THE JOURNAL OF PHYSICAL CHEMISTRY B
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

pubs.acs.org/JPCB

Article

Type Label Framework for Bonded Force Fields in LAMMPS

Published as part of The Journal of Physical Chemistry B virtual special issue "Recent Advances in Simulation Software and Force Fields".

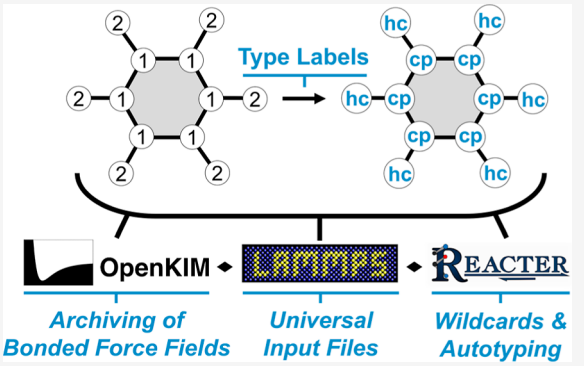
Jacob R. Gissinger,* Ilia Nikiforov, Yaser Afshar, Brendon Waters, Moon-ki Choi, Daniel S. Karls, Alexander Stukowski, Wonpil Im, Hendrik Heinz, Axel Kohlmeyer, and Ellad B. Tadmor*

Cite This: *J. Phys. Chem. B* 2024, 128, 3282−3297

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: New functionality is added to the LAMMPS molecular simulation package, which increases the versatility with which LAMMPS can interface with supporting software and manipulate information associated with bonded force fields. We introduce the "type label" framework that allows atom types and their higher-order interactions (bonds, angles, dihedrals, and impropers) to be represented in terms of the standard atom type strings of a bonded force field. Type labels increase the human readability of input files, enable bonded force fields to be supported by the OpenKIM repository, simplify the creation of reaction templates for the REACTER protocol, and increase compatibility with external visualization tools, such as VMD and OVITO. An introductory primer on the forms and use of bonded force fields is provided to motivate this new functionality and serve as an entry point for LAMMPS and OpenKIM users unfamiliar with bonded force fields. The type label framework has the potential to streamline modeling workflows that use LAMMPS by increasing the portability of software, files, and scripts for preprocessing, running, and postprocessing a molecular simulation.

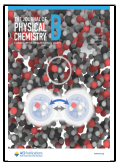


1. INTRODUCTION

Classical fixed-bond or *bonded* force fields are a mainstay of molecular modeling, capable of approximating the dynamics of materials from ionic liquids[1] to metal organic frameworks[2] to biological molecules[3] such as viruses. One of the strengths of bonded force fields is their ability to accurately represent local chemical environments through the definition of atom types, which can then be used in any setting in which the environment occurs. For example, after parametrizing a force field for the oxygen in a hydroxyl group, these parameters can generally be used for any hydroxyl group, no matter the type of molecule to which it is attached. The consistent representation of chemical environments and the ability to mix different compounds without additional parametrization is a powerful aspect of bonded force fields, compared to reactive interatomic potential functions that algorithmically determine the nature of bonding between atoms based on their chemical species and environment, such as bond-order[4] or machine-learned[5] potentials. However, defining atom types explicitly means that these types must be assigned manually to each atom and that bonded force fields do not intrinsically support changes in covalent bonding, such as during chemical reactions or defect formation. Herein, we present an enhanced framework for using bonded force fields within the popular LAMMPS molecular simulation package and the supporting tools.

Despite the utility of bonded force fields for describing a variety of organic and inorganic materials, setting up classical molecular dynamics (MD) simulations with these models remains a niche skill that requires significant expertise. In this work, we introduce the *type label* framework, which is aimed at making simulation files that use bonded force fields more human-readable, user-friendly, and universal. New functionality has been added to LAMMPS to represent atom types using the string labels used by bonded force fields, instead of arbitrary integers as traditionally done in LAMMPS. Type labels simplify workflows that involve multiple input files or modifying or referencing atom types. Type labels make it possible for LAMMPS to use bonded force fields archived in the online OpenKIM[6,7] repository of interatomic potentials with the benefit of increased simulation reliability and reproducibility. Further, for the REACTER[8,9] tool for modeling chemical reactions in classical MD simulations,

Received: December 26, 2023
Revised: February 10, 2024
Accepted: February 16, 2024
Published: March 20, 2024



https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282−3297

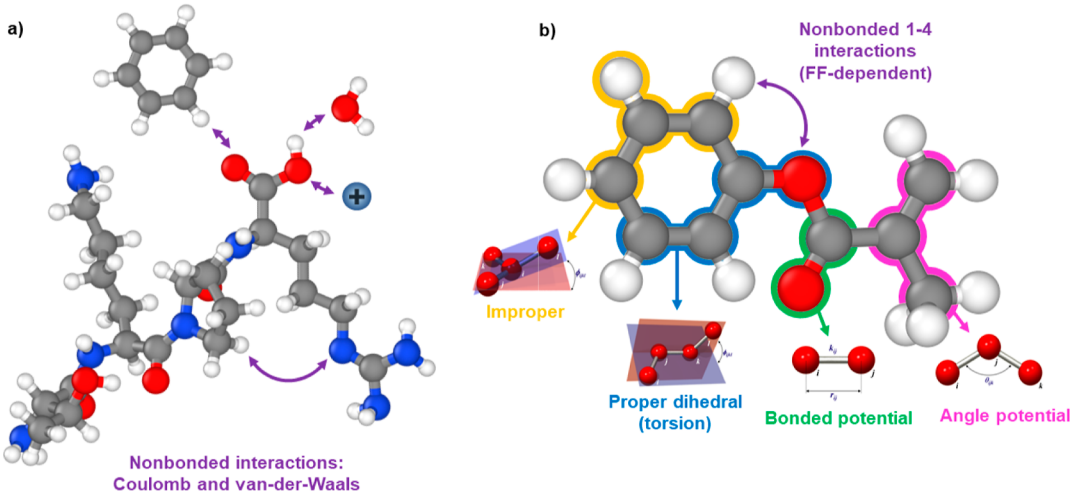


Figure 1. Overview of the atomic interactions that require type assignment in a bonded force field. (a) Nonbonded interactions, including van der Waals and electrostatic Coulomb, govern interactions between molecules, ions, solvents, and solid substrates (not shown). Additionally, intramolecular nonbonded interactions contribute to the conformation of molecular chains such as the small tuftsin peptide shown here. (b) Intramolecular covalent bonding is typically comprised of four "bonded" terms: bonds, angles, dihedrals, and impropers, shown here on a small organic molecule. These bonded interactions, especially dihedrals, combine with nonbonded interactions to determine the conformation and dynamics of molecular chains.

type labels enable more generic reaction templates and the automatic reassignment of certain force field parameters after a reaction. In this work, we begin with a primer on bonded force fields, including how they are constructed and used, and then describe the new capabilities that type labels facilitate for bonded force fields in LAMMPS, OpenKIM, REACTER, and related tools and visualization software.

