. A new practitioner does not have to be an expert of all these fields but needs to understand some key concepts from each of these disciplines. In this section, we survey some topics that we believe even basic users of molecular simulations need to grasp, with suggestions for further reading on these subjects, as a preface for Section 4. In each subsection, we begin by highlighting some of the critical topics from the corresponding area, then describe what these are and why they are important to molecular simulations.

3.1 Classical mechanics

3.1.1 Key concepts

Critical concepts from classical mechanics include:

- Newton's equations of motion and constants of motion
- Hamilton's equations
- Point particles and rigid bodies
- Holonomic constraints

Molecular simulation methods work on many particle systems following the rules of classical mechanics. Basic knowledge of key concepts of classical mechanics is important for understanding simulation methods. Here, we will assume you are already familiar with Newtonian mechanics.

Classical molecular models typically consist of point particles carrying mass and electric charge, as well as potentially additional interactions such as van der Waals interactions and bonded interactions of various types. Sometimes it is much more efficient to freeze the internal degrees of freedoms and treat the molecule as a rigid body where the particles do not change their relative orientation as the whole body moves; this is commonly done, for example, for rigid models of the water molecule. Due to the high frequency of the O-H vibrations, accurately treating water classically would require

solving the equations of motion with a very small timestep, so for computational efficiency water is often instead treated as a rigid body. Keeping specified objects rigid in a simulation involves applying holonomic constraints, where the rigidity is defined by imposing a minimal set of fixed bond lengths and angles through iterative procedures during the numerical integration of the equation of motion (see Section 4.6 for more on constraints and integrators). It is important to understand the concept of point particles, rigid bodies and constraints.

Classical mechanics has several mathematical formulations — namely the Newtonian, Hamiltonian and Lagrangian formulations. These formulations are equivalent, but for certain applications one formulation can be more appropriate than the other. Many simulation methods use the Hamiltonian formulation and therefore basic knowledge of Hamiltonian mechanics is essential if you wish to understand the details of simulation methods.

Classical mechanics has several conserved quantities and simulators should be familiar with these, for example, the total energy of a system is a constant of motion. These concepts play very important role in development and proper implementation of simulation methods. For example, a particularly straightforward check of the correctness of an MD code is to test the quality of the energy conservation.

Most books on molecular simulations have a short discussions or appendices on classical mechanics that can serve the purpose of very quick introductions to the basic concepts; Shell's book also has a chapter on simulation methods which covers some of these details [**?**]. A variety of good books on classical mechanics are also available and give further details on these concepts.

3.2 Thermodynamics

3.2.1 Key concepts

A variety of thermodynamic concepts are particularly important for molecular simulations:

- Temperature, pressure, stress
- Internal energy, enthalpy
- Gibbs and Helmholtz free energy
- Entropy

One of the main objectives of molecular simulations is to estimate/predict thermodynamic behavior of real systems as observed in the laboratory. Typically this means we are interested in macroscopic systems, consisting of 10^{23} particles or more (i.e. at least several moles of particles). But properties of interest include not only macroscopic, bulk thermodynamic properties, such as density or heat capacity, but also microscopic properties like specific free energy differences associated with, say, changes in the conformation of a molecule. For this reason, it is important to understand key concepts in thermodynamics, such as temperature, pressure, entropy, internal energy, various forms of free energy, and the relationships between them. Paramount, however, is an understanding of the connection between thermodynamics and statistical mechanics, which allows us to relate macroscopic, experimental measurements to the behavior of the much smaller system that is simulated. This topic involves a variety of subtleties and thus can be a confusing and difficult, so we refer the reader to a more extensive discussion in one of several books [**? ?**].

As an example, consider temperature. In a macroscopic sense, we understand this quantity very intuitively as how hot or cold something is. The laws of thermodynamics provide us with a further abstraction, telling us that this is in fact the derivative of the internal energy with respect to the entropy. This mathematical definition itself is not particularly helpful, but provides a starting point for other derivations. If we want to understand temperature from the point of view of understanding molecular behavior, we finally must turn to statistical mechanics. Since molecular dynamics is mostly used to simulate behavior at the molecular or atomistic level, it is necessary to utilize statistical-mechanical expressions in computing what would be observed as the macroscopic, thermodynamic temperature.

This discussion should not provide the impression that statistical mechanics is more important than thermodynamics. The two are intimately connected and we must rely on both to successfully conduct and obtain information from MD simulations. In particular, thermodynamics provides rigid rules that must be satisfied if we are to faithfully reproduce reality. For instance, if energy is not conserved, the first law is not satisfied and we are for sure simulating a system out of equilibrium (i.e. we are somehow adding or removing energy). In this sense, the laws of thermodynamics provide us rigorous sanity checks in addition to many useful mathematical relations for computing properties. Basic thermodynamic principles thus also dictate proper simulation protocols and associated best practices.

The concept of the thermodynamic limit is particularly important here. Specifically, as the size of a finite system is increased, keeping the particle number density roughly constant, at some point it is said to reach the thermodynamic limit where its behavior is bulk-like and no longer depends on the extent of the system. Thus, small systems will exhibit unique behaviors that reflect their microscopic size, but sufficiently large systems are said to have reached the thermodynamic limit and macroscopic thermodynamics applied. This is due to the fact that the effect of interfaces or boundaries have largely been removed, and, more importantly, that averages of system properties are now over a sufficiently large number of molecules that any instantaneous snapshot

of the system roughly corresponds to average behavior (i.e. fluctuations in properties become negligible with increasing system size).

Although we usually think of thermodynamics applying macroscopically and statistical mechanics applying on the microsopic level, it is important to remember that the laws of thermodynamics still hold *on average* regardless of the length scale. That is, a molecule in contact with a thermal bath will exchange energy with the bath, but its average energy is a well-defined constant. This allows us to define thermodynamic quantities associated with microscopic events, such as the binding of a ligand to a protein. This is of course useful because it allows us to assign molecular meaning to a well-defined thermodynamic processes that can only be indirectly probed by experiment. Importantly, as long as we have carefully defined our ensemble and thermodynamic path, we can apply the powerful relationships of thermodynamics to more easily calculate many properties of interest. For instance, one may use molecular dynamics to efficiently numerically integrate the Clapeyron equation and construct equations of state along phase coexistence curves [**? ?**].

3.2.2 Books

Equilibrium thermodynamics is taught in most undergraduate programs in physics, chemistry, biochemistry and various engineering disciplines. Depending on the background, the practitioner can choose one or more of the following books to either learn or refresh their basic knowledge of thermodynamics. Here are some works we find particularly helpful:

- Atkins and De Paula's "Physical Chemistry" [**?**], chapters 1 to 4.
- McQuarrie and Simon's extensive work, "Physical Chemistry: A Molecular Approach" [**?**]
- Dill's "Molecular Driving Forces" [**?**]
- Kittel and Kroemer's "Thermal Physics" [**?**]
- Shell [**?**]: Chapters 1-15.

3.3 Classical statistical mechanics

3.3.1 Key concepts

Key concepts from statistical mechanics are particularly important and prevalent in molecular simulations:

- Fluctuations
- Definitions of various ensembles
- Time averages and ensemble averages
- Equilibrium versus non-equilibrium

DLM: This list is bothering me because it is longer than the others, has more statements in it, and doesn't totally connect with what's in this section. Not sure what to do with it. JIM: I've proposed changes. Helpful or not?

Traditional discussions of classical statistical mechanics, especially concise ones, tend to focus first or primarily on macroscopic thermodynamics and microscopic *equilibrium* behavior based on the Boltzmann factor, which tells us that configurations $\mathbf{r}^N$ occur with (relative) probability $\exp[-U(\mathbf{r}^N)/k_BT]$, based on potential energy function U and temperature T in Kelvin units. Dynamical phenomena and their connection to equilibrium tend to be treated later in discussion, if at all. But in both fundamental and practical ways, this ordering is wrong. Think Arrhenius first, then Boltzmann.

MD simulation, like nature itself, runs dynamics. Any equilibrium phenomena may (or may not) occur as a consequence and equilibrium behavior is hardly automatic [**?**]. In fact, based on current and foreseeable computational technology, it is much safer to assume that your simulation will not exhibit equilibrium behavior. However, an MD simulation is guaranteed to exhibit dynamical behavior.

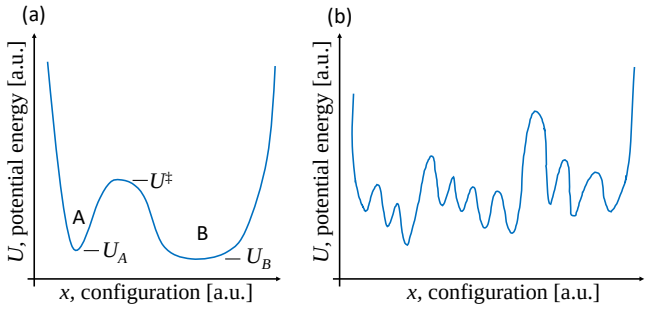

Figure 1. Energy landscapes. (a) A highly simplified landscape used to illustrate rate concepts and (b) a schematic of a complex landscape with numerous minima and ambiguous state boundaries.

The key dynamical concept to understand is embodied in the twin characteristics of timescales and rates. The two are literally reciprocals of one another. In Fig. 1(a), assume you have started an MD simulation in basin A. The trajectory is likely to remain in that basin for a period of time – the "dwell" timescale – which increases exponentially with the barrier height according to the (reciprocal) Arrhenius factor as $\exp[(U^\ddagger - U_A)/k_BT]$; barriers many times the thermal energy k_BT imply long dwells. The rate k_{AB}, which is the transition probability per unit time, exhibits reciprocal behavior – i.e., $k_{AB} \sim \exp[-(U^\ddagger - U_A)/k_BT]$ according to the traditional Arrhenius factor. Note that all transitions occur in a random, *stochastic* fashion and are not predictable except in terms of average behavior. More detailed discussions of rate constants can be found in numerous textbooks (e.g., [**? ?**]).

