

Best Practices for Foundations in Molecular Simulations : v0.1

Efrem Braun¹, Justin Gilmer², Heather Mayes³, David L. Mobley⁴, Jacob I. Monroe⁵, Samarjeet Prasad⁶, Daniel M. Zuckerman⁷

¹University of California, Berkeley; ²Vanderbilt University; ³University of Michigan; ⁴University of California, Irvine; ⁵University of California, Santa Barbara; ⁶National Institutes of Health; ⁷Oregon 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 July 30, 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); hbmayer@umich.edu (HM); dmobley@mobleylab.org (DLM); jimonroe@uconn.edu (JIM); p.samar.j@gmail.com (SP); zuckerm@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 quite simple; 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 equation of motion is numerically integrated and a dynamical trajectory of the system is generated. 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. This foundational docu-

ment 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, but MC as a whole needs its own 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 classical 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) is used to calculate non-bonded as well bonded interactions. Classical 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.

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 [? ?]. Recent 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, but the field would like to reach even longer timescales, meaning that switching to a more detailed energy model is only done with some trepidation 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 its chemical identity is a constant over the course of the simulation.

DLM: Refs?

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" [?], Frenkel and Smit's "Understanding Molecular Simulations" [?], and Tuckerman's "Statistical Mechanics: Theory and Molecular Simulation" [?], though there are many other sources.

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 ???. 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 particles 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 a very small time step, so for computational efficiency water is often instead treated as a rigid body. Implementation of rigid bodies in a simulation protocol involves 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 ?? for more on constraints and integrators). It is important to understand the concept of point particle, 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 Hamilto-

nian formulation and therefore basic knowledge of Hamiltonian mechanics is essential. Classical mechanics has several conserved quantities, for example, the total energy of the 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. 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 thermodynamics 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. It is important to understand key concepts in thermodynamics, such as temperature, pressure, internal energy, various forms of free energy and relationship between them. Basic properties of these quantities should be faithfully mimicked by simulation methods. It is quite straightforward to calculate internal energy, temperature and pressure from simulations, and these calculations can serve as quick sanity checks for the simulation setup. Clear understanding of these concepts is absolutely essential for grasping certain technical aspects of MD simulations namely, thermostat and barostat, which will be discussed in later sections. Basic thermodynamic principles also dictate proper simulation protocols and associated best practices.

DLM: The discussion we have here of thermodynamics key concepts is way shorter than the stat mech key concepts below. Perhaps need to address; perhaps in an update?

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.

- Atkins' physical chemistry book [?] : chapters 1 to 4.
- Zemansky and Dittman [?]: Chapters 1 to 4, 5, 8 and 15.

DLM: Pull in references from https://github.com/MobleyLab/basic_simulation_training/issues/23 here

3.3 Classical statistical mechanics

3.3.1 Key concepts

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

- Ensembles, distribution functions for different ensembles. Equivalence of ensembles
- What equilibrium means and the difference between equilibrium and non-equilibrium. For instance, what is usually called an "equilibrium trajectory" generally will not embody a good sample of the equilibrium ensemble due to insufficient sampling. On the other hand, truly non-equilibrium conditions such as driven transitions and relaxation are fundamentally different. Note that relaxation can occur to the equilibrium ensemble or a non-equilibrium condition (e.g., steady state).
- Clarify differences between nonequilibrium ensembles: driven nonequilibrium steady-state, systems driven out of equilibrium by a time-dependent field, systems initially out of equilibrium but relaxing to equilibrium
- Time averages and ensemble averages
- Fluctuations
- Correlation functions

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 $\exp[-U(\mathbf{r}^N)/k_B T]$, 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.