2. PRIMER ON BONDED FORCE FIELDS

Bonded force fields include energy terms that specifically represent covalent bonding in addition to ionic bonding and dispersion interactions.[10−20] Covalent bonding is dominant in organic compounds and biomolecules such as hydrocarbons, proteins, and DNA, as well as in a range of inorganic compounds like silicates, nitrides, sulfides, organometallic compounds, and the sulfate, phosphate, and nitrate ions. Covalent and ionic bonding covers a spectrum from fully covalent bonding (e.g., H_2, S_8, graphene) to partially covalent bonding (e.g., esters, amides, aluminates), predominantly ionic bonding (e.g., $CaCl_2$, MgO, LiF), and comparatively rare fully ionic bonding (e.g., NaCl and CaF_2). Bonded force fields, therefore, play an important role in representing the nature of chemical bonding in molecular simulations. Specific bonded terms are typically included for atoms participating in bonds that are more than half covalent. The list of atoms and their connectivity is referred to as the *bonding topology*. When bonding is more than half ionic, only nonbonded terms are used. For borderline chemistries, either including or excluding specific bonded terms typically leads to similarly reliable predictions.

2.1. Bonded Force Field Energy Expression. A molecular force field is a function that provides the potential energy of an atomic configuration defined by the positions and species (chemical elements) of a set of atoms. The negative gradient of this energy with respect to the positions of the atoms gives the forces on the atoms that can then be used in an MD simulation (as done in a code like LAMMPS) to move the atoms iteratively in time and study the behavior of the configuration in isolation or its response to external stimuli.

There are two philosophies in the development of force fields.[21] (1) Reactive interatomic potentials take as input the atom coordinates and species and infer the nature of bonding from an atom's environment using complex functional forms, or more recently, machine learning methods. (2) Bonded force fields take a different empirical approach. The functional forms tend to be simpler, but different "atom types" are introduced for the same chemical element to reflect its chemical environment. There can be, for instance, many different types of carbon depending on the locations and species of the atoms that surround it. Here, we focus on bonded force fields.

The energy expression in popular bonded force fields is composed of "bonded" interactions related to covalent and partially covalent bonds and other "nonbonded" interactions including van der Waals and electrostatic interactions. The potential energy E_{pot} is taken as a sum of these bonded and nonbonded contributions. Bonded interactions include bond stretching (E_{bond}), angle bending (E_{angle}), bond torsions (also called dihedral angles, $E_{torsion}$), and impropers (also called out-of-plane terms, $E_{improper}$). Nonbonded interactions include Coulomb (E_{coul}) and van der Waals interactions (E_{vdW}) between atoms. Some force fields also include cross terms (E_{cross}). Together, we have

$$E_{pot} = E_{bond} + E_{angle} + E_{torsion} + E_{improper} + (E_{cross}) + E_{coul} + E_{vdw} \quad (1)$$

Bonded terms are illustrated in Figure 1 along with additional nonbonded interactions (Coulomb and van der Waals) that occur between all pairs of atoms that are not directly bonded, usually excluding 1−2 bonded atoms (connected by a single bond), 1−3 bonded atoms (connected by a chain of two bonds), and sometimes 1−4 bonded atoms (connected by a chain of three bonds). E_{pot} depends only on the current Cartesian coordinates of the atoms, regardless of prior positions, and is thus fully deterministic. Computation of E_{pot} requires knowledge of the atom types, atomic charges, bonding topology, and all applicable force field parameters and rules. These rules include combination rules, exclusion rules

https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282−3297

for nonbonded interactions between nearby bonded atoms, and cutoffs related to the computation of long-range electrostatics and van der Waals interactions.

Force fields with bonded terms are often categorized into "Class I" and "Class II", with the latter having a more elaborate functional form aimed at improved accuracy. Class I force fields include all energy terms, except cross terms that capture correlations between higher-order interactions. Class II force fields include these additional cross terms, which include bond−bond, bond−angle, angle−angle, torsion−angle, and torsion−torsion terms (all summed in E_{cross} and not further specified in eq 1). In addition, Class I and Class II force fields differ in the treatment of bond and angle terms. Bond and angle terms are usually harmonic potentials in Class I force fields and extended into an additive combination of harmonic, cubic, and quartic terms in Class II force fields.[16−18]

Specifically, common energy expressions for Class I force fields include the following terms (compare the annotated underbraces with eq 1)

$$
\begin{aligned}
E_{pot} = & \underbrace{\sum_{ij\ \text{bonded}} K_{r,ij}(r_{ij} - r_{0,ij})^2}_{E_{bond}} \\
& + \underbrace{\sum_{ijk\ \text{bonded}} K_{\theta,ijk}(\theta_{ijk} - \theta_{0,ijk})^2}_{E_{angle}} \\
& + \underbrace{\sum_{ijkl\ \text{bonded}} f[K_{\varphi,ijkl}, \cos(n\varphi_{ijkl} - \varphi_{ijkl,0})]}_{E_{torsion}} \\
& + \underbrace{\sum_{ijkl\ \text{bonded}} f(K_{\chi,ijkl}, \chi_{ijkl}, \chi_{ijkl,0})}_{E_{improper}} \\
& + \underbrace{\frac{1}{4\pi\epsilon_0} \sum_{ij\ \text{nonbonded}\ (1,2\ \text{and}\ 1,3\ \text{excl.})} \frac{q_i q_j}{r_{ij}}}_{E_{coul}} \\
& + \underbrace{\sum_{ij\ \text{nonbonded}\ (1,2\ \text{and}\ 1,3\ \text{excl.})} \epsilon_{ij}\left[\left(\frac{r_{\min,ij}}{r_{ij}}\right)^{12} - 2\left(\frac{r_{\min,ij}}{r_{ij}}\right)^{6}\right]}_{E_{vdw}}.
\end{aligned}
\tag{2}
$$

The parameters $K_{r,ij}$, $r_{0,ij}$, $K_{\theta,ijk}$, $\theta_{0,ijk}$, $K_{\varphi,ijkl}$, n, $\varphi_{ijkl,0}$, $K_{\chi,ijkl}$, $\chi_{ijkl,0}$ in eq 2 make up the bonded part of the force field and represent vibrational constants for bond stretching, equilibrium covalent bond lengths, vibrational constants for angle bending, equilibrium covalent bond angles, specifics of the torsion potential, and specifics of the improper energy corrections.[a] The functional forms of the torsion potential and the out-of-plane (improper) potential can differ from one force field to another. Torsion potentials typically contain additive cosine functions that describe preferred dihedral angles and related energy barriers for bond rotations for each type of dihedral angle present. Improper terms often take the form of a harmonic energy penalty when three atoms i, j, k are connected to the same central atom l and bend out of the common plane, or a prescribed angle $\chi_{ijkl,0}$. The parameters q_i, $r_{min,ij}$, and ϵ_{ij} constitute the nonbonded part of the force field and represent atomic charges, the equilibrium nonbonded distance between pairs of atoms, and the pairwise equilibrium nonbonded energy, respectively.

Equation 2 is written with a 12−6 Lennard-Jones (LJ) potential to represent van der Waals interactions, but other types of LJ potentials such as 9−6 or 14−7 are used as well (not shown here for brevity). Class I force fields usually follow eq 2 with a 12−6 LJ potential, for example, IFF,[20] CHARMM,[10] AMBER,[11] CVFF (the common variant without cross or Morse terms),[12] DREIDING,[13] and OPLS-AA.[14,15] Class II force fields include additional terms that are not shown here for simplicity, for example, CFF, PCFF,[16,17] COMPASS,[18] as well as a Class II variant of CVFF (including cross terms). CFF, PCFF, and COMPASS use all energy terms in eqs 1 and 2, plus additional quartic and cross terms and a 9−6 LJ potential instead of a 12−6 LJ potential. Also, 14−7 LJ potentials have been proposed in MMFF, a Class II force field,[19] and AMOEBA (which also uses additional multipole moments for specific functional groups).[22] Parameters for compounds that employ only a subset of terms of a given Class I or Class II energy expression can be flexibly used in combination with parameters for other compounds that utilize more or all Class I or Class II terms without adverse effects.