Once you have understood that MD behavior reflects system timescales, you must set this behavior in the context of an *extremely* complex energy landscape consisting of almost innumerable minima and barriers, as schematized in Fig. 1(b). Each small basin represents something like a dif-

ferent rotameric state of a protein side chain or perhaps a tiny part of the Ramachandran spaces (backbone phi-psi angles) for one or a few residues. Observing the large-scale function motion of a protein then would require an MD simulation longer than the sum of all the timescales for the necessary hops, bearing in mind that numerous stochastic reversals are likely during the simulation. Because functional biomolecular timescales tend to be on μs - ms scales, it is challenging if not impossible to observe them in traditional MD simulations. There are numerous enhanced sampling approaches [**?** **?**] but these are beyond the scope of this discussion and they have their own challenges which often are much harder to diagnose (see [**?**] and https://github.com/dmzuckerman/Sampling-Uncertainty).

What is the connection between MD simulation and equilibrium? The most precise statement we can make is that an MD trajectory is a single sample of a process that is relaxing to equilibrium from the starting configuration [**?** **?**]. *If* the trajectory is long enough, it should sample the equilibrium distribution – where each configuration occurs with frequency proportional to its Boltzmann factor. In such a very long trajectory (only), a time average thus will give the same result as a Boltzmann-factor-weighted, or ensemble, average. We refer to such a system, where the time and ensemble averages are equivialent, as "ergodic." Note that the Boltzmann-factor distribution implies that every configuration has some probability, and so it is unlikely that a single conformation or even a single basin dominates an ensembles. Beware that in a typical MD trajectory it is likely that only a small subset of basins will be sampled well – those most quickly accessible to the initial configuration. It is sometimes suggested that multiple MD trajectories starting structures can aid sampling, but unless the equilibrium distribution is known in advance, the bias from the set of starting structures is simply unknown and harder to diagnose.

A fundamental equilibrium concept that can only be sketched here is the representation of systems of enormous complexity (many thousands, even millions of atoms) in terms of just a small number of coordinates or states. The conformational free energy of a state, e.g., F_A or F_B is a way of expressing the average or summed behavior of all the Boltzmann factors contained in a state: the definition requires that the probability (or population) p^{eq} of a state in equilibrium be proportional to the Boltzmann factor of its conformational free energy: $p_A^{\mathrm{eq}} \sim \exp(-F_A/k_BT)$. Because equilibrium behavior is caused by dynamics, there is a fundamental connection between rates and equilibrium, namely that $p_A^{\mathrm{eq}} k_A B = p_B^{\mathrm{eq}} k_B A$, which is a consequence of "detailed balance". There is a closely related connection for on- and off-rates with the binding equilibrium constant. For a *continuous* coordinate (e.g., the distance between two residues in a protein), the probability-determining free energy is called the "potential of mean force" (PMF): the Boltzmann factor of the PMF gives the relative probability of a given coordinate. Any kind of free energy implicitly includes *entropic* effects; in terms of an energy landscape (Fig. 1), the entropy quantifies the *width* of a basin. These points are discussed in textbooks, as are the differences between free energies for different thermodynamic ensembles – e.g.., F, the Helmholtz free energy, when T is constant, and G , the Gibbs free energy, when both T and pressure are constant – which are not essential to our introduction [**?** **?**].

A final essential topic is the difference between equilibrium and non-equilibrium systems. We noted above that an MD trajectory is not likely to represent the equilibrium ensemble because the trajectory is probably too short. However, in a living cell where there is no shortage of time, biomolecules may exhibit non-equilibrium behavior for a quite different reason – because they are *driven* by the continual addition and removal of (possibly energy-carrying) substrate and product molecules. In this type of non-equilibrium situation, the distribution of configurations will not follow a Boltzmann factor distribution. Specialized simulation approaches are available to study such systems [**?** **?**] but they are not beginner-friendly. Non-equilibrium molecular concepts pertinent to cell biology have been discussed at an introductory level (e.g. http://www.physicallensonthecell.org/).

3.3.2 Books

Books which we recommend as particularly helpful in this area include:

- Reif's "Fundamentals of Statistical and Thermal Physics" [**?**]
- McQuarrie "Statistical Mechanics" [**?**]
- Dill and Bromberg's "Molecular Driving Forces" [**?**]
- Hill's "Statistical Mechanics: Principles and Selected Applications" [**?**]
- Shell's "Thermodynamics and Statistical Mechanics" [**?**]
- Zuckerman's "Statistical Physics of Biomolecules" [**?**]
- Chandler's "Introduction to Modern Statistical Mechanics" [**?**]

3.3.3 Online resources

Several online resources have been particularly helpful to people learning this area, including:

- David Kofke's notes: http://www.eng.buffalo.edu/~kofke/ce530/Lectures/lectures.html
- Scott Shell's notes: https://engineering.ucsb.edu/~shell/che210d/assignments.html

3.4 Classical electrostatics

3.4.1 Key concepts

Key concepts from classical electrostatics include

- The Coulomb interaction and its long-range nature
- Polarizability, dielectric constants, and electrostatic screening
- When and why we need lattice-sum electrostatics and similar approaches

Electrostatic interactions are both some of the longest-range interactions in molecular systems and the strongest, with the interaction (often called "Coulombic" after Coulomb's law) between charged particles falling off as $1/r$ where r is the distance separating the particles. In classical molecular simulations, atoms are typically represented by sites bearing charge in units of fractions of an elementary charge, so atom-atom interactions are thus necessarily long range compared to other interactions in these systems (which fall off a $1/r^3$ or faster). This means atoms or molecules separated by considerable distances can still have quite strong electrostatic interactions, though this also depends on the degree of shielding of the intervening medium (or its relative permittivity or dielectric constant).

The static dielectric constant of a medium, or relative permittivity ϵ_r (relative to that of vacuum), affects the prefactor for the decay of these long range interactions, with interactions falling off as $\frac{1}{\epsilon_r}$. Water has a relatively high relative permittivity or dielectric constant close to 80, whereas non-polar compounds such as n-hexane may have relative permittivities near 2 or even lower. This means that interactions in non-polar media such as non-polar solvents, or potentially even within the relatively non-polar core of a larger molecule such as a protein, are effectively much longer-range even than those in water. The dielectric constant of a medium also relates to the degree of its electrostatic response to the presence of a charge; larger dielectric constants correspond to larger responses to the presence of a nearby charge.

It turns out that atoms and molecules also have their own levels of electrostatic response; particularly, their electron distributions polarize in response to their environment, effectively giving them an internal dielectric constant. This polarization can be modeled in a variety of ways, such as (in fixed charge force fields) building in a fixed amount of polarization which is thought to be appropriate for simulations in a generic "condensed phase" or by explicitly including polarizability via QM or by building it into a simpler, classical model which includes polarizability such as via explicit atomic polarizabilities [**? ?**] or via Drude oscillator-type approaches [**?**], where inclusion of extra particles attached to atoms allows for a type of effective polarization.

Because so many interactions in physical systems involve polarity, and thus significant long-range interactions that decay only slowly with distance, it is important to regard electrostatic interactions as fundamentally long-range interactions. Indeed, contributions to the total energy of a system from distant objects may be even more important in some cases than those from nearby objects. Specifically, since interactions between charges fall off as $1/r$, but the volume of space at a given separation distance increases as r^3, distant interactions can contribute a great deal to the energies and forces in molecular systems. In practice, this means that severe errors often result from neglecting electrostatic interactions beyond some cutoff distance [**? ? ? ? ?**]. Thus, we prefer to include *all* electrostatic interactions, even out to very long range. Once this is decided, it leaves simulators with two main options, only one of which is really viable. First, we can simulate the actual finite (but large) system which is being studied in the lab, including its boundaries. But this is impractical, since macroscopic systems usually include far too many atoms (on the order of at least a mole or more). The remaining option, then, is to apply periodic boundary conditions (see Section 4.2) to tile all of space with repeating copies of the system. Once periodic boundary conditions are set up, defining a periodic lattice, it becomes possible to include all long-range electrostatic interactions via a variety of different types of sums which can be described as "lattice sum electrostatics" or Ewald-type electrostatics [**? ?**] where the periodicity is used to make possible an evaluation of all long range electrostatic interactions, including those of particles with their own periodic images.

In practice, lattice sum electrostatics introduce far fewer and less severe artifacts than do cutoff schemes, so these are used for most classical all-atom simulation algorithms at present. A variety of different efficient lattice-sum schemes are available [**?**]. In general these should be used whenever long range electrostatic interactions are expected to be significant; they may not be necessary in especially nonpolar systems and/or with extremely high dielectric constant solvents where electrostatic interactions are exclusively short range, but in general they should be regarded as standard (see also Section 4.7, below).

3.4.2 Books

On classical electrostatics, we have found the undergraduate-level work by David J. Griffiths, "Introduction to Electrodynamics" [**?**], to be quite helpful. The graduate-level work of Jackson, "Classical Electrodynamics" [**?**], is also considered a classic/standard work, but may prove challenging for those without a background relatively heavy in mathematics.

3.5 Molecular interactions

3.5.1 Key concepts

Molecular simulations are, to a large extent, about molecular interactions, so these are particularly key, including:

- Bonded and nonbonded interactions
- The different types of nonbonded interactions and why they are separated in classical descriptions
- The dividing line between bonded and nonbonded interactions

Key interactions between atoms and within or between molecules are typically thought of as consisting of two main types – bonded and non-bonded interactions. While these arise from similar or related physical effects (ultimately all tracing back to QM and the basic laws of physics) they are typically treated in rather distinct manners in molecular simulations so it is important to consider the two categories separately.

Bonded interactions are those between atoms which are connected, or nearly so, and relating to the bonds connecting these atoms. In typical molecular simulations these consist of bond stretching terms, angle bending terms, and terms describing the rotation of torsional angles. Torsions typically involve four atoms and are often of two types – "proper" torsions, around bonds connecting groups of atoms, and "improper" torsions which involve neighbors of a central atom; these are often used to ensure the appropriate degree of planarity or non-planarity around a particular group (such as planarity of an aromatic ring). It is important to note that the presence of bonded interactions between atoms does not necessarily preclude their also having nonbonded interactions with one another (see discussion of exclusions and 1-4 interactions, below).