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. ??(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 –

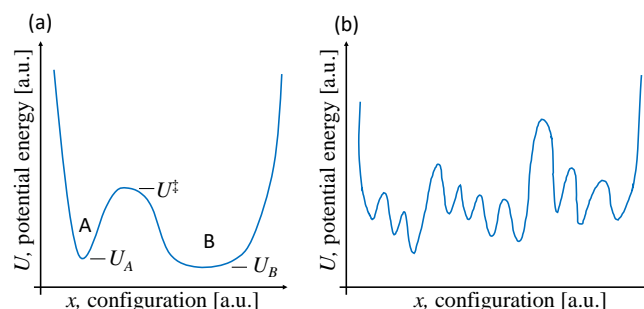


Figure 1. Energy landscapes. (a) A highly simplified landscape used to illustrate rate concepts and (b) a schematic of a complex landscape with numerous minima and ambiguous state boundaries.

the "dwell" timescale – which increases exponentially with the barrier height according to the (reciprocal) Arrhenius factor as $\exp[(U^\ddagger - U_A)/k_B T]$; barriers many times the thermal energy $k_B T$ imply long dwells. The rate k_{AB} , which is the transition probability per unit time, exhibits reciprocal behavior – i.e., $k_{AB} \sim \exp[-(U^\ddagger - U_A)/k_B T]$ 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. ??(b). Each small basin represents something like a different 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. 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 ensemble. 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: $p_A^{\text{eq}} \sim \exp(-F_A/k_B T)$. Because equilibrium behavior is caused by dynamics, there is a fundamental connection between rates and equilibrium, namely that $p_A^{\text{eq}} k_{AB} = p_B^{\text{eq}} k_{BA}$, 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. ??), 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 for constant T and G for both constant T and pressure – 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

- Reif
- McQuarrie
- Dill
- Hill
- Shell
- Zuckerman
- Kaznessis

DLM: Pull in references from https://github.com/MobleyLab/basic_simulation_training/issues/23 here

3.3.3 Online resources

- 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 as $1/r^3$ 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 charge in proximity.

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 ??) 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 ??, below).

3.4.2 Books

- Griffiths
- Jackson

DLM: Pull in references from https://github.com/MobleyLab/basic_simulation_training/issues/23 here

DLM: I removed the unwritten “stochastic dynamics” sub-section for now; I am not sure it is critical, and we could add later if we decide it is. (We address Langevin elsewhere.

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 non-polar 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.

3.5.2 Books

- “Intermolecular and surface forces” by Jacob N. Israelachvili

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 ??, with the most common 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 ?). 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. More exotic potentials based on three-body intermolecular orientations, or terms directly coupling bond lengths and bending angles are also possible. 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

¹ It is worth noting there is 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.

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 (see Part II of Amber reference manuals[?] and Ch. 4 of the reference manual for GROMACS [?]).

JIM: I just referenced the manuals that I'm familiar with... additional citations would be appreciated and seem appropriate. Right?

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 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 provided within a distributed molecular dynamics package. 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 [? ?].

For almost all force fields, many versions, variants, and modifications exist. Each force field version not only includes specific functional forms to describe intermolecular potentials, but also specifies the parameters governing such interactions. Parameters are usually specific to certain types of atoms, bonds, molecules, etc., and include point charges on atoms if electrostatic terms are in use. A subtle and often overlooked point involving force field implementation concerns the consistent treatment of constraints, cut-offs or other simulation settings that impact the potential energy function. To exactly replicate a described parametrization of a force field, constraints should be specified consistently in addition to usage of the same intermolecular cut-offs as used in the original paper. 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 properties of interest are desired. This is especially important for the purposes of

free energy calculations, where the potential energy itself is recorded.