From a historical perspective, Class II force fields (CFF, PCFF, COMPASS) were introduced in the 1990s aiming at better representations of molecular interactions, conformations, and improved property predictions relative to Class I force fields.[17,18] The parametrization of molecular interactions in Class II force fields involved 6-31G* Hartree−Fock and similar less accurate ab initio methods in vacuum (e.g., B3LYP and LDA), especially related to dihedral angles and barriers, and fitting to experimental data. The aim was to improve property predictions by adding numerous higher order and cross terms in the Class II energy expression, as well as additional dipole moments and scaling factors.[17] However, Class II force fields continued to show significant deviations from the experimental data. Subsequent developments of IFF[20] and later generations of CHARMM,[23] OPLS-AA,[24] AMBER, and other Class I force fields in the 2000s showed that Class II additions are not necessary. Better physical interpretation and validation of Class I force field parameters, especially atomic charges, improved rationales for defining atom types, and more reliable quantum methods led to major improvements in accuracy. Data on structure and property prediction over the last two decades indicate that cross terms in Class II force fields have limited or no benefits to increased accuracy.

The bonded energy expressions shown in eq 2 are applicable to metals, ceramics, soft matter, and biomacromolecules.[20] The parameters can be assigned based on a clear physical-chemical interpretation. Parameters for 12−6 LJ potentials (eq 2) and 9−6 LJ potentials differ slightly for algebraic reasons. Parameters among force fields with 12−6 LJ potentials can also have some differences due to specific combination rules to obtain σ_{ij} and ϵ_{ij} for pairs of atoms of different atom types from the homoatomic interaction parameters σ_{ii} and ϵ_{ii}, following the conventions for the respective force field.[20] Another factor that can affect bonded as well as nonbonded parameters to a small extent are force-field-specific scaling rules for nonbonded interactions between 1−4 bonded atoms, e.g., they matter for polymers and midsize molecules and play no role for water, carbonate ions, or metals. Details of energy expressions, combination rules, and scaling factors for a variety of materials have been reviewed in ref 25.

https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282−3297

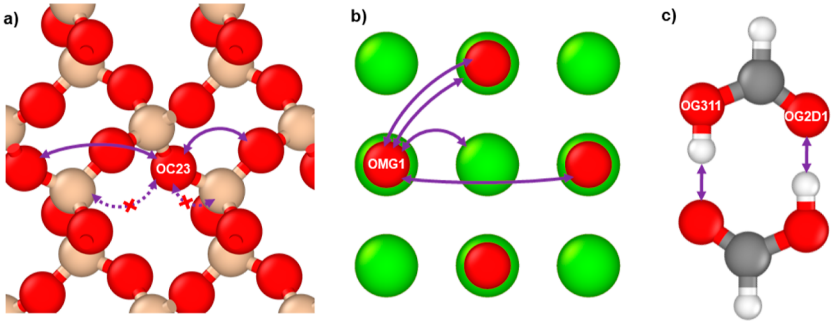


Figure 2. Different atom types are used to represent the interaction of a chemical element with different chemical environments. Arrows indicate nonbonded interactions. (a) Oxygen in silica, a covalent network solid, is assigned the atom type OC23 in IFF. Nonbonded interactions between two atoms are turned on or off depending on how many covalent bonds separate them. (b) Oxygen in magnesia, a largely ionic metal oxide modeled without covalent bonds, is represented by IFF atom type OMG1 with different interaction parameters compared to silica. (c) Carboxyl groups have two oxygen atom types within one molecule in this CHARMM model, one participating in a double bond (OG311) and the other in a single bond (OG2D1). The nonbonded parameters of these atom types must also accurately reproduce intermolecular interactions, such as the interaction between formic acid molecules to form a dimer shown here.

A practical aspect of using common 12−6 LJ potentials in simulations is the existence of two equivalent mathematical forms

$$E_{\mathrm{LJ}} = \epsilon_{ij}\left[\left(\frac{r_{\mathrm{min},ij}}{r_{ij}}\right)^{12} - 2\left(\frac{r_{\mathrm{min},ij}}{r_{ij}}\right)^{6}\right] = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^{6}\right] \quad (3)$$

The first representation is more common in the chemistry and biomolecular simulation community and uses $r_{\mathrm{min},ij}$ as the distance at the minimum potential energy—ϵ_{ij}, for example, in the force field parameter lists of IFF, CHARMM, and AMBER. The latter representation is more common in the physics community, which uses σ_{ij} as the interatomic distance where the energy is zero, and is employed in the LAMMPS parameter list. The variable σ_{ij} is smaller than $r_{\mathrm{min},ij}$ and the two quantities are related by $2^{1/6}\cdot\sigma_{ij} = r_{\mathrm{min},ij}$. The energy ϵ_{ij} is the same in both representations. When using different force fields and simulation programs, it is essential to verify whether $r_{\mathrm{min},ij}$ or σ_{ij} is being used and to convert between them as needed.

2.2. Geometry and Atom Types in a Bonded Force Field. When given the composition and geometry of a molecule or a periodic bonded structure, such as a biopolymer or a mineral, the questions to ask are (1) what are the atomic charges and then (2) what are the atom types? These two assignments are critical for the correct usage of a bonded force field in a molecular simulation. Here, we focus on the workflow of running a simulation, as opposed to force field development, and discuss the two steps in reverse order: we first discuss the assignment of atom types and topology and then atomic charges.

The species (chemical element) of a particle and the atom type specify differences in the chemical environment. We often encounter multiple atom types for one and the same element due to major differences in atomic charges, the number of bonded neighbor atoms, and the types of covalent bonds. For example, oxygen atoms in water, in a metal oxide, in silica, in an amide bond of a protein (−(C═O)−NH−), or in a dihydrogen phosphate ion with a fractional negative charge require different atom types since the bonded parameters as well as the nonbonded parameters are dissimilar. Carboxyl groups alone feature two distinct types of oxygen (Figure 2). Distinct atom types of metals in elemental form, in metal sulfides, and in metal ions of a specific valency in aqueous solution require different atom types and different force field parameters, since the atomic charges range from zero to their formal charges, the cohesion in solid metals is often an order of magnitude stronger than in aqueous ions, and metal salts can have predominant contributions by covalent bonding. Also, octahedral or distorted coordination geometries in metal ion complexes can necessitate multiple atom types of the same element to specify unique bond lengths, bond angles, and vibration properties. Furthermore, the chemistries of polymers such as acrylates, proteins, and DNA require multiple types of carbon atoms, oxygen atoms, nitrogen atoms, and hydrogen atoms to reflect different chemical environments, encode specific geometries, and have different torsion potentials that play a key role in the prediction of folding into secondary and tertiary structures.