Nonbonded interactions between atoms are all interactions which are included in the potential energy of the system *aside* from bonded interactions. Commonly these include at least point-charge Coulomb electrostatic interactions and "non-polar" interactions modeled by the Lennard-Jones potential or another similar potential which describes short range repulsion and weak long-range interaction even between nonpolar atoms. Additional terms may also be included, such as interactions between fixed multipoles, interactions between polarizable sites, or occasionally explicit potentials for hydrogen bonding or other specialized terms. These are particularly common in polarizable force fields such as the AMOEBA model.

Often, the energy functions used by molecular simulations explicitly neglect nonbonded interactions between atoms which are immediately bonded to one another, and atoms which are separated by only one intervening atom, partly to make it easier to ensure that these atoms have preferred geometries dictated by their defined equilibrium lengths/angles regardless of the nonbonded interactions which would otherwise be present. This neglect of especially short range nonbonded interactions between near neighbors is called "exclusion", and energy functions typically specify which interactions are excluded.

The transition to torsions, especially proper torsions, is where exclusions typically end. However, many all-atom energy functions commonly used in biomolecular simulations retain only *partial* nonbonded interactions between terminal atoms involved in a torsion. The atoms involved in a torsion, if numbered beginning with 1, would be 1, 2, 3, and 4, so the terminal atoms could be called atoms 1 and 4, and nonbonded interactions between such atoms are called "1-4 interactions". These interactions are often present but reduced, though the exact amount of reduction differs by the energy function or force field family. For example, the AMBER family force fields usually reduce 1-4 electrostatics to $\frac{1}{1.2}$ of their original value, and 1-4 Lennard-Jones interactions to $\frac{1}{2}$ of their original value. 1-4 interactions are essentially considered the borderline between the bonded and non-bonded regions.

3.5.2 Books

For a discussion of molecular interactions, we recommend "Intermolecular and surface forces" by Jacob N. Israelachvili. A variety of other books discuss these from a simulation perspective, e.g. Leach [**?**] and Allen and Tildesley [**?**].

4 Basic simulation concepts and terminology

Above, we covered a variety of fundamental concepts needed for understanding molecular simulations and the types of interactions and forces we seek to model; here, we shift our attention to understanding basics of how molecular simulations actually work.

4.1 Force fields

The term "force field" simply refers to the included terms, particular form, and specific implementation details, including parameter values, of the chosen potential energy function.[1]

Most of the terms included in potential energy functions have already been detailed in Section 3.5, with the most com-

[1] It is worth noting there is a occasionally a bit of ambiguity when the term "force field" is used. In some cases it is used to refer to a library of parameters that could be applied to assign an energy function to a specific molecular system via a parameterization process after applying some specific chemical perception like atom typing to that system [**?**]. For example, one might speak of the AMBER ff15FB [**?**] protein force field, which essentially provides a recipe for assigning parameters to a protein once atom types are assigned. In other cases, "force field" is used to refer to the specifics of the potential energy function after application to a specific system — what could also be called a "parameterized system". For our purposes here, the distinction between a force field library and a parameterized system is not particularly important, but it is worth noting the potential ambiguity.

mon being Coulombic, Lennard-Jones, bond, angle, and torsional (dihedral) terms. Here, we very briefly describe the mathematical forms used to represent such interactions.

Non-bonded interactions of the Lennard-Jones form are well-described throughout the literature (for instance see Ch. 4 of **?**]); these model a short-range repulsion that scales as $1/r^{12}$ and a long-range attraction that scales as $1/r^6$ Coulombic interactions, including both short and long-range components, are described in detail elsewhere in this document. To represent bonded interactions, harmonic potentials are often employed. The same is true for angles between three bonded atoms, but the harmonic potential is applied with respect to the angle formed and not the distance between atoms. Torsional terms are also commonly employed, usually consisting as sums of cosines, i.e. a cosine expansion.

While the above are perhaps the most common potentials used, there are a variety of common variations as well. More exotic potentials based on three-body intermolecular orientations, or terms directly coupling bond lengths and bending angles are also possible. Some historic force fields also added an explicit (non-Coulombic) hydrogen bonding term, though these are less frequently used in many cases today. Additionally, other choices of potential function are of course acceptable, including Buckingham or Morse potentials, or the use of "improper" dihedral terms to enforce planarity of cyclic portions of molecules. This may even include empirical corrections based on discrete binning along a particular set of degrees of freedom [**? ?**], as well as applied external fields (i.e. electric fields) and force field terms describing the effect of degrees of freedom, such as solvent, that have been removed from the system via "coarse-graining." [**?**] For a more in-depth discussion of common (as well as less common) force field terms, see Ch. 4 of **?**], or for an in-depth review of those specific to simulating biomolecules, see **?**].

Functional forms used to describe specific terms in a potential energy function may be vastly different in mathematical character even though they seek to describe the same physics. For instance, the Lennard-Jones potential implements an r^{-12} term to represent repulsions, while an exponential form is used in the Buckingham potential. This results in very different mathematical behavior at very short distances and as a result differences in numerical implementation as well as evaluation efficiencies via a computer. For this reason, one functional form may be preferred above another due to enhanced numerical stability or simplicity of implementation, even though it is not as faithful to the underlying physics. In this regard, force field selection is a form of selecting a model – one should carefully weigh the virtues of accuracy and convenience or speed, and be ever-conscious of the limitations introduced by this decision (for instance, see **?**]). It is also important to know that most MD simulation engines only support a subset of functional forms. For those forms that are supported, the user manuals of these software packages are often excellent resources for learning more about the rationale and limitations of different potential energy functions and terms (e.g. see Part II of Amber reference manuals[**?**] and Ch. 4 of the reference manual for GROMACS [**?**]).

For practical purposes, most beginning users will not be fitting a force field or choosing a functional form, but will instead be using an existing force field that already relies on a particular functional form and is available in their simulation package of choice, so for such users it is more important to know how the functional form represents the different interactions involved than to necessarily be able to justify why that particular functional form was chosen.

Many examples of force fields abound in the literature — in fact, too many to provide even a representative sample or list of citations, as most force fields are specifically developed for particular systems or categories of systems under study. However, reviews are available describing and comparing force fields for biomolecular simulations [**? ?**], solid, covalently-bonded materials [**?**], polarizable potentials [**?**], and models of water [**? ?**], to name just a few. Many force fields are open-source and parameter file libraries may be found through the citations in the resources above or are often distributed with molecular simulation packages. Limited databases of force fields also exist, most notably for simulations of solid materials where interatomic potentials display a much wider array of mathematical forms [**? ?**].

Specification of a force field involves not just a choice of functional form, but the details of the specific parameters for all of the interacting particles which will be considered — that is, the specific parameters governing the interactions as specified by the functional form. Parameters are usually specific to certain types of atoms, bonds, molecules, etc., and include point charges on atoms if electrostatic terms are in use.

Some choices which are often considered auxiliary actually comprise part of the choice of the force field or interaction model. Specifically, settings such as the use of constraints, the treatment of cut-offs and other simulation settings affect the final energies and forces which are applied to the system. Thus, to replicate a particular force field as described previously, such settings should be matched to prior work such as the work which parameterized the force field. The choice of how to apply a cutoff, such as through direct truncation, shifting of the potential energy function, or through the use of switching functions, should be maintained if identical matches to prior work computing the properties of interest are desired. This is especially important for the purposes of free energy calculations, where the potential energy itself is recorded. However, force fields are in some cases slow to

adapt to changes in protocol, so current best practices seem to suggest that lattice-sum electrostatics should be used for Coulomb electrostatics in condensed phase systems, even if the chosen force field was fitted with cutoff electrostatics, and in many cases long-range dispersion corrections should be applied to the energy and pressure to account for truncated Lennard-Jones interactions [**?**].

DLM: Probably need more cites here.

For almost all force fields, many versions, variants, and modifications exist, so if you are using a literature force field or one distributed with your simulation package of choice, it is important to pay particular attention (and make note of) exactly what version you are using and how you obtained it so you will be able to accurately detail this in any subsequent publications.

As clearly described in **?**], it is of paramount importance to understand the capabilities and limitations of various force field models that may seem appropriate for one's work. Depending on the physics being simulated and the computational resources at hand, no force field in the literature may provide results that accurately reproduce experiment. But with so many force fields to pick from, how is this possible? The issue lies in what is termed "transferability." Simply put, a classical description of dynamics, as implemented in MD, cannot universally describe all of chemistry and physics. At some level of finer detail, all of the potential functions described above are simply approximations. Due to this, force field developers must often make the difficult decision of sacrificing accuracy or generality. For instance, a force field may have been developed to very accurately describe a single state point, in which case it is obvious that extensive testing should be performed to ensure that it is also applicable at other conditions. Even with force fields developed to be general and transferable, it is essential to ensure that the desired level of realism is achieved, especially if applying such a model to a new system (even more caution is advised when mixing force fields!). Either way, it is always a good idea to check results against previous literature when possible. This helps ensure that the force field is being implemented properly and, though it may seem laborious on a short-time horizon, can pay substantial dividends in the long-run.

Because this balance of accuracy versus generality and transferability can be challenging, some efforts eschew transferability entirely and instead build "bespoke" force fields, where each molecule is considered as a unique entity and assigned parameters independently of any other molecule or representation of chemical space (e.g. [**?**]). Such approaches offer the opportunity to assign all molecules with parameters assigned in a consistent way; however, they are unsuitable for applications where speed needs to exceed that of the parameter assignment process – so, for example, for docking of a large library of potential ligands to a target receptor, if compounds must be screened at seconds or less per molecule, such approaches may not be suitable.

4.2 Periodic boundary conditions

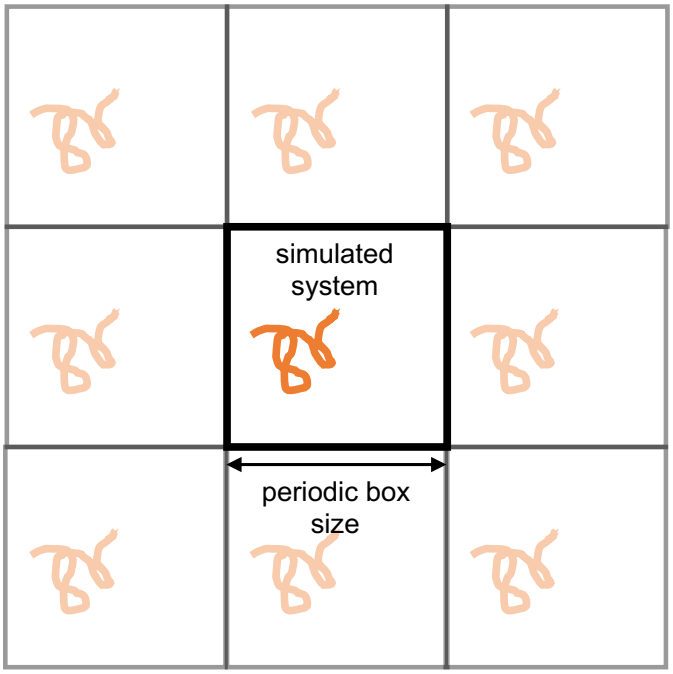

Figure 2. Periodic boundary conditions are shown for a simple 2D system. Note that the simulated system is a sub-ensemble within an infinitely sized system of identical, small ensembles.