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

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

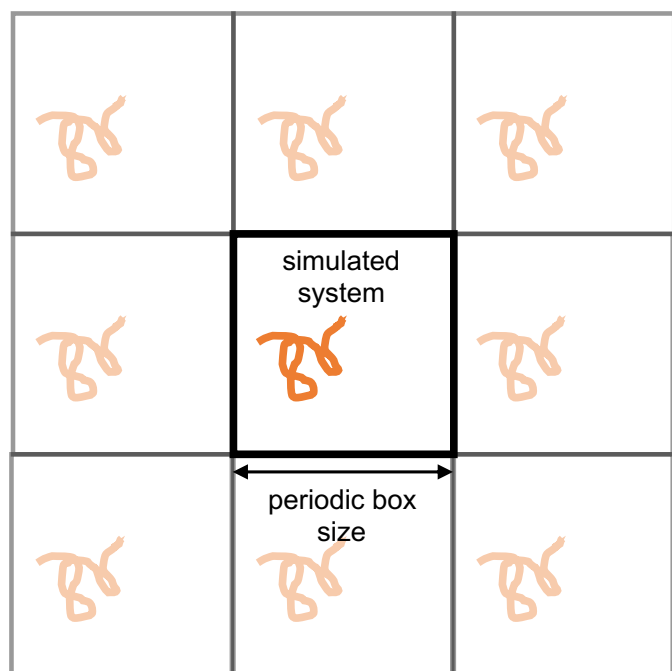


Figure 2. Periodic boundary conditions are shown for a simple 2D system. Note that the simulated system is a sub-ensemble within an infinitely sized system of identical, small ensembles.

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 ?? 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, 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 ?? . 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.

JIM: Other examples to cite? I think there's some examples also involving calculations of entropy, but I'll have to check this

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. Addition-

ally, 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.

System preparation is the most variable of these steps, and often requires unique tools for every system of interest. 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, it may be that a freely-available tool for constructing such a system does in fact exist. Examples include tools for constructing specific crystal structures, proteins, and lipid membranes, such as Moltemplate, Packmol, and AtomsK. The goal of all of these tools, and system preparation in general, is to create a representation of the system of interest that can be interpreted by the desired simulation package. It is further desirable that this representation not vary too far from the known, 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. However, for a force field that reliably reproduces the energetics of a system, a starting configuration that is close to equilibrium is only a time-saving convenience in that it greatly reduces the equilibration time and overall simulation length by preventing trapping in metastable states.

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 time step). 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.

At the end of energy minimization, it is important to achieve a system configuration with small enough forces that the desired time step will well-approximate the dynamics (see [?], section 7.3.4). Such a configuration is a suitable starting point for molecular dynamics techniques. However, this only represents a static set of positions, while the propagation of dynamics also requires a set of starting velocities. These may be assigned in a variety of ways, but are usually randomly assigned to atoms such that the correct Maxwell-Boltzmann distribution at the desired temperature is achieved. Even though velocities are assigned according to the correct distribution, the selected thermostat will still usually need to add heat to 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. Once the kinetic and potential energies fluctuate around constant values, 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 ??), and the temperature fluctuates with the kinetic energy at each time step. Similarly, equilibration in the NPT ensemble is necessary before production in the NVT if an average density consistent with a specific pressure is desired. In this case, the system may be scaled to the desired average volume before the production simulation. In the above example, the 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. The schematic below (??) 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.

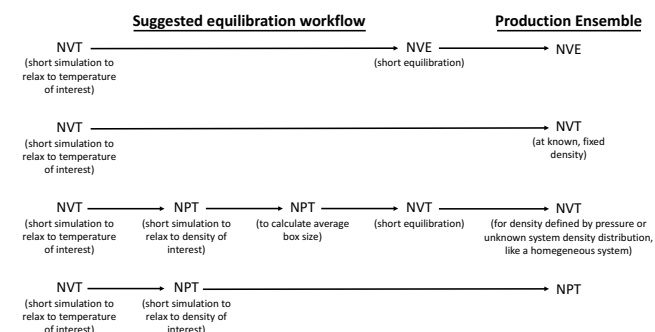


Figure 3. Common equilibration work-flows

Once equilibration is complete, the production simulation may be performed. The production simulation is that from which specific properties of the system of interest will be calculated. As mentioned above, the equilibration procedure should be selected that is appropriate for the desired production ensemble. It should be noted that “equilibration” within the production run may still be necessary before properties or metrics are computed from this simulation (see [?], lecture on Computing properties). This falls under the category of correctly obtaining unbiased statistics and conver-

gence, which is covered in a separate Best Practices document (<https://github.com/dmzuckerman/Sampling-Uncertainty>). Otherwise, if a brief simulation in the same ensemble is not performed during the equilibration step immediately prior to production, any period of the production simulation should be ignored where drift is observed in the energies, temperatures, pressures, densities, or other defining state-variables of the ensemble. This of course precedes estimation of convergence in property calculation. For more specific details on procedures and parameters used in production simulations, see the appropriate best practices document for the system of interest.