Atom types can also include artificial particles such as virtual electrons in polarizable models for metals to account for image charges and applied voltages,[26,27] virtual π electrons in graphene-like structures and aromatic molecules to account for internal multipoles,[28,29] and polarizable core−shell ion pairs in Lorentz oscillator models to represent polarizability of ions or atoms in molecules[30,31] (often erroneously called "Drude models" that are used to represent a free electron gas as opposed to the intended oscillator models[26]). In these instances, a single atom is effectively split into a multi-component atom represented by multiple particles with distinct atom types.

In all instances, the atom types have a predefined mass given in the declaration section of a given force field. If a single particle represents one atom, as is often the case for atomistic simulations, the atom type is assigned the mass of a single atom, e.g., 15.9994 au for all atom types of oxygen, 40.08 au for all atom types of calcium, and so forth. If multiple particles are needed to represent a single atom, such as Pt with a virtual electron or aromatic carbon with two virtual π electrons, the mass of the elemental atom is split between the atom type for the core atom and the atom types for virtual electrons. For example, the Pt core has a mass of 194.08 au and the virtual electron has a mass of 1.00 au, together yielding the mass of a Pt atom of 195.08 au[27] Similarly, the aromatic carbon atom has a total mass of 12.011 au, which can be split into 10.011 au for the core and 1.00 au for each of the two virtual π electrons.[28] The specifics of mass assignments of atom types in

https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282−3297

multicomponent atoms follow the declarations in a given force field.

2.3. Topology Building. The topology specifies the covalent bonds between atoms, and building the topology is guided by definitions in the force field. Usually, structural templates for the covered chemistries are supplied along with the force field parameters that exemplify the intended use of the atom types. The templates may include sample structures of molecules or functional groups using specific atom types (e.g., water and sulfate ion), sample structures of molecular crystals (e.g., a unit cell of polyethylene or muscovite mica), or fragments of polymers (e.g., a nucleotide in RNA). Structural elements, such as molecules, solvents, electrolytes, and nanostructures, can be combined with each other and scaled up to different shapes and sizes. Also, fragments or functional groups may be linked with each other to form new bonded molecules using the same atom types, sometimes requiring additional bonded (and nonbonded) parameters for new linkages by analogy.

Some matching between a given structure and the available force field atom types is usually required. Scripts and programs are available to automate topology and model building, including automated templates in Materials Studio, TopoTools in VMD, model builders in CHARMM-GUI, psfgen in VMD, msi2namd, iff2gro, msi2lmp, and MoSDeF.[32−39] Use of the topology templates requires bond distances to be recognized and assigned correctly. Once the bonds are established, angles and torsions are automatically defined as 1−2−3 and 1−2−3−4 interactions between continuously bonded atoms, as well as any further bonded terms such as impropers and cross terms as applicable. The necessary parameters to describe each of these bonded interactions are then looked at in the force field. Newer frameworks like CHARMM-GUI are very reliable in topology construction; however, it is recommended to double check structures for possible errors in bonding since even a single missing or misplaced bond in a large system can produce incorrect energies and dynamics. In 3D periodic boxes with continuously bonded structures, such as minerals or long polymers, it is critical that bonds extend correctly across the box boundaries.

Symmetry, wildcards, and atom type equivalences are often used to reduce the number of repetitive entries in a force field and require attention in topology building. Equivalences state that a subset of the interactions defined for an atom type are identical to another atom type and therefore do not need to be specified independently. For example, in PCFF, the atom type `c=1` ("non aromatic, next to end doubly bonded carbon") has unique bond and torsion coefficients, but its nonbonded, angle, and improper coefficients are defined to be equivalent to atom type `c=` ("Non-aromatic end doubly bonded carbon"). This indicates that if bespoke coefficients for nonbonded, angle, or improper interactions involving atom type `c=1` are not available, coefficients for `c=` may be substituted. Multiple possible torsions around the same central bond may be assigned the same torsion potential using a wildcard * (representing all atom types) when the outer bonded atoms are similar (* type1 type2 *), and the sequence may be read in either direction. The use of such atom equivalences, wildcards, and symmetry does not affect accuracy and simplifies working with force fields.

Some programs, such as Materials Studio, also provide a two-tiered assignment of topology. The first tier is intended to be of highest accuracy; if dedicated bonded parameters (or even nonbonded parameters) cannot be found for the constructed topology, a message is issued to the user, and an approximate set of "automated" parameters is assigned to missing terms to enable a simulation. In the second tier, automated assignments use default parameters and atom equivalences for a large class of chemistries, which often lead to reduced accuracy when many second tier atom types and parameters are used.

2.4. Assigning Charges. Atomic charges represent the distribution of valence electrons in atomic configurations,[40] and they are often prescribed by the force field in an explicit way in templated structures or structure model databases. Atomic charges are typically assigned (1) using charge increments depending on bonded nearest neighbors or (2) a database of fragments that specifies the atom types along with atomic charges and possible different charges for the same atom types depending on chemical modifications. Charge increments consider the bonded neighbors of a given atom and apply a tabulated charge increment for each bond count with a neighboring atom. The addition of all bond increments for a given atom yields atomic charge and maintains an overall charge-neutral structure. For example, consider $CHCl_3$ with C−H bond increments of −0.03e for C and +0.03e for H, and C−Cl bond increments of +0.1e for C and −0.1e for Cl. The application of charge increments yields charges of +0.27e, +0.03e, and −0.3e for C, H, and Cl, respectively. Often, however, functional groups can be ionized because of acid−base reactions; e.g., phosphoric acid can be present as hydrogen phosphate ionic species with nonbonded cations depending on the pH value in solution, and then bond increments no longer suffice to assign atomic charges. Overall, a database approach can be more useful than charge increments since nonbonded interactions with cations or anions are difficult to include in charge increments, and their charge is often fractional.

Importantly, atomic charges are not determined by the atom type; i.e., atoms of the same type can assume slightly different charges depending on the chemical environment. (In contrast to atomic charges, LJ parameters are determined by and tied to a given atom type.) For example, carbon atoms in hydrocarbons can have different negative charges depending on the number of hydrogen atoms attached, or silicon atoms in ring and chain silicates can have different charges depending on the degree of polymerization and ionization of silanol groups (Si−OH versus Si−O− Na+ groups) while maintaining the same atom type. If atomic charges vary by a lot, e.g., more than 0.1e or 20%, usually different atom types with different LJ parameters and bonded parameters are needed. However, sometimes the simplicity of using fewer atom types is sufficient, especially if the atom types have undergone rigorous validation for the original chemistry, and it is then not necessary to change atom types and LJ parameters. Also, it can be detrimental to the accuracy of a simulation to import atomic charges from automated charge equilibration schemes, such as QEq, or from DFT calculations, which often have large uncertainties.[20,40]

2.5. Special Considerations for Simulations. Extra considerations for simulations with bonded force fields include choosing the additional parameters used for computing LJ interactions and electrostatic interactions, atom constraints, and reactive protocols for bond breaking and bond formation. Spherical cutoffs are usually employed to limit the range of LJ interactions, for example, 8 Å for AMBER, 12 Å for IFF, or no

https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282−3297

cutoff using LJ-PME in CHARMM.[41] Cutoffs greater than 10 Å provide for a more accurate representation of compounds across the periodic table, for example, heavy metal or halide ions with nonbonded diameters larger than 5 Å, and improve the transferability of force field parameters.[41] Electrostatic interactions in three-dimensional periodic simulation systems are calculated using Ewald, PPPM, or PME methods[42−44] to include long-range contributions. For the PPPM and PME methods, the electrostatic energy is the sum of a short-range contribution and a *k*-space contribution.