Periodic boundary conditions allow more accurate estimation of bulk properties from simulations of finite, essentially nanoscale systems. More precisely, simulations of comparatively small systems with periodic boundary conditions can be a good approximation to the behavior of a small subsystem in a larger bulk phase (or at least are a much better approximation than simply simulating a nanodroplet or a finite system surrounded by vacuum). Periodic boundary conditions can alleviate many of the issues with finite size effects because each particle interacts with periodic images of particles in the same system. Clearly, though, it is undesirable for a single particle to interact with the same particle multiple times. To prevent this, a cut-off of many non-bonded interactions should be chosen that is less than half the length of the simulation box in any dimension. (However, as noted in Section 3.4 these cut-offs are not normally applied to electrostatic interactions because truncating these interactions induces worse artifacts than does including interactions with multiple copies of the same particle. Instead, what are often termed "cut-offs" that are applied to electrostatics are instead a shift from short-range to long-range treatments.) Such cut-offs impose

a natural lower limit to the size of a periodic simulation box, as the box must be large enough to capture all of the most significant non-bonded interactions. Further information on periodic boundary conditions and discussion of appropriate cut-offs may be found in **?**], sections 6.5 and 6.7 and **?**], lecture on Simulations of bulk phases.

It is very important to note that periodic boundary conditions are simply an approximation to bulk behavior. They DO NOT effectively simulate an infinitely sized simulation box, though they do reduce many otherwise egregious finite-size effects. This is most easily seen by imagining the placement of a solute in a periodic simulation box. The solute will be replicated in all of the surrounding periodic images. The concentration of solute is thus exactly one per the volume of the box. Although proper selection of non-bonded cutoffs will guarantee that these solutes do not directly interact (hence the common claim that such systems are at infinite dilution), they may indirectly interact through their perturbation of nearby solvent. If the solvent does not reach a bulk-like state between solutes, the simulation will still suffer from obvious finite-size effects.

Macroscopic, lab-scale systems, or bulk systems, typically consist of multiple moles of atoms/molecules and thus from a simulation perspective are effectively infinite systems. We attempt to simulate these by simulating finite and fairly small systems, and, in a sense, the very idea that the simulation cell is not infinite, but simply periodic, immediately gives rise to finite-size effects. Thus, our typical goal is not to remove these completely but to reduce these to levels that do not adversely impact the results of our simulations. Finite-size effects are particularly apparent in the electrostatic components of simulations, as these forces are inherently longer ranged than dispersion forces, as discussed in Section 3.4. One should always check that unexpected long-range correlations (i.e. on the length-scale of the simulation box) do not exist in molecular structure, spatial position, or orientation. It should also be recognized that periodic boundary conditions innately change the definition of the system and the properties calculated from it. Many derivations, especially those involving transport properties, such as diffusivity [**?**], assume infinite and not periodic boundary conditions. The resulting differences in seemingly well-known expressions for computing properties of interest are often subtle, yet may have a large impact on results. Such considerations should be kept in mind when comparing results between simulations and with experiment.

4.3 Main steps of a molecular dynamics simulation

While every system studied will present unique challenges and considerations, the process of performing a molecular dynamics simulation generally follows these steps:

1. System preparation
2. Minimization/Relaxation
3. Equilibration
4. Production

Additional explanations of these steps along with procedural details specific to a given simulation package and application may be found in a variety of tutorials [**? ?**]. It should be noted that these steps may be difficult to unambiguously differentiate and define in some cases. Additionally, it is assumed that prior to performing any of these steps, an appropriate amount of deliberation has been devoted to clearly defining the system and determining the appropriate simulation techniques.

4.3.1 System preparation

System preparation focuses on preparing the starting state of the desired system for input to an appropriate simulation package, including building a starting structure, solvating (if necessary), applying a force field, etc. Because this step differs so much depending on the composition of the system and what information is available about the starting structure, it is a step which varies a great deal depending on the nature of the system at hand and as a result may require unique tools.

Given the variable nature of system preparation, it is highly recommended that best practices documents specific to this issue and to the type of system of interest be consulted. If such documents do not exist, considerable care should be exercised to determine best practices from the literature.

Loosely speaking, system preparation can be thought of as consisting of two *logical* components which are not necessarily consecutive or separate. One comprises building the configuration of the system in the desired chemical state and the other applying force field parameters.

For building systems, freely available tools for constructing systems are available and can be a reasonable option (though their mention here should not be taken as an endorsement that they necessarily encapsulate best practices). Examples include tools for constructing specific crystal structures, proteins, and lipid membranes, such as Moltemplate, Packmol, and Atomsk.

A key consideration when building a system is that the starting structure ideally ought to resemble the equilibrium structure of the system at the thermodynamic state point of interest. For instance, highly energetically unfavorable

configurations of the system, such as blatant atomic overlaps, should be avoided. In some sense, having a good starting structure is only a convenience to reduce equilibration times (if the force field is adequate); however, for some systems, equilibration times might otherwise be prohibitively long.

System preparation is arguably the most critical stage of a simulation and in many cases receives the least attention. Specifically, if your system preparation is flawed, such flaws may prove fatal. Potentially the worst possible outcome is if the prepared system is not what you intended (e.g. it contains incorrect molecules or protonation states) but is chemically valid and well described by your force field and thus proceeds without error through the remaining steps — and in fact this is a frequent outcome of problems in system preparation. It should not be assumed that a system has been prepared correctly if it is well-behaved in subsequent equilibration steps; considerable care should be taken here.

Assignment or development of force field parameters is also critical, but is outside the scope of this work. For our purposes, we will assume you have already obtained or developed force field parameters suitable for your system of interest.

4.3.2 Minimization

The purpose of minimization, or relaxation, is to find a local energy minimum of the starting structure so that the molecular dynamics simulation does not immediately "blow up" (i.e. the forces on any one atom are not so large that the atoms move an unreasonable distance in a single timestep). This involves standard minimization algorithms such as steepest descent. For a more involved discussion of minimization algorithms utilized in molecular simulation, see **?**], sections 5.1-5.7.

4.3.3 Assignment of velocities

Minimization ideally takes us to a state from which we can begin numerical integration of the equations of motion without overly large displacements (see **?**], section 7.3.4); however, to begin a simulation, we need not just positions but also velocities. Minimization, however, provides only a final set of positions. Thus, starting velocities must be assigned; usually this is done by assigning random initial velocities to atoms in a way such that the correct Maxwell-Boltzmann distribution at the desired temperature is achieved as a starting point.

In some cases, we seek to obtain multiple separate and independent simulations of different instances or realizations of a particular system to assess error, collect better statistics, or help gauge dependence of results on the starting structure. It is worth noting that even very small differences in initial configuration, such as even a difference in position of a single atom, lead to exponential divergence of the time evolution of the system [**?**], meaning that simply running different simulations starting with different initial velocities will lead to dramatically different time evolution over long enough times. Still, an even better way to generate independent realizations is to begin with different starting configurations, such as different conformations of the molecule(s) being simulated, as this leads to behavior which is immediately different.

4.3.4 Equilibration

Ultimately, we usually seek to run a simulation in a particular thermodynamic ensemble (e.g. the NVE or NVT ensemble) at a particular state point (e.g. target energy, temperature, and pressure) and collect data for analysis which is appropriate for those conditions and not biased depending on our starting conditions/configuration. This means that usually we need to invest simulation time in bringing the system to the appropriate state point as well as relaxing away from any artificially induced metastable starting states. In other words, we are usually interested in sampling the most relevant (or most probable) configurations in the equilibrium ensemble of interest. However, if we start in a less-stable configuration a large part of our equilibration may be the relaxation time (this may be very long for biomolecules or systems at phase equilibrium) necessary to reach the more relevant configuration space.

The most straightforward portion of equilibrium is bringing the system to the target state point. Usually, even though velocities are assigned according to the correct distribution, a thermostat will still need to add or remove heat from the system as it approaches the correct partitioning of kinetic and potential energies. For this reason, it is advised that a thermostatted simulation is performed prior to a desired production simulation, even if the production simulation will ultimately be done in the NVE ensemble. This phase of equilibration can be monitored by assessing the temperature and pressure of the system, as well as the kinetic and potential energy, to ensure these reach a steady state on average. For example, an NPT simulation is said to have equilibrated to a specific volume when the dimensions of the simulation box fluctuate around constant values with minimal drift. This definition, though not perfectly rigorous, is usually suitable for assessing the equilibration of energies, temperature, pressure, and box dimensions during equilibration simulations.

A more difficult portion of equilibration is to ensure that other properties of the system which are likely to be important are also no longer changing systematically with simulation time. At equilibrium, a system may still undergo slow fluctuations with time, especially if it has slow internal degrees of freedom – but key properties should no longer show systematic trends away from their starting structure. Thus, for example, for biomolecular simulations it is common to

examine the root mean squared deviation (RMSD) of the molecules involved as a function of time, and potentially other properties like the number of hydrogen bonds between the biomolecules present and water, as these may be slower to equilibrate than system-wide properties like the temperature and pressure.

Once the kinetic and potential energies fluctuate around constant values and other key properties are no longer changing with time, the equilibration period has reached its end.

Depending on the target ensemble for production, the procedure for the end of equilibration is somewhat different. If an NVE simulation is desired, the thermostat may be removed and a snapshot selected that is simultaneously as close to the average kinetic and potential energies as possible. This snapshot, containing both positions and velocities may be used to then start an NVE simulation that will correspond to a temperature close to that which is desired. This is necessary due to the fact that only the average temperature is obtained through coupling to a thermostat (see Section 4.4), and the temperature fluctuates with the kinetic energy at each timestep.