4.4 Thermostats

4.4.1 Thermostats seek to maintain a target temperature

As mentioned throughout the article above, molecular dynamics simulations are used to observe and glean properties of interest from some system of study. However, these properties traditionally are not measurable from the initial configuration of the system, nor is that usually physically meaningful. This generally requires transitioning the system to some other state point to collect the proper data once the system has equilibrated. 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. This section will review the 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.

4.4.2 Background and How They Work

The difference between the types of temperatures that are measured in a simulation is important to know. There is the target temperature — a temperature value that we attempt to maintain the system near. In other words, the target temperature is the specified temperature which a thermostat is used to maintain. However, due to the fluctuations, it is not guaranteed that the temperature measured at single point in time will be the target temperature; in reality, the instantaneous temperature should undergo fluctuations around the target temperature. The instantaneous temperature is also known as the kinetic temperature and is typically computed via the equipartition theorem. It should be noted that this value can be greater than, less than, or equal to the temperature corresponding to the average kinetic energy. The kinetic temperature should also be paid special attention to when describing the system. The kinetic temperature does **not**

state “the temperature is at some value T ”, it merely states: “the energy of the system is similar to that of a system at temperature T ”.

Finally, there are generally three types of thermostats: deterministic, stochastic, and extended system. Deterministic thermostats are usually known to scale the momenta or forces on the particles in the system based on their fitness to the target temperature. Stochastic thermostats behave similarly to the deterministic thermostats, however, there is a random force or momenta sampled out of a probability distribution that can scale or modify these properties of selected particles. Extended system thermostats are similar to what their name suggests, the system has an added variable that has a degree of freedom. This emulates an external heat bath that can interact with the particles in the system affecting their momenta in a predictable fashion.

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, just some of the more popular and historic thermostats used in MD.

Deterministic Thermostats

1. 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 as well. This thermostat relies on rescaling the momenta of the particles every N time steps based on their ratio of the instantaneous temperature to the target temperature[?]. This leads to multiple issues when trying to sample data using this thermostat. First, this method is not reversible, there lacks a way for the particles to have knowledge of their previous thermal history. This makes any dynamical value impossible to measure (diffusion for example). There is also no distribution of momenta; this thermostat does not sample the canonical distribution and cannot be used for sampling at thermal equilibrium. The isokinetic (constant kinetic energy) ensemble samples the same configurational phase space as the canonical ensemble, so position-dependent equilibrium properties can be obtained equivalently with either ensemble[?]. However, the isokinetic ensemble is not properly sampled by the simple velocity rescaling thermostat, and its usage can lead to simulation artifacts, so it is not recommended[?].

2. Hoover (Gaussian)

EB: I recommend referring to this thermostat as the Gaussian Thermostat. I've seen it referred to as the Hoover-Evans thermostat as well (I haven't seen it called the Hoover thermostat, but I'm sure somebody has called it that), but simply calling it the Gaussian thermostat avoids wading into the argument about who came up with it first.

The goal of the Gaussian thermostat is to ensure the change in instantaneous temperature ΔT is always 0 ($\Delta T \equiv 0$); 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 the method to solve for the smallest perturbative forces needed to keep the change in temperature equal to 0. Using the Gaussian principle of least constraint, forces are calculated to maintain a net 0 change in temperature, while minimally perturbing the system[?]. This thermostat is reversible and it samples the isokinetic ensemble. Position-dependent equilibrium properties will therefore be in agreement with the canonical ensemble (as explained above), but dynamical properties will not. Due to the nature of the thermostat preventing a change in temperature, whatever instantaneous temperature the system is at when the thermostat is applied, it will stay there.

3. Berendsen

The Berendsen thermostat is another type of velocity rescaling thermostat, but instead of the rescaling happening very abruptly each force adjustment time, we include a relaxation term to allow the system to more slowly approach the target temperature[?]. With the introduction of a coupling parameter τ , the strength of this parameter will determine how fast the system will approach the target temperature. However, it should be noted that while allowing for temperature fluctuations, the Berendsen thermostat is not canonical nor is it reversible. Its usage can lead to simulation artifacts, so it is not recommended[?].