Computation time can be reduced by fixing the relative positions of hydrogen atoms, related bonds, and bond angles using the SHAKE algorithm, as time steps can be increased from a typical value of 1.0 to 1.5 fs, or sometimes to 2 fs without sacrificing stability and convergence of MD simulations. The largest reasonable time step is determined by mobile particles of the lowest mass, and methods to constrain particles of lower mass into "super-particles" of higher mass allows the use of a larger time step and longer simulation times overall. Sometimes, the mobility of entire parts of a model structure can be constrained to reduce the wall time of simulations, such as when a structural segment is located far from the active site of interest and is not interfering with its dynamics.

A recurring challenge in molecular simulations using bonded force fields is modeling chemical reactions. Bond breaking can be incorporated by the replacement of harmonic bonds by shifted Morse bonds with tuned bond dissociation energies as shown in IFF-R, which allows accurate simulations of stress−strain curves up to failure.[45] Complex reaction mechanisms involving multiple bonds that may be broken or formed can be simulated using the REACTER[8,9] tool, discussed further in Section 3.3, which is designed to handle covalent bond rearrangements typical of reactions in organic and polymer chemistry.

3. LAMMPS TYPE LABEL FRAMEWORK

3.1. Description of the Framework. Type labels, illustrated in Figure 3, are alphanumeric strings assigned to atoms, bonds, angles, and so forth, to supplement the numeric types that are traditionally used in LAMMPS. This convention is more common for MD software packages that cater to biological force fields, such as GROMACS. The introduction of type labels into LAMMPS helps to streamline common tasks

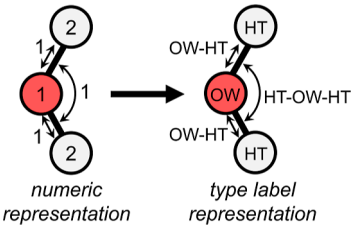


Figure 3. Improved type labels representation of bonded interactions is demonstrated using a water molecule. Type labels offer a more descriptive encoding of atom types, bond types, angle types, and so forth, in molecules. Previously, LAMMPS assigned arbitrary numbers to each interaction type, preventing any sort of transferable information about the chemical identity of each atom, bond, or higher-order interaction. With type labels, chemical elements can be inferred, and similarly the constituent atom types of a particular bond type can be inferred for a given force field.

and workflows and opens many possibilities for new features. For example, atom types can be assigned labels that match the conventions of a given force field, making them human readable and compatible with tools that follow the same conventions. The use of type labels enables force fields to be archived separately from input files with parameters defined in terms of atoms types (as described in Section 4 for OpenKIM), and for input files to be more universal by using labels rather than assigning new arbitrary numbers to interaction types for each simulation. The increased portability of topology files greatly simplifies setting up simulations that require multiple data files or molecule template files. Workflows involving LAMMPS features that require creation of multiple files or manipulation of interaction types, such as the REACTER protocol (described in Section 3.3), are considerably streamlined by using type labels. Similarly, type labels allow topology files to be applied without modification to different systems that use the same force field.

An example workflow using type labels is shown in Figure 4. OpenKIM archives bonded force field parameters in terms of atom types including the type definitions and parameters for bond, angle, dihedral, and improper interactions as well as nonbonded interaction parameters. A LAMMPS script can access this information by specifying the OpenKIM identifier for the selected bonded force field ("KIM ID") using the "kim init" command. OpenKIM functionality implemented within LAMMPS then issues the necessary interaction commands for a molecular system defined in the LAMMPS script or associated data files that use the type label formalism. The use of type labels makes these files more portable in that they can be used with different force fields with the same atom type conventions and enables other type-label-aware tools to function.

The primary input files that benefit from the type label framework are data files, molecule templates, and input scripts. Data files contain the initial configuration of a molecular system, such as atomic coordinates, atom types, and lists of bonded interactions. Traditionally, interaction types are numeric values from 1 to *n*, where *n* is the number of types. The type label framework introduces a new section into LAMMPS data files that assigns strings to each numeric type. Subsequent sections within a data file that reference a numeric type are first mapped onto the corresponding type label defined in that file. For example, if two data files are read into the same simulation, then the types no longer have to be in the same numerical order as long as they use the same type labels. To increase the human readability of the files, numeric types can also be directly replaced with type labels throughout the data file. Molecule templates are another common type of input that typically contains information about a small molecule or multiple molecules that can be referenced during the simulation. For example, the REACTER protocol, implemented in LAMMPS as "fix bond/react", requires these files to define the topology of the reaction site before and after the reaction occurs. When molecule templates are utilized within the type label framework, all interaction types are replaced directly by type labels within molecule template files, making them universal and human-readable. Finally, interaction types can be defined or used as arguments for commands in the input script, which define the settings and output of the simulation. Commonly used commands that involve atom types, such as "mass", "pair_coeff", "bond_coeff", and so forth, have been modified to accept type labels.

https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282−3297

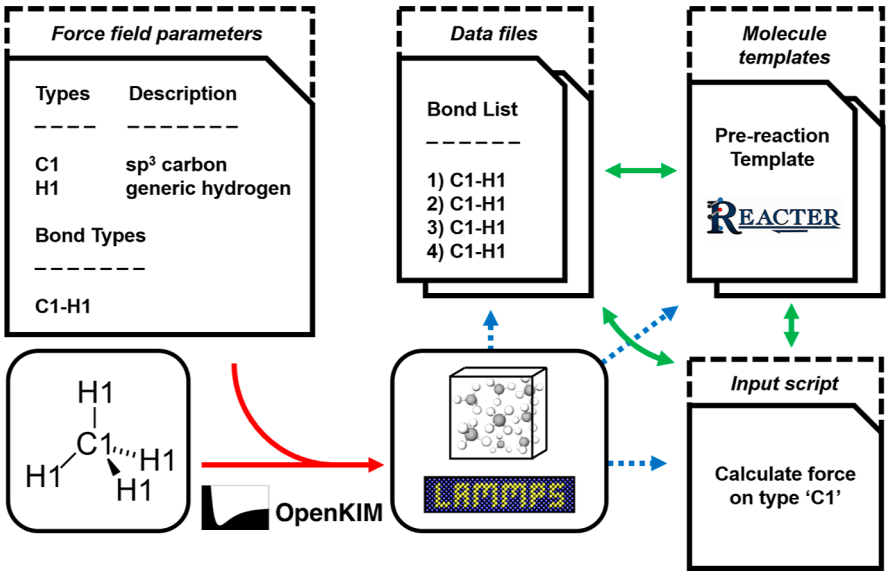


Figure 4. A general workflow to set up MD simulations that utilizes the type label framework in LAMMPS. The initial configuration, represented here by a methane molecule, is endowed with type labels and interaction parameters for a given interatomic potential using OpenKIM. When type labels are used within all inputs files, such as data files, molecule templates, and the input script, then these files are inherently compatible with each other and portable to simulations that use the same force field conventions.

Additionally, a utility function called "label2type" has been added to convert type labels to numeric types for use with commands that do not yet support type labels. These new features together enable input scripts and files to be compatible without modification with other files that use the same set of type label conventions.