If the target is a simulation in the NVT ensemble at a particular density, equilibration should be done in the NPT ensemble. In this case, the system may be scaled to the desired average volume before starting a production simulation (and if rescaling is done, additional equilibration might be needed).

The schematic below (3) demonstrates what is generally an appropriate equilibration work-flow for common production ensembles. Clearly, this schematic cannot cover every case of interest, but should provide some idea of the general approach. For more information on equilibration procedures, see **?**], section 7.4 and **?**], lectures on Molecular dynamics and Computing properties.

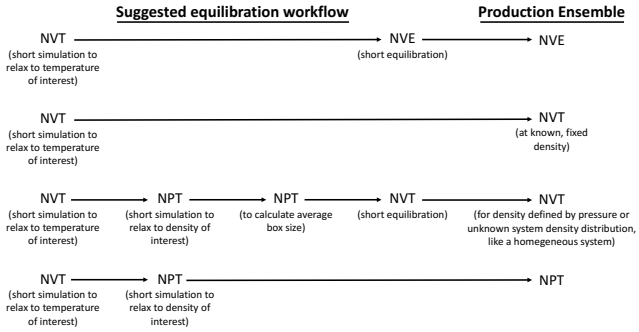

Figure 3. Common equilibration work-flows are shown; these vary depending on the target ensemble for production simulations (right). Typically, an initial phase of equilibration at constant volume and temperature is needed to bring the system to the desired target temperature or energy. For stability reasons, this initial phase is usually needed even if the goal is to also bring the system to a target pressure. If the production ensemble is an NVE ensemble, an initial NVT simulation is usually followed by a short additional NVE equilibration before collection of production data. If the production ensemble is NVT, protocols may differ depending on whether it is necessary to allow the system to equilibrate to a particular density/volume or whether the volume is selected *a priori* (second and third rows). And if production is to be NPT, it is usually equilibrated first at NVT before equilibrating to the target pressure (final row).

4.3.5 Production

Once equilibration is complete, we may begin collecting data for analysis. Typically this phase is called “production”. The main difference between equilibration and production is simply that in the production simulation, we plan to retain and analyze the collected data. Production must always be preceded by equilibration appropriate for the target production ensemble, and production data should never be collected immediately after a change in conditions (such as rescaling a box size, energy minimizing, or suddenly changing the temperature or pressure) except in very specific applications where this is the goal.

For bookkeeping purposes, sometimes practitioners choose to discard some initial production data as additional equilibration; usually this is simply to allow additional equilibration time after a change in protocol (such as a switch from NVT to NPT), and the usual considerations for equilibration apply in such cases (see **?**], lecture on Computing Properties).

Analysis of production is largely outside the scope of this work, but requires considerable care in computing observables and assessing the uncertainty in any computed properties. Usually, analysis involves computing expectation values of particular observables, and a key consideration is to obtain *converged* estimates of these properties — that is, estimates that are based on adequate simulation data so that they no longer depend substantially on the length of the simulation which was run or on its initial conditions. This is closely related to the above discussion of equilibration. Depending on the relaxation timescales involved, one may only realize after analysis of a “production” trajectory that the system was still equilibrating in some sense.

A separate Best Practices document addresses these critical issues of convergence and error analysis (https://github.com/dmzuckerman/Sampling-Uncertainty). For more specific details on procedures and parameters used in production simulations, see the appropriate best practices document for the system of interest.

DLM: Probably need to add something here about how often to store data.

4.4 Thermostats

Here, we discuss why thermostats, which seek to control the temperature of a simulation, are (often) needed for molecular simulations. We review background information about thermostats and how they work, introduce some popular thermostats, and highlight common issues to understand and avoid when using thermostats in MD simulations.

4.4.1 Thermostats seek to maintain a target temperature

As mentioned above, molecular dynamics simulations are used to observe and glean properties of interest from some system of study. In many cases, to emulate experiments done in laboratory conditions (exposed to the surroundings), sampling from the canonical (constant temperature) ensemble is desired [**?**]. Generally, if the temperature of the system must be maintained during the simulation, some thermostat algorithm will be employed.

4.4.2 Background and How They Work

The temperature of a molecular dynamics simulation is typically measured using kinetic energies as defined using the equipartition theorem: $\frac{3}{2}Nk_{\mathrm{B}}T = \left\langle \sum_{i=1}^{N} \frac{1}{2} m_i v_i^2 \right\rangle$. The angled brackets indicate that the temperature is defined as a time-averaged quantity. If we use the equipartition theorem to calculate the temperature for a single snapshot in time of a molecular dynamics simulation instead of time-averaging, this quantity is referred to as the instantaneous temperature. The instantaneous temperature will not always be equal to the target temperature; in fact, in the canonical ensemble, the instantaneous temperature should undergo fluctuations around the target temperature.

Thermostat algorithms work by altering the Newtonian equations of motion that are inherently microcanonical (constant energy). Thus, it is preferable that a thermostat not be used if it is desired to calculate dynamical properties such as diffusion coefficients; instead, the thermostat should be turned off after equilibrating the system to the desired temperature. However, while all thermostats give non-physical dynamics, some have been found to have little effect on the calculation of particular dynamical properties, and they are commonly used during the production simulation as well[**?**].

There are several ways to categorize the many thermostatting algorithms that have been developed. For example, thermostats can be either deterministic or stochastic depending on whether they use random numbers to guide the dynamics, and they can be either global or local depending on whether they are coupled to the dynamics of the full system or of a small subset. Many of the global thermostats can be made into local "massive" variants by coupling separate thermostats to each particle in the system rather than having a single thermostat for the whole system. There are also several methods employed by thermostat algorithms to control the temperature. Some thermostats operate by rescaling velocities outside of the molecular dynamics' equations of motion, e.g., velocity rescaling is conducted after particles' positions and momenta have been updated by the integrator. Others include stochastic collisions between the system and an implicit bath of particles, or they explicitly include additional degrees of freedom in the equations of motion that have the effect of an external heat bath.

4.4.3 Popular Thermostats

Within this section, various thermostats will be briefly explored, with a small description of their uses and possible issues that are associated with each. This is by no means an exhaustive study of available thermostats, but is instead a survey of just some of the more popular and historic thermostats used in MD.

1. **Gaussian**

   The goal of the Gaussian thermostat is to ensure that the instantaneous temperature is exactly equal to the target temperature. This is accomplished by modifying the force calculation with the form $F = F_{interaction} + F_{constraint}$, where $F_{interaction}$ is the standard interactions calculated during the course of the simulation and $F_{constraint}$ is a Lagrange multiplier that keeps the kinetic energy constant. The reasoning for the naming of this thermostat is due to its use of the Gaussian principle of least constraint to determine the smallest perturbative forces needed to maintain the instantaneous temperature [**?**]. Clearly, this thermostat does not sample the canonical distribution; it instead samples the isokinetic (constant kinetic energy) ensemble. However, the isokinetic ensemble samples the same configurational phase space as the canonical ensemble, so position-dependent (structural) equilibrium properties can be obtained equivalently with either ensemble [**?**]. However, velocity-dependent (dynamical) properties will not be equivalent between the ensembles. This thermostat is generally only used in certain advanced applications [**?**].

2. **Simple Velocity Rescaling**

   The simple velocity rescaling thermostat is one of the easiest thermostats to implement; however, this thermostat is also one of the most non-physical thermostats. This thermostat relies on rescaling the momenta of the particles such that the simulation's instantaneous temperature exactly matches the target temperature [**?**]. Similarly to the Gaussian thermosat, simple velocity rescaling aims to sample the isokinetic ensemble rather than the canonical ensemble. However, it has been

shown that the sample velocity rescaling fails to properly sample the isokinetic ensemble except in the limit of extremely small timesteps[**?**]. Its usage can lead to simulation artifacts, so it is not recommended[**? ?**].

3. **Berendsen**

   The Berendsen[**?**] thermostat (also known as the weak coupling thermostat) is similar to the simple velocity rescaling thermostat, but instead of rescaling velocities completely and abruptly to the target kinetic energy, it includes a relaxation term to allow the system to more slowly approach the target. Although the Berendsen thermostat allows for temperature fluctuations, it samples neither the canonical distribution nor the isokinetic distribution. Its usage can lead to simulation artifacts, so it is not recommended[**? ?**].

4. **Bussi-Donadio-Parrinello (Canonical Sampling through Velocity Rescaling)**

   The Bussi[**?**] thermostat is similar to the simple velocity rescaling and Berendsen thermostats, but instead of rescaling to a single kinetic energy that corresponds to the target temperature, the rescaling is done to a kinetic energy that is stochastically chosen from the kinetic energy distribution dictated by the canonical ensemble. Thus, this thermostat properly samples the canonical ensemble. Similarly to the Berendsen thermostat, a user-specified time coupling parameter can be chosen to vary how abruptly the velocity rescaling takes place The choice of time coupling constant does not affect structural properties, and most dynamical properties are fairly independent of the coupling constant within a broad range[**?**].

5. **Andersen**

   The Andersen[**?**] thermostat works by selecting particles at random and having them "collide" with a heat bath by giving the particle a new velocity sampled from the Maxwell-Boltzmann distribution. The number of particles affected, the time between "collisions", and how often it is applied to the system are possible variations of this thermostat. The Andersen thermostat does reproduce the canonical ensemble. However, it should only be used to sample structural properties, as dynamical properties can be greatly affected by the abrupt collisions.

6. **Langevin**

   The Langevin[**?**] thermostat supplements the microcanonical equations of motion with Brownian dynamics, thus including the viscosity and random collision effects of an implicit solvent. It uses a general equation of the form $F = F_{interaction} + F_{friction} + F_{random}$, where $F_{interaction}$ is the standard interactions calculated during the course of the simulation, $F_{friction}$ is the damping used to tune the "viscosity" of the implicit bath, and F_{random} effectively gives random collisions with solvent molecules. Careful consideration must be taken when choosing the friction damping parameter; in the limit of a zero damping parameter, the dynamics are microcanonical, and in the limit of an infinite damping parameter, the dynamics are purely Brownian.