Overall, *none* of these deterministic thermostats are suitable for most production-level simulations, though the Gaussian thermostat has uses in some advanced applications[?], because they do not sample from the canonical ensemble [?]. The simple velocity rescaling and Berendsen thermostats can also lead to unexpected problems in systems with weak or poor coupling between degrees of freedom, such as the “flying ice cube” [?] problem in some systems, and problems in alchemical free energy calculations where portions of the

system are decoupled from the rest of the system (e.g. very rapid movement of atoms in the decoupled portion of the system); they should not be used. To some extent, however, the choice of thermostat may depend on the property being calculated; e.g. for transport properties and kinetics, rather different issues may need consideration [?].

Stochastic Thermostats

1. Andersen

The Andersen[?] thermostat is the first of three stochastic thermostats that will be explored. This thermostat has vestiges of the rescaling thermostats mentioned earlier, but with a randomization element also included. In this case, the Andersen thermostat attempts to model a system that is coupled to some random degree of freedom[? ?]. In this case, this random element will be a new set of velocities that are chosen from a probability distribution and then if the reassignment probability passes, the velocities of the particle will be replaced. The number of particles affected, time between “collisions”, and how often this is applied to the system are possible variations of this thermostat. Like the previous thermostats, this is not reversible, but the Andersen thermostat does reproduce the canonical ensemble. There are two versions of this thermostat, one which applies massive collisions (reassignment of all velocities) sporadically, and another which applies more gradual buffeting of selected particles.

2. Langevin

The Langevin[?] thermostat applies the idea of an implicit solvent as a way to randomly impart or remove some force acting on the particles. With 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 parameter used to tune the “viscosity” of the implicit bath, and F_{random} is a random force perturbing the forces on the particles. Careful consideration must be taken when choosing the friction damping parameter. However, in many cases this thermostat is an excellent option for molecular simulations.

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

The Bussi[?] thermostat is a velocity rescaling thermostat 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. 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 dynamic properties are fairly independent from the choice within a broad range.

Extended System Thermostats

1. Nosé-Hoover

The Nosé-Hoover thermostat [?] essentially abstracts away the thermal bath from the previous thermostats and condenses it into a singular additional degree of freedom. This bath with a “mass” that can be changed can interact with the particles in the system in a predictable and reproducible way while maintaining the canonical ensemble. This is one of the most widely implemented and used thermostats due to these attributes. However, it should be noted that with small enough systems, ergodicity can be an issue and the thermostat might not properly sample the canonical ensemble [? ?]. 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.

2. Nosé-Hoover Chain

Briefly mentioned above, there are certain conditions where the Nosé-Hoover thermostat might not be sufficient to properly sample the system, due to system size and ergodicity issues[? ?]. However, Martyna et. al. [?] discovered that by coupling more heat baths to the system, the canonical ensemble can be rediscovered, at the minimal increase in computations required. In certain situations, it will be useful to chain additional heat baths to the system when under the Nosé-Hoover thermostat.

4.4.4 Summary

To summarize, observe (??), as a general summary and guide for exploring the usage of various thermostats. Make sure to pay special attention to the reversibility when measuring time dependent properties, the ensemble sampled for all cases, and if a proper distribution of momenta and kinetic energy fluctuations are supported. Do note that depending on the system of interest, it might not be necessary to worry about some of this information when initializing the system for a production run. For example, if the thermal history of the system is not necessary during equilibration, a faster algorithm like Andersen or Berendsen could possibly be employed, with a switch to Nosé-Hoover for the production run. Knowing the

system you are simulating and the benefits and weaknesses to each thermostat is crucial to successfully and efficiently collect meaningful, physical data. In general, deterministic thermostats should never be employed for production use; Langevin, Andersen, and Nosé-Hoover chains can serve as reasonable general-purpose thermostats, and other choices should be employed only with considerable caution.