The type label functionality was designed to be a general framework that does not mandate using a specific label format. However, as a tool meant to increase the portability of files, the full potential of type labels will only be realized if default formats can be established, especially within individual modeling communities. In particular, it is helpful if higher-order interactions, such as bonds, angles, and so on, encode the constituent atom types within the type label. This codification is especially useful for many force fields in the materials science community for which bonded interactions can be derived from atom types. OpenKIM, REACTER, and TopoTools and associated efforts making use of LAMMPS type labels have adopted the hyphen (`-`) as the delimiter separating atom types that comprise multiatom bonded interactions. The hyphenated format was selected for its readability and consistency with notation used in other codes like CHARMM and VMD. For example, a bond between a carbon of type `C1` and a hydrogen of type `H1` would be assigned the type label `C1-H1`. The designation `C1-H1` improves on the arbitrary numeric bond type available until now in LAMMPS, which is not machine translatable to the constituent atom types.

The use of a hyphen as a delimiter precludes atom types that have hyphens in the middle of their names, but such atom types do not appear in common force fields. The more common case of a hyphen (or minus sign) at the end of an atom type label is able to be parsed unambiguously; thus, `A--B` would be correctly identified as a bond connecting atom type `A-` with atom type `B`. The only other restrictions currently placed on atom types is that they cannot start with a number, a number sign (#) or an asterisk (*). Although the developers encourage the use of a standard format whenever possible, the type label framework is meant to support any task that is simplified by assigning custom alphanumeric strings to interaction types.

A possible source of ambiguity when defining higher-order interactions is the order in which the constituent atoms are listed. To determine if the two types of labels are equivalent, the symmetry of the interaction must be considered. For example, a bond type of `C1-H1` has the same interaction parameters as a bond of type `H1-C1`. Generally, angle and dihedral interactions also have a simple palindromic symmetry. However, improper dihedrals are more complex four-body interactions often used by force fields to describe out-of-plane interactions. Impropers have a variety of formulations and conventions, and their symmetries must be encoded individually. The symmetry of each improper type supported by LAMMPS has been encoded in the source code in support of the type label framework. As discussed in Section 3.2, consideration of interaction symmetries avoids requiring a canonical ordering of atom types. Further opportunities to improve the functionality of type labels are discussed in detail in the relevant sections below.

3.2. Usage of OpenKIM Bonded Force Fields in LAMMPS. A simulation in LAMMPS is managed through an input script (text file) composed of a series of lines, each beginning with a command name followed by one or more arguments. This programming-like mechanism can define variables, execute LAMMPS functionality, perform conditional tests, initiate loops, invoke shell commands, and so on.[46] LAMMPS also includes a series of `kim` commands, providing a wrapper to the OpenKIM repository of interatomic potentials and bonded force fields.[6,47]

The `kim` commands in LAMMPS allow users to use interatomic models (IMs) (interatomic potentials and force fields) and their predictions for various physical properties archived in the OpenKIM repository. IMs stored in OpenKIM can be of two types: (1) Portable Models (PMs) that include the software defining the IM and can be used with *any* simulation package compatible with the KIM standard; and (2) Simulator Models (SMs) designed to work with a single

https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282−3297

package, like LAMMPS, and contain the parameter files, input commands, and metadata (supported species, units, and so on) to run an IM implemented within that package.

Currently, bonded force fields in LAMMPS are supported through the KIM SM mechanism, which also enables the use of long-range electrostatic interactions unsupported by PMs. There are many benefits to using bonded force fields archived in OpenKIM instead of defining parameters in data files and input scripts, as traditionally done in LAMMPS simulations. Briefly, these include *reliability* that the force fields have been vetted and are correct and have been subjected to tests to help users select a suitable model for a given application, *reproducibility* in that each IM in OpenKIM is provided with a DOI that can and should be cited in publications to allow others to replicate the work, and *accessibility* making it easy to obtain and use an IM, query its predictions for a host of properties (from within LAMMPS input scripts), and obtain citation information through LAMMPS. For more information, see the "`kim commands`" documentation in LAMMPS[47] and the OpenKIM documentation.[7]

Prior to the type label framework introduced here, archiving bonded force fields as KIM SMs involved hard-wiring LAMMPS numeric atom types to specific force field atom types. For example, in the previously archived IFF model,[20,48−51] aluminum of type `ay1` is set to LAMMPS atom type 1, oxygen of type `oy7` is set to atom type 5, and so on. The bond between these atom types (aluminum and oxygen) is set to LAMMPS bond type 1. This mapping could not be changed afterward. Thus, to use this force field, users had to adapt their input scripts and associated files to conform to this numbering, making using bonded force field SMs impractical in many scenarios. For example, it would be difficult for users to use advanced input generation mechanisms like CHARMM-GUI[34] to build complex systems and prepare inputs. One would need to go through and modify the numerical mapping for the input generator by hand. The process was not only error-prone but also complicated and cumbersome, and there was no mechanism to validate the modifications and changes to confirm their correctness.

Here, we address this challenge by integrating the functionality of type labels in LAMMPS into KIM SMs. The type label framework facilitates the definition of label mappings for all atom types, bond types, dihedral types, and so forth, to numeric values used internally in LAMMPS. Under the new scheme for archiving bonded SMs, atom and interaction types are associated with string type labels instead of numeric types, with interactions following the hyphenated convention described in Section 3.1. For example, in the new version of the IFF model,[52] the user only needs to assign the respective aluminum and oxygen labels as `ay1` and `oy7` and assign the bond between them with the label `ay1-oy7` (or `oy7-ay1`, as discussed below), and so on for other interactions.

Two required subcommands are employed when using OpenKIM IMs in LAMMPS, one to select the IM by specifying its KIM ID and perform necessary initialization (`kim init`), and the second to set up the IM for use by executing any necessary LAMMPS commands (`kim interactions`). The `kim init` command sets the IM and additional prerequisites in the LAMMPS default settings for using the selected model. If the IM selected by `kim init` is an SM, then any commands that must be issued before creating atoms and bonding topology are invoked internally by the LAMMPS KIM package during the execution of `kim init`. This includes, for example, the `atom_style` command, which determines whether bonded interactions or charges are allowed.

After `kim init`, the user must create the simulation box, the atoms (or molecules) within it, and the bonded interactions between those. This can be accomplished by reading a data file using `read_data` or directly in the input script with `create_box`, `create_atoms`, `create_bonds`, and so forth, commands. This process is the same as for a non-KIM simulation except for two caveats: (1) if using a KIM bonded SM, type labels must be defined in the input script using the `labelmap` command, or in the data file, and (2) interaction coefficients and masses do not need to be specified when using a bonded force field SM since all necessary data are provided in the SM.

After the simulation box, particles and bonding topology are defined, the `kim interactions` command must be issued. The `kim interactions` command performs all of the necessary steps to set up the OpenKIM IM specified in the `kim init` command. For reactive interatomic potentials, `kim interactions` are followed by a list of chemical element symbols for mapping LAMMPS numeric atom types to chemical elements. For bonded SMs, this mapping is provided by the previously defined type labels, and the syntax to be used is `kim interactions fixed_types`. At this point, the LAMMPS type label framework allows for a straightforward, LAMMPS-native way of assigning masses, interaction coefficients, and (for some SMs) charges[c] to any defined atom and interaction types. These assignments rely on LAMMPS support for type labels in the `mass`, `set` and various `coeff` commands, as well as conditionals enabled by the built-in `is_typelabel` function, which identifies whether a string is a valid type label. The `kim interactions` command overwrites any coefficients provided in the data file, set using the `read_data` command, or in the input script and replaces them with the SM coefficients.