7. **Nosé-Hoover**

   The Nosé-Hoover thermostat [**?**] abstracts away the thermal bath from the previous thermostats and condenses it into a singular additional degree of freedom. This fictitious degree of freedom has a "mass" that can be changed to interact with the particles in the system in a predictable and reproducible way while maintaining the canonical ensemble. The choice of "mass" of the fictitious particle (which in many simulation packages is instead expressed as a time damping parameter) can be important as it affects the fluctuations that will be observed. For many reasonable choices of the mass, dynamics are well-preserved [**?**]. This is one of the most widely implemented and used thermostats. However, it should be noted that with small systems, ergodicity can be an issue[**? ?**]. This can become important even in systems with larger numbers of particles if a portion of the system does not interact strongly with the remainder of the system, such as in alchemical free energy calculations when a solute or ligand is non-interacting. Martyna et al. [**?**] discovered that by chaining thermostats, ergodicity can be enhanced, and most implementations of this thermostat use Nosé-Hoover chains.

4.4.4 Summary

To summarize, observe (Table 1), as a general summary and guide for exploring the usage of various thermostats. Knowing the system you are simulating and the benefits and weaknesses to each thermostat is crucial to successfully and efficiently collect meaningful, physical data. If you are only interested in sampling structural properties such as radial distribution functions, many of the given thermostats can be used, including the Gaussian, Bussi, Andersen, Langevin, and Nosé-Hoover thermostats. If dynamical properties will be sampled, it is preferable to turn off the thermostat before beginning production cycles, but the Bussi and Nosé-Hoover thermostats (and in cases with implicit solvent, the Langevin thermostat), can often be used without overly affecting the calculation of dynamical properties. Since dynamical properties are not important during equilibration, faster algorithms like the Andersen or Bussi thermostats can be used, with a switch to the Nosé-Hoover thermostat for production. Overall, the Bussi thermostat has been shown to work well for most purposes, and its use is recommended as a general-purpose

thermostat.

4.5 Barostats

Here, we discuss why barostats are used, give their background, discuss roughly how they work, describe some popular options, and summarize with some recommendations.

4.5.1 Motivation

Typically, thermodynamic properties of interest are measured under open air conditions in a laboratory, which (for short timescales) means at they are measured at essentially constant temperature and pressure. Such conditions correspond to what is called the isothermal-isobaric ensemble, probably one of the most popular ensembles for MD simulationsl. As is the case with thermostats, if the pressure must be maintained in a simulation, a barostat algorithm will be needed to sample this ensemble.

4.5.2 Background and How They Work

In many experiments, the container is either open to the atmosphere, meaning that it is subject to a roughly constant pressure of approximately one atmosphere. To obtain a different pressure, some device, like a piston, inert gas, etc., would be needed to control the pressure and volume of the system [**? ?**].

For the purpose of molecular modeling, consider a hypothetical system that is being compressed and/or expanded by a fictitious piston that has some mass which acts in all directions uniformly. Since the piston is acting on the system from all directions, it can be considered as applying a uniform compression or expansion. The mass of the piston can be tuned to change the compression of the system, which will change how often the particles in the system will interact with the system enclosure. These impacts from the particles on the "enclosure" will impart a stress on the system box which can be related to the stress the surroundings are imparting on the system. With this relationship, we can use the virial theorem (an expectation value relating to positions and forces) to calculate the pressure of a system [**? ?**]. However, this is much more challenging when considering pairwise interactions and periodic boundary conditions [**? ? ?**], and a different approach is often utilized. Our main point here, however, is that pressure can be related to instantaneous properties of the system allowing us to calculate an instantaneous pressure in a similar manner to how we calculate an instantaneous temperature for thermostats.

Thus, barostat algorithms apply to keep the instantaneous pressure of a system at or near the target pressure.

Barostat algorithms control pressure alone, not temperature, so if the target ensemble is isothermal-isobaric, they must also be applied with a thermostat. If a barostat is applied without a thermostat, only the number of particles (N), the pressure (P), and the enthalpy (H) of the system is held constant. This is known as the isoenthalpic-isobaric ensemble (NPH). To sample from the isothermal-isobaric ensemble (NPT), a thermostating algorithm like the ones discussed earlier must also be applied.

Many barostats are available, but can usually can be classified into three main categories: volume rescaling, weakly coupled, and extended ensemble barostats [**? ?**]. The next section will describe the main differences between these barostats, and give some recommendations for proper use.

4.5.3 Popular and Notable Barostats

Here, we introduce a few notable barostats and give a high-level summary of each, noting some key issues. This is not an exhaustive list of barostats and barostat algorithms, just a sampling of popular and historic ones used in MD.

Volume Rescaling

1. **Volume Rescaling**

   Volume rescaling barostats are the simplest example of pressure control in molecular simulations. Every time this barostat is executed, the volume of the system is modified to produce the exact pressure desired. This does **not** sample the proper ensemble and thus cannot be used for production sampling [**?**]. This also does not smoothly approach the target pressure either, which might cause very unphysical issues with the system during integration.

Weakly Coupled Barostats

1. **Berendsen**

   The Berendesen [**?**] barostat is very similar to the Berendsen thermostat discussed earlier. It seeks to improve upon the volume rescaling methods mentioned above. This was to be achieved by coupling the system to a weakly interacting pressure bath [**?**]. This bath scales the volume periodically by a scaling factor, which produces more realisitc fluctuations in the pressure as it slowly approaches the target pressure. In contrast to volume rescaling, Berendsen will approach the target pressure more realistically, but the ensemble it is sampling from is not well defined and cannot be guaranteed to be NPT or NPH. Berendsen can be useful for the beginning stages of equilibration, but should **not** be used for production sampling.

Extended System Barostats

1. **Andersen Barostat**

Table 1. Basic summary of popular thermostats. ✗ indicates that the thermostat does not fulfill the statement, ✓ indicates that the thermostat does fulfill the statement, and (✓) indicates that the thermostat fulfills the statement under certain circumstances.

Thermostat	Ensemble	Deterministic/ Stochastic	Global/ Local	Physical?	Correct Structural Properties?	Correct Dynamical Properties?
None	Microcanonical	Deterministic		✓	✓	✓
Gaussian	Isokinetic	Deterministic	Global	✗	✓	✗
Simple Velocity Rescaling	Undefined	Deterministic	Global	✗	✗	✗
Berendsen	Undefined	Deterministic	Global	✗	✗	✗
Bussi	Canonical	Stochastic	Global	✗	✓	(✓)
Andersen	Canonical	Stochastic	Local	✗	✓	✗
Langevin	Canonical	Stochastic	Local	✗	✓	(✓)
Nosé-Hoover	Canonical	Deterministic	Global	✗	✓	(✓)

First described by Andersen [**?**] in 1980, the system is coupled to a fictitious pressure bath, by adding an additional degree of freedom to the equations of motion. This behaves as if the system is being acted upon by an isotropic piston. This is similar to the Nosé-Hoover thermostat, which is also an extended system algorithm. This barostat does sample the correct ensemble. However, it is isotropic in nature and applying anisotropic pressures to parts of the system is not possible.

2. **Parrinello-Rahman Barostat**

   The Parrinello-Rahman [**?**] barostat is an extension to the Andersen barostat. Unlike the Andersen barostat, Parrinello-Rahman supports the anisotropic scaling of the size and shape of the simulation box [**?**]. This can be quite useful in solid simulations, where phase changes can be shape changes in a crystal lattice, compared to a liquid or gas, which has no well defined shape. This barostat has essentially the same properties as the Andersen one, with the additional support anisotropy.

3. **Martyna-Tuckerman-Tobias-Klein (MTTK) Barostat**

   The MTTK barostat has substantial similarity to the Parrinello-Rahman and Andersen barostats. When Parrinello Rahman's equations of motion were discovered to only hold true in the limit of large systems, the MTTK barostat introduced alternate equations of motion to correctly sample the ensemble for smaller systems as well [**? ?**]. Thus, MTTK [**? ?**] is usually seen as an improvement over Parrinello-Rahman [**?**] for such systems.

Monte Carlo Barostats

1. **MC Barostat**

   Constant pressure may also be achieved by periodically performing Monte Carlo moves that adjust the system volume. For an explanation of how such moves are accepted or rejected, see "Monte Carlo simulations in other ensembles" in **?**]. These MC barostats are computationally advantageous in that the Virial need not be computed, and they may be easily extended to accommodate anisotropic systems. However, they rigorously only explore the correct distribution of volumes in the NPT ensemble without any attempt at preserving the dynamic fluctuations involved in this sampling. Unlike for extended system barostats, there is no sense of relaxation time over which the volume of the system responds. Instead, the rate at which the volume may respond is limited by the frequency with which MC moves are performed and the maximum allowed change in volume. Thus, long-time dynamics are not accurately reproduced in any sense for MC barostats.

4.5.4 Summary

In summary, there are three types of barostats usually implemented in molecular dynamics codes which can greatly affect the data you are collecting from the system. Volume rescaling is not recommended for collection of production data. This barostat does not sample from any correct ensemble, nor does it utilize any "realistic" approach to achieve the target pressure. Weak coupling barostats provide some improvement compared to volume rescaling methods. However, these methods cannot be used to bring the system to equilibrium effectively. They can be used for approaching the target pressure in a more realistic fashion compared to the volume rescaling barostat, which itself is primarily useful only as a very stable thermostat for very early simulation stages if other algorithms have trouble beginning from particularly strained starting structures. (Alternatively, such issues can be avoided by running NVT equilibration before using a barostat, Figure 3.) Finally, extended ensemble barostats are suitable for the production runs of most systems. It is usually not recommended to use these for the equilibration process, as

these barostats do not behave as well when not near the target pressure. These can be affected by the starting configuration and pressure values much more than the Berendsen or volume rescaling barostats. MTTK and Parinello-Rahman allow for more flexibility in terms of the shape modulation of the simulation box. Ultimately, however, one's choice often is limited by which extended-ensemble barostat has been implemented in your simulation engine of choice. It is recommended to begin with a volume rescaling or weakly coupled barostat to quickly bring the system to the target pressure, then switch to an extended ensemble barostat for final equilibration and production.