Thermostat	Physical?	Reversible?	Canonical?	Kinetic Energy Fluctuations?
DETERMINISTIC				
Velocity Scaling	X	X	X	X
Hoover	X	X	X	X
Berendsen	X	X	X	✓
STOCHASTIC				
Andersen	X	X	✓	✓
Langevin	X	X	✓	✓
EXTENDED SYSTEM				
Nosé-Hoover	X	✓	(✓)	✓
Nosé-Hoover Chain	X	✓	(✓)	✓

Figure 4. Basic summary of popular thermostats, where X signifies that the thermostat does not fulfill that statement, ✓ does, and (✓) does under certain circumstances.

EB: The Bussi thermostat should be added to this graphic under the stochastic grouping, with the canonical and fluctuations boxes checked.

4.5 Barostats

- What is it?
- Brief description of how it works
- Popular barostats

4.6 Integrators

- Numerical solution of dynamical equations of motions
- Importance of energy conservation
- Commonly used integrators

4.6.1 How to choose an appropriate time step?

DLM: Need to switch consistently to “time step” or “timestep”; ran out of time to do this on the plane

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. Obviously, this discretization process introduces errors, as quickly becomes apparent if solutions from integrators are compared to dynamics for systems for which analytical solutions do exist (like a simple

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 [?]. The reason for this outlined in greater detail in

JIM: Maybe add citation to Tuckerman paper from 1992

, but the basic idea is that the mathematical operator associated with an integrator must be unitary in order for phase space volume to be preserved. By definition of a unitary operator, the application of the operator to negative times is equal to applying the inverse of the operator to an equivalent positive time interval. This is in fact also 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.

The maximum time step 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[? ?].

Energy conservation 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.

DLM: Probably need to address energy drift, as well as how energy changes should scale with timestep.

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 a general rule of thumb, the timestep must be at least as small as an order of magnitude less than the smallest timescale of motion present in the system (often this is bond vibration). However, depending 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.

The most commonly used integrators are variants of the Verlet algorithm. Detailed discussion and derivation of such integrators may be found in Leach [?] and Frenkel and Smit [?].

JIM: Insert citations and correct sections in these books

Such integrators are not applicable, however, for simulations involving stochastic dynamics. These 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 ??.

JIM: Need to cite the right chapter in McQuarrie, and others?

As detailed in section ??, 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).

JIM: Need to cite Fass, Chodera, Crooks, etc.

Again, this may be remedied through a shorter timestep, but thus is more computationally expensive and increases truncation error.

JIM: Happy to introduce Trotter decompositions, but is it really necessary? Also, we need to add information on constrained dynamics. Anything else? Needs more details, or just send people to citations?

DLM: I don't think necessary to introduce, but in favor of adding citations to useful work/additional resources.

with Periodic Boundary Conditions. The Journal of Physical Chemistry B. 2004; 108(40):15873–15879. <http://dx.doi.org/10.1021/jp0477147>, doi: 10.1021/jp0477147.

4.7 Long range electrostatics

- Cut-off is bad
- Need for special treatment
- Idea of an Ewald sum
- PPPM
- How to choose parameters

References

- **Leach AR**. Molecular Modelling: Principles and Applications. Second ed. Essex, England: Pearson Education Limited; 2001.
- **Lemkul J**, GROMACS Tutorials;. <http://www.bevanlab.biochem.vt.edu/Pages/Personal/justin/gmx-tutorials>.
- **Madej B**, Walker R, AMBER Tutorial B0: An Introduction to Molecular Dynamics Simulations Using AMBER;. <http://ambermd.org/tutorials/basic/tutorial0/index.html>.
- **Shell MS**, Principles of modern molecular simulation methods: Lecture Notes;. <https://engineering.ucsb.edu/~shell/che210d/assignments.html>.
- **Yeh IC**, Hummer G. System-Size Dependence of Diffusion Coefficients and Viscosities from Molecular Dynamics Simulations

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

- ☐ This depends on factors such as the use of constraints in the system (e.g. for all-atom systems constraining hydrogen bonds can allow the use of a slightly longer timestep; 2 fs is relatively typical)