Symmetry permutations for bonded interactions are considered to prevent mistakes. The SM metadata contains all symmetry permutations allowed by the developer of the force fields; e.g., both `ay1-oy7` and `oy7-ay1` will be included with the same bond coefficients. Although this takes the burden of symmetry consideration off the user and allows for the aforementioned elegant LAMMPS-native implementation, it shifts the burden to the creator of the OpenKIM SM (often a researcher external to OpenKIM). Because of this, future plans include the incorporation of automatic support for symmetry permutations of bonded interactions within the LAMMPS KIM package.

In summary, since interaction coefficients for a bonded force field are specified within the OpenKIM SM, input or data files for a simulation using such the model specify only the geometry, bonding topology, and charges (if needed). This simplifies inputs and removes possible errors in the coefficients. Further, when the parameters of a force field are modified (or even when switching to a different force field that shares atom types), every simulation designed for the old force field can be reused by only modifying the KIM ID of the OpenKIM model in the input file.

3.3. REACTER Framework. The REACTER protocol is a method for modeling chemical reactions in classical MD simulations.[8,9] It has been used to model the polymerization of thermoplastics, thermosets, and composite materials,[53] and for applications ranging from solid electrolyte interphases for

https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282−3297

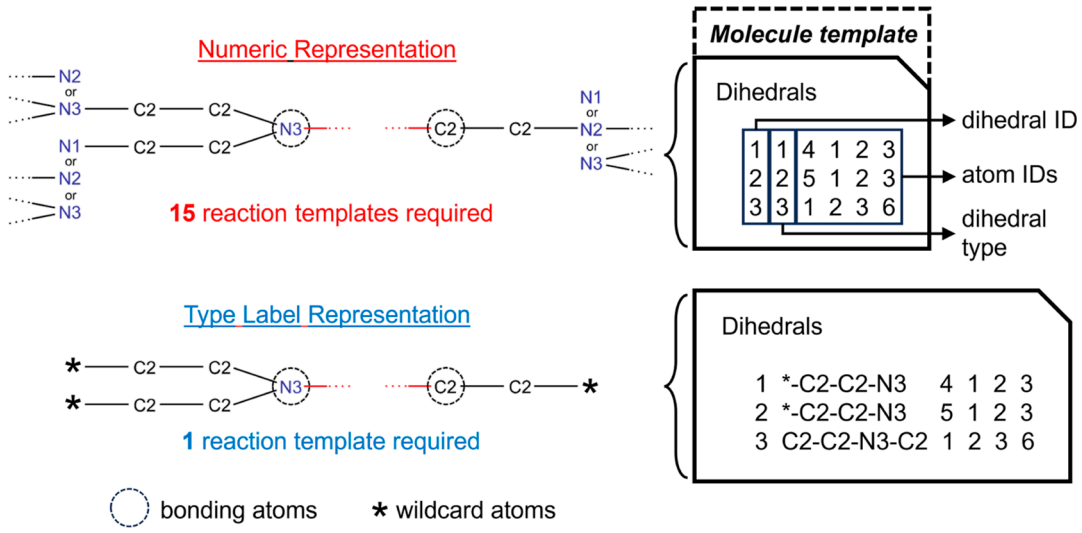


Figure 5. The REACTER protocol previously required 15 reaction templates to model a polymerization reaction to create a model of polyethylenimine, due to explicit definition of dihedral interactions.[56] Using wildcards and autotyping, the number of required reaction templates is reduced to one. Image adapted from the cited reference with permission from the publishers.

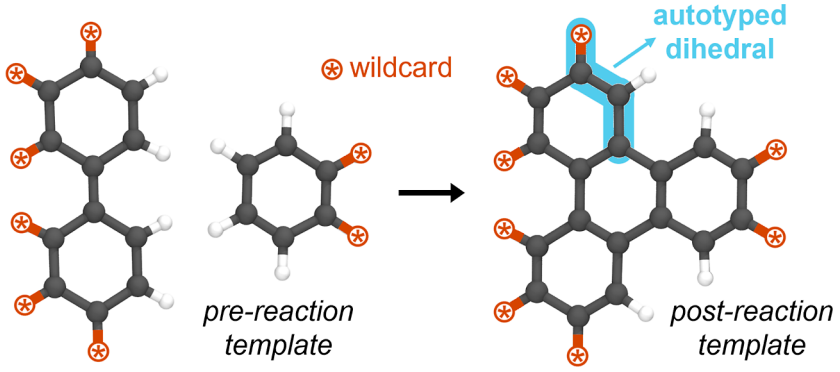


Figure 6. An example of a Diels–Alder type reaction mechanism that can be modeled by the REACTER protocol using type labels. Gray atoms represent carbon (C), white atoms represent hydrogen (H), and wildcard atoms (★) could be C, H, or any other element according to the local simulation environment. Using automatic typing via type labels, just one pair of molecule templates can define this reaction, in contrast to 2^8 templates when atom types must be listed explicitly. Angles, dihedrals, and impropers that involve wildcard atoms are typed automatically. One such autotyped dihedral interaction is highlighted in blue.

lithium-ion batteries[54] to materials for efficient CO_2 capture.[55] REACTER is a template-based method that requires as input two molecular templates: one for the reactants and one for the products. The most time-consuming step when using REACTER is the creation of these reaction templates, which can contain atom types, all higher-order bonded interactions and their types, as well as partial charges and other per-atom information. Prior to the introduction of type labels, ensuring that identical atoms use the same numeric types between the data file and all reaction templates was a tedious and error-prone process. Using the type label framework, this issue is bypassed entirely because interaction types can be standardized and are automatically synced when reading multiple files. Similarly, molecule templates can be used without modification for different simulations that use the same force field convention. This improvement in the universality of files used by REACTER can significantly simplify the workflow to create new reaction templates and enable reuse of existing templates.

Type labels also enable new functionality within the REACTER protocol such as the automatic assignment of types for new bonded interactions created during reactions. As discussed in Section 3.1, type labels allow the constitutive atom types to be encoded in higher-order bonded interactions, so that the corresponding types can be inferred. In many cases, an automatic type assignment substantially decreases the number of reaction templates required to describe a given class of reactions. Prior to type labels, every interaction type had to be included in the post-reaction template to ensure generality for all force fields, which have a variety of conventions for dihedral improper interactions. This limitation is especially problematic for four-body dihedral interactions that involve atoms three bonds removed from reactive sites. For example, Kawagoe et al.[56] defined reaction templates for the polymerization of polyethylenimine in which an atom in the templates, the one farthest from the reaction site, could potentially be one of three different atom types. This minor difference between reactive sites in the simulation led to the requirement of 15 distinct templates to describe one of the polymerization reactions. Using type labels and automatic type assignment, the number of required templates is reduced to one, as illustrated in Figure 5. To demonstrate the power of type labels to generalize reaction templates using a more extreme example, consider a Diels–Alder-style reaction as shown in Figure 6, where atoms

https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282–3297

at the "edge" of the template can be either carbon or hydrogen. Attempting to model all possibilities would result in 2^8 templates according to a combinatorial calculation, whereas just one template is needed if atoms near the template edges are marked as wildcards and new dihedral interactions are inferred and typed automatically. Note that atoms are typically labeled using the atom type of a particular force field, rather than the chemical element, in simulations that use bonded force fields.