4.6 Integrators

For systems consisting of more than three interacting bodies with no constrained degrees of freedom, there is no analytical solution to the equations of motion. Instead, we must approximate the dynamics in a discrete manner. This is usually termed numerical integration of the equations of motion. Algorithms to perform this integration take many forms and are usually called integrators. Here, we explain the need for integrators, discuss key criteria like energy conservation, and highlight a number of commonly used integrators.

4.6.1 Desireable integrator properties

So-called "good" integrators contain certain features that are appealing for molecule simulations. We start with the most obvious feature, which is that the integrator induces little error in the dynamics. Since integration is fundamentally about taking discrete steps to approximate continuous dynamics, this discretization process introduces errors (as can be observed by comparison to analytically soluble problems, like the harmonic oscillator). These errors are termed discretization errors, whereas additional errors called truncation errors are also accumulated through loss of precision during computer calculations. As will be discussed shortly, there are many strategies for avoiding discretization errors. For truncation errors, the only solution is to utilize a higher precision data type during calculations (i.e. use doubles instead of floats).

Integrators that minimize discretization error should preserve phase space volume and conserve energy. If phase space volume is not preserved, then the sampled ensemble at a later timestep will not be the same as that in which the system was initialized. This means that the collected data will not in fact reflect the ensemble of interest. Luckily, this issue may be avoided simply by guaranteeing that the integrator is reversible [**?**]. More details may be found in **?**], but basically if the mathematical operator representing the integrator preserves phase space volume, it also satisfies the definition of reversibility: if the operator is applied to propagate forward by Δt, the starting condition may be recovered by in turn applying the operator to the result using $-\Delta t$ as the timestep.

Energy conservation is also a desirable integrator property and is imperative in simulating the microcanonical (NVE) ensemble. This is a much trickier property to examine, and varies with different integrators. For instance, some classes of integrators better-preserve energy over short times, while others better-preserve energy at long times. The latter is generally preferred, though it may necessitate other sacrifices such as greater energy fluctuations away from the desired, exact system energy.

JIM: Should show citations for this section where people analyze energy conservation of different integrators and make comment directing people to this. Update: I'm having trouble finding good citations here. The work is either very theoretical and involved or very old. This is really true for this entire section, so help is appreciated.

When the energy does change over the course of a simulation, it is said to "drift." The most common reason for energy drift is due to a timestep that is overly long. If the timestep is much too long, the system can become unstable and blow up (energies become very large) due to overlap of atoms. More subtly, the timestep may be long enough that the system is still stable over long times, but too long for the chosen integrator to conserve energy. Other simulation parameters may also impact energy drift, such as the method of truncating forces and energies, as well as the choice of numerical precision. The latter effect, due to truncation errors, will become obvious if two simulations with different timesteps are compared. Shorter timesteps, and hence more steps to achieve a simulation of the same length, will result in *more* drift, since errors get larger with the number of calculations performed by the computer. This is exactly opposite to the behavior that is expected for poor energy conservation associated with discretization error, where a shorter timestep will reduce energy drift.

Overall, then, integrators do exhibit energy fluctuations that are timestep-dependent. All Verlet-equivalent integrators exhibit energy fluctuations which decrease with the square of the timestep [**?**], which is often an important check when assessing the correctness of an implementation. Thus, both energy drift and energy fluctuations are important criteria to understand when assessing integrators, and can be useful measures of simulation quality in the NVE ensemble.

Additionally, it is also desirable that an integrator be computationally efficient. Integrator cost mostly appears in the length of the timestep that may be taken while still avoiding discretization error. As discussed further below, the timestep must be at least an order of magnitude less than the smallest timescale of motion present in the system. However, de-

pending on the accuracy of the integrator with respect to reproducing the true dynamics, a smaller timestep might be necessary. If the integrator requires a very small timestep to avoid discretization error, then the computational cost greatly increases. Hence, a truly "good" integrator allows for long timesteps while still achieving low discretization error. This has the added benefit of also reducing truncation error, which is proportional to the number of timesteps taken. It is worth noting that the issue of integrator choice versus timestep is not always simple; in some cases, a "better" integrator might allow longer timesteps but also carry an additional computational cost that outweighs the benefits of an increased timestep.

4.6.2 Deterministic integrators

The most commonly used integrators are variants of the Verlet algorithm (e.g. Velocity Verlet or Leapfrog). Such integrators include terms for updating particle positions up to the order of the square of the timestep (i.e. they include forces). Inclusion of higher-order terms is favored in other families of algorithms, but generally leads to greater complexity and reduced computational efficiency at only marginal improvement in accuracy. Detailed discussion and derivation of many common integrators may be found in section 7.3 of **?**] and 4.3 of **?**]. Such integrators are not applicable, however, for simulations involving stochastic dynamics, as discussed below.

4.6.3 Stochastic integrators

Stochastic dynamics simulations include application of a random force to each particle, and represent discretizations of either Langevin or Brownian dynamics. A detailed description of such stochastic dynamics may be found in McQuarrie [**?**], Chapter 20. As detailed in section 4.4, it is common to apply temperature control through the use of Langevin dynamics. As a brief aside, this highlights the fact that the choice of integrator is often tightly coupled to the choice of thermostat and/or barostat. Different combinations may demonstrate better performance and for expanded ensemble methods it is absolutely necessary to utilize an integrator specific to the selected temperature- or pressure-control algorithm.

For simulating Langevin or other stochastic dynamics, the presence of random forces usually prevents the integrator from preserving phase-space volume. However, through fortuitous cancellation of error, some stochastic integration schemes may achieve preservation of *part* of the full phase-space (i.e. configurations *or* velocities are preserved) [**?**]. Though this may sound dire, in practice this is easily remedied through an appropriate choice of timestep - this just might need to be shorter or longer depending on the integration scheme. When using Langevin or Brownian dynamics, one should also be aware that calculations of any dynamic properties with longer timescales than the application of the random forces will be very different than those from deterministic trajectories. If one is only interested in configurational or thermodynamic properties of the system, this is of no consequence. If dynamics are of interest, the dependence of these properties on the integrator parameters (e.g. friction factor) should be assessed [**?**].

DLM: I need to review the paragraphing here; some of these are rather long and cover a lot.

4.6.4 How to choose an appropriate timestep?

The maximum timestep for a molecular dynamics simulation is dependent on the choice of integrator and the assumptions used in the integrator's derivation. For the commonly-used second order integrators such as the Verlet and Leapfrog algorithms, the velocities and accelerations should be approximately constant over the timestep. Thus, the timestep is limited by the highest frequency motion present in the system, which for all-atom simulations is usually bond vibrations. It is commonly found that using a timestep that is one tenth of this vibration's characteristic period is sufficient to conserve energy in the microcanonical ensemble. For example, if hydrogen molecules are present in the simulation box and the H-H bond vibration is the highest-frequency motion in the system, one can determine that with a force field harmonic force constant of approximately 500 N/m, the oscillation period will be 8 fs; thus, a 0.5 fs timestep can be used. For all-atom simulations with constraints on the high-frequency bonds, timesteps can be commonly increased to 2 fs; coarse-grained simulations with particles of higher mass and smaller force constants can have much larger timesteps. After choosing a timestep, a test simulation should be run in the microcanonical ensemble to ensure that the choice of timestep yields dynamics that conserve energy. Methods also exist to increase the timestep beyond the limit imposed by the system's highest-frequency motion. Many such enhanced timestepping algorithms exist such as multiple-timestep methods which separately integrate high-frequency and low-frequency motion and schemes which repartition atomic masses to decrease the highest-frequency motion seen in the system[**? ?**].

4.7 Long range electrostatics

DLM: I need to edit this section after we get Samarjeet's changes in; skipping for now.

- Cut-off is bad
- Need for special treatment
- Idea of an Ewald sum

- PPPM
- How to choose parameters

4.7.1 Motivation

Calculation of non-bonded interaction is generally the most time-consuming step of classical energy calculation. While the number of type of bonded interactions remain unchanged during the course of MD simulation, the ones for non-bonded interactions have to be updated frequently as molecules move into and out of the cutoff regions specified for extramolecular interactions. Furthermore, r^{-1} dependence of coulombic interaction in a 3D space complicates the calculation in the following two ways :

- Coulombic potential is conditionally convergent. What do we mean by this? Consider for example the alternating harmonic series :

  $$\sum_{k=1}^{\infty} \frac{-1^{k+1}}{k} = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \ldots = ln2$$

  But if we now change the order of the summation to, lets say :

  $$(1 - \frac{1}{2}) - \frac{1}{4} + (\frac{1}{3} - \frac{1}{6}) - \frac{1}{8} + (\frac{1}{5} - \frac{1}{10}) - \frac{1}{12} + (\frac{1}{7} - \frac{1}{14}) - \frac{1}{16} + \ldots$$

  $$= \frac{1}{2} - \frac{1}{4} + \frac{1}{6} - \frac{1}{8} + \frac{1}{10} - \ldots$$

  $$= \frac{1}{2}(1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} - \ldots) = \frac{1}{2}ln2$$

  Using a different order might give us a different result. Thus the order of summation matters !
- Coulombic potential is a slow-decaying, long range potential. This means that we need to account for non-bonded pairs over larger pairwise distances in order to minimize the errors. As enlisting atom-pairs is fundamentally an $O(n^2)$ operation, the time taken to compute non-bonded pairs increases quadratically with respect to an increase in the number of atoms.

As discussed earlier, simulations are generally performed under periodic boundary conditions, i.e. potential at any point is due to all the other charges in the system as well as their infinite images. Hence different methods have been developed to optimize the electrostatic potential. An obsolete method is to just use spherical truncation, i.e. only consider the interactions within a cutoff distance (r_c). This method has been shown to suffer from the following artifact : Consider the radius-rc sphere around a particle. In general, this sphere would be charged as exactly same number of positive and negative charges would be required to make it neutral. During the course of dynamics, charged particles can move back and forth the boundary at r_c, thus creating an artificial effect due the boundary(ref: Allen and Teldsley).

4.7.2 Ewald Summation

One way of handling the aforementioned issues in an efficient manner is to use the Ewald summation technique (ref, Ewald 1921). To understand this technique, lets represent the relation between the charge distribution and the coulombic potential in the differential form (Poisson equation) :

$$\Delta\phi(\boldsymbol{x}) = -\frac{1}{\epsilon}\rho(\boldsymbol{x})$$

where, $\phi(\boldsymbol{x})$ is the potential at point $\boldsymbol{x}$, $\rho(\boldsymbol{x})$ is the charge density at point $\boldsymbol{x}$ and ϵ is the permissivity of the medium. This equation is an elliptical partial differential equation(pde) of the second order. The standard way to determine the potential from this equation is a two step method - discretization of the equation followed by solution. These techniques however depend on the smoothness of the functions - ρ and ϕ - involved. However, in the case of charge distribution in our simulation system, ρ is a set of delta functions which are clearly not smooth! As ρ is not smooth, ϕ is not smooth either.

Ewald method is based on replacing the point charge distributions by smooth charge distributions in order to use the fast solution techniques of the pde. The most common smooth function used in Ewald method is the gaussian distribution although other distributions have been used as well. Thus,

$$\rho = \rho^{sr} + \rho^{lr}$$

$$\rho^{sr} = q\delta(\boldsymbol{x} - \boldsymbol{x_i}) - qG$$

$$\rho^{lr} = qG$$

Direct space or short range charge (ρ^{sr}) consists of the original point charge screened by the gaussian-distributed charge (G)of the same magnitude.

$$E^{sr} = \frac{1}{2}\sum_{n\in Z^3}{}' \sum_{i,j} q_i q_j \frac{erfc(\alpha|r_{ij} + nL|}{|r_{ij} + nL|}$$

Unlike the potential due to the original charge, the potential due to direct space charge decays rapidly as shown in the figure. This is due to erfc function which decays very fast. In fact, it decays even faster than the Van der Waals term r^{-6} and hence the cutoff used for Van der Waals can be used for direct space coulombic potential calculation as well.

Potential due to the long range charge however doesn't decay rapidly and if calculated in the direct space would require summation over the infinite images. However, as we discussed earlier, the smoothness of the charge ρ^{lr} (and hence potential (ϕ^{lr}) allows the use of fast pde solvers. Fourier based

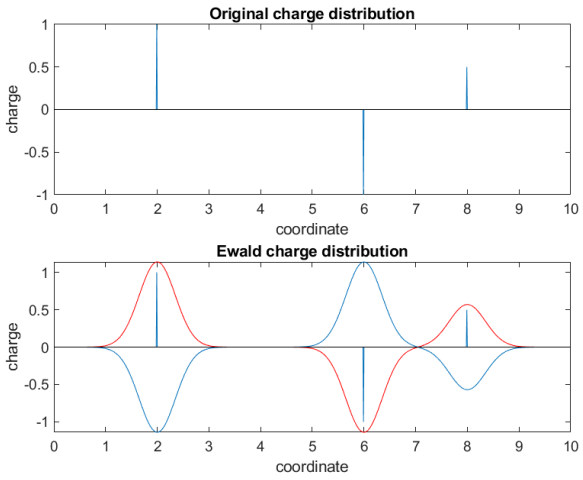

Figure 4. Screening charge distribution. (top) Original charge distribution. (bottom)Point charges can be split into Direct space(blue) and Reciprocal space charges(red). Direct space charge consists of the original charges and gaussian-distributed screening charge. Reciprocal space charge is only the gaussian-distributed charge.

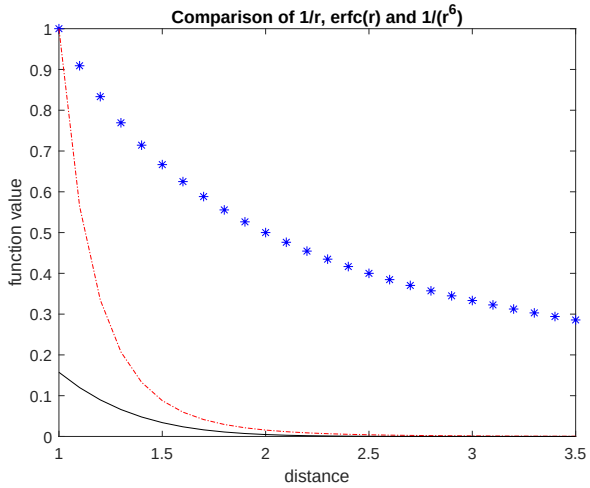

Figure 5. Comparison of decay of original r^{-1} term(blue,*), erfc(r) in direct space(black,-) and r^{-6} in van der waals term (red, -.).

solvers use the important result that differentiation operation in direct space corresponds to multiplication by (ik) in reciprocal space!

$$E^{lr} = (const) \sum_{k \neq 0} \frac{\exp(-k^2)}{k^2} |\rho(k)|^2$$

The interesting factor here in E^{lr} is the $\exp(-k^2)$ which decays exponentially. This means that we need to calculate this term only for a small values of $|k|$ before the term dies down. If we used only the original point charges and not the gaussian-distributed charges as we are using in reciprocal space, we would not have this dampening factor.

The final term in Ewald summation is the so-called self term.

$$E^{s} = (const) \sum_{i} q_i^2$$

It is not a physical quantity and comes up due to the way the reciprocal space Ewald summation is set up. It has to be subtracted out from the Ewald summation. Interestingly, self term needs to be calculated only once at the beginning of the simulation as it depends only on the charges and does not change with a change in the positions of the charges. For the same reason, it does not contribute to the force either.

4.7.3 Grid based Ewald summation

Ewald summation, through Fourier transform, as described in the previous section takes $O(n^{3/2})$ time. Discrete Fourier Transform (DFT) provides a faster way of solving the problem and reduces the time complexity to $O(nlog(n))$. In order to use DFT, the charge distribution has to be discretized over a grid. Particle-Particle Particle Mesh (P3M), Particle Mesh Ewald (PME) and Smooth Particle Mesh Ewald (SPME) are three different implementations of the grid-based solution and are very similar in spirit though the details are slightly different. The choice of specifics in each one is based on a combination of accuracy, speed and ease of implementation. In this subsection, we give an overview of the grid-based approach.

There are five general steps involved in these methods :

1. Charge assignment : In this step, charges are interpolated onto the grid. While the original PME method uses Lagrangian interpolation for charge assignment, the SPME method uses the smoother cardinal B-splines(hence the name Smooth-PME). These interpolation functions make sure that the charge density is distributed over the neighboring grid points.
2. Transformation of the grid to reciprocal space : Fast Fourier Transform (FFT) is used to convert the charges on the grid to their equivalent fourier space structure factors.
3. Energy calculation : Reciprocal space potential is calculated by solving the Poisson equation in Fourier space. As noted earlier, solving the Poisson equation in Fourier space is simply a division by k^2, where k is the reciprocal space frequency. At the same time, the grid is modified to store the reciprocal space potential.
4. Transformation of the grid back to real space : Inverse FFT is used to convert the reciprocal space potential back to the real space.
5. Force calculation : Force is given by the gradient of the energy. PME, SPME and P3M use different methods for

calculating it. SPME specifically calculated it by analytic differentiation in the real space.

Optimizing the performance of grid based methods is tricky as many of the choices would already have been made in the implementation of the method. As a user, there are a few ways to tweak the performance of the SPME :

- Grid dimensions : A fine grid would be slower as the interpolation and calculations would have to be done at larger number of points. However its accuracy will be higher than a coarse grid.
- Screening function : Some variants of Ewald summation have used screening function other than gaussian. In SPME implementations, one uses the Gaussian. The screening function can be varied via the spread of the gaussian. However, as discussed earlier, it is tightly coupled with direct space cutoff.
- Cutoff of the direct space : Although it can be changed, it is generally kept the same as van der waals cutoff for the ease of implementation. Decreasing the cutoff improves the direct space performance but increases the complexity of the reciprocal space calculations.

DLM: Need to write some kind of wrap-up/conclusion rather than just ending abruptly. Also probably should mention again data analysis and point to Zuckerman work.

DLM: Perhaps also a brief “what NOT to do with your MD data” blurb, e.g., don’t just make movies and look at them. Don’t treat them as the answer. Don’t overinterpret, etc.

DLM: Also need to point out the checklist below and discuss it in the text somewhere.

DLM: I also need to go over the checklist again and make sure it is what we want/addresses key issues (and everything there is addressed in the text.

TAKE STOCK OF YOUR PLANS

- ☐ **Count the cost:** Think about what you know about the timescales of what you want to observe and determine whether it is tractable to simulate this given the size of your system, your computational resources, and the expense of the simulation.
- ☐ Pick the desired ensemble (*NVT*, *NPT*, *NVE*, μVT, μPT)
- ☐ Determine reference states that you are trying to emulate/discover.
- ☐ What temperature, pressure, etc. are you interested in?
- ☐ What is already known in the literature and what data do you wish to compare to?

PREPARE TO IMPLEMENT YOUR PLANS AND MAKE CRITICAL DECISIONS ABOUT SYSTEM TYPE

- ☐ Choose a simulation package suitable for simulating that ensemble (see best practices document)
- ☐ Determine whether you are simulating a bulk (typically periodic) or finite system and choose the appropriate cutoff types and periodicity (full periodicity for bulk systems, partial periodicity for interfaces, etc.) as discussed in [section]

DETERMINE HANDLING OF CUTOFFS

- ☐ As a general rule, electrostatics are long-range enough that either the cutoff needs to be larger than the system size (for finite systems) or periodicity is needed
- ☐ Nonpolar interactions can often be safely treated with cutoffs of 1-1.5 nm as long as the system size is at least twice that, but long-range dispersion corrections may be needed

CHOOSE APPROPRIATE SETTINGS FOR THE DESIRED ENSEMBLE:

- ☐ Pick a thermostat that gives the correct distribution of temperatures, not just the correct average temperature
- ☐ Pick a barostat that gives the correct distribution of pressures
- ☐ Consider the known shortcomings and limitations of certain integrators and thermostats/barostats and whether your choices will impact the properties you are calculating

CHOOSE AN APPROPRIATE TIMESTEP FOR STABILITY AND AVOIDING ENERGY DRIFT

- ☐ Determine the highest-frequency motion in the system (typically bond vibrations unless bond lengths are constrained)
- ☐ As a first guess, set the timestep to approximately one tenth of the highest-frequency motion's characteristic period
- ☐ Test this choice by running a simulation in the microcanonical ensemble, and ensure that energy is conserved