3.4. Support in Visualization Software. Alongside MD simulation software, tools for visualization and analysis of MD trajectory data play crucial roles in molecular simulations. Visual inspection of MD trajectories is usually the first step in assessing the progress and success of a simulation. Thus, the corresponding tools have to be compatible and kept up-to-date with the current file formats in use. In addition, some software packages include the ability to construct or manipulate geometry and topology data and thus can be used as part of the preparation pipeline for MD simulation inputs. In the following, we discuss support for type labels in two software packages that are popular with LAMMPS users: VMD[57,58] and OVITO.[59]

3.4.1. VMD. VMD[57] is a molecular visualization program for displaying, animating, and analyzing large biomolecular systems using three-dimensional graphics and built-in scripting. VMD was originally conceived as a companion program to the NAMD[60] MD simulation software, which itself was conceived to be primarily used in combination with the CHARMM force field.

In this context, atom geometry data is typically imported from Protein Data Bank (PDB) format files, where atoms are identified by their name (which includes explicit definition of the chemical element) and the name of the residue to which they belong. Since the name of the residue is standardized for amino acids and DNA bases and the atom name is an indication of the location within the residue, it is straightforward to construct the force field topology data—i.e., to assign atom types, bonds, angles, dihedrals, and more—by looking up the assignments in a residue database provided by the force field. The resulting system-specific information is then stored in a protein structure file (PSF), where the core information is the atom type, per-atom charge, and the connectivity. When not using an OpenKIM force field, the force field parameters (except for charges, which are residue specific, not atom-type specific) are then looked up from a parameter file, where the parameters for bonded interactions are inferred from the constituent atom types. For dihedrals, there also may be wildcard atom types (`X`) that would match *any* atom type.

Prior to type labels, LAMMPS used solely numeric atom, bond, angle, dihedral, and improper types starting from 1. Therefore, when importing LAMMPS simulation data into VMD for further processing, in the absence of atom type information in the data file, it is beneficial to create a dummy PSF format file with CHARMM compatible atom types and atom names. With this, built-in heuristics in VMD recognize specific elements and atom types for advanced analysis and visualization (e.g., showing and counting hydrogen bonds or the secondary structure of proteins). A PSF file will also provide the explicit bonding patterns used by LAMMPS. Without explicit topology and type or element information, VMD will use heuristics that will fail when the code cannot recognize the elements correctly (e.g., with numeric atom types) and will only derive information needed for visualization (e.g., identification of bonds without assigning bond types). The TopoTools plugin has additional utility functions that can be used to restore information, e.g., to derive element names from atomic masses or covalent radii from element names.

For reading topology and trajectory files, VMD uses a standard application programming interface (API) called MolFile and a plugin system to support different file formats. Unfortunately, this API has significant limitations when it comes to supporting the LAMMPS data file format, which requires the knowledge of the atom style in order to correctly interpret the `Atoms` section with position and other per-atom data. All the per-atom and topology information stored in VMD, however, is also accessible through the Tcl script interface, where it can be read *and* set. This led to the development of the VMD TopoTools plugin which allows flexible input, output, and manipulation of geometry and topology data. It contains topology manipulation methods that can be utilized to either create or manipulate topology data in a more abstract and convenient way compared to the low level basic data access functionality in the Tcl script interface. This abstract functionality can then also be used in combination with other VMD Tcl script commands to implement high-level functionality to export or import topology and geometry data in custom formats. For example, the TopoGromacs[61] functionality enables the conversion of PDB + PSF data for CHARMM force field simulations with NAMD into files suitable for use with GROMACS.

As of version 1.9, the TopoTools plugin[33] has been extended to support type labels and thus has expanded functionality for reading and writing LAMMPS data files. Since TopoTools is written entirely in Tcl and because the required Tcl script commands have not changed in recent releases of VMD, it is not necessary to use newer versions of VMD to utilize the new functionality; the updated TopoTools plugin can be installed into older versions of VMD without a problem. Compatibility with type labels is achieved through an additional keyword to the `topo writelammpsdata` command. In order for the type label information in LAMMPS data files to be read correctly by the `topo readlammpsdata` command, the files must contain a `LABELMAP` metadata tag in the first line of the file, that is, the "title" line, which is otherwise ignored. This is similar to the `CGCMM` tag, which indicates a data file with numeric types, but with additional alphanumeric type and style information added for use with LAMMPS, VMD, and other visualization and analysis tools as comments.

An example for how to generate LAMMPS data files with the VMD TopoTools plugin version 1.9 or later, with and without type label information, is provided in the Supporting Information.

3.4.2. OVITO. OVITO is a simulation postprocessing and visualization software for particle-based models. It is useable in two ways: as a desktop application with an interactive graphical user interface and as a Python package that may be incorporated into other programs or noninteractive workflows. OVITO is designed to work particularly well with the LAMMPS package and can handle some of the peculiarities of this simulation code, e.g., unlabeled atom types, varying particle counts, complex particle shapes, and arbitrary per-atom or bond attributes that LAMMPS may dump onto the disk upon user request.

When importing a data file from LAMMPS, the potential or force field used by the simulation is irrelevant for a

https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282−3297

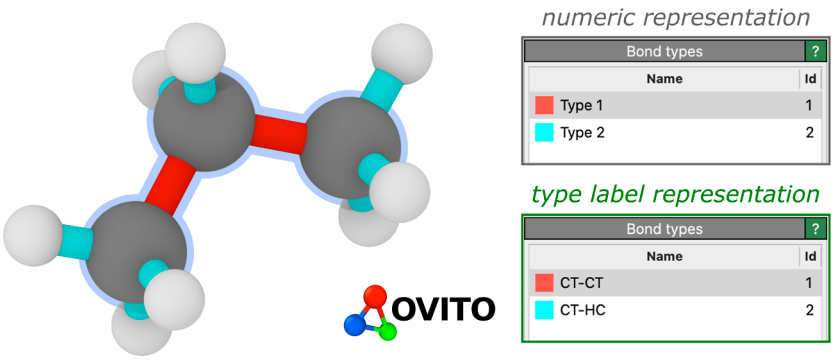


Figure 7. Type labels support for bonded interactions in OVITO. Before, with numeric labels only, it was difficult for the user to see the physical meaning of each interaction.

visualization tool like OVITO. However, for the visual representation of the particles or atoms, it is useful that the software determines their chemical types in order to preselect meaningful colors and radii according to common conventions. So far, this had to be done using various hints that OVITO looks for in a data file, e.g., nonstandard annotations added by the user (or the writing software) or the masses of the defined atom types, which are matched against a lookup table of chemical elements. If the automatic identification of atom types is not possible based on such hints, the user ultimately has to manually assign correct names to the numeric types within OVITO.

With the introduction of type labels in the LAMMPS data file format, the workflow is considerably simplified. Labels can be designed to explicitly indicate the chemical identity or the role of a particle in a force field, as shown in Figure 7, thereby offering a direct way to select meaningful and reproducible visualization parameters. The capability to read and write the extended data format was introduced in OVITO version 3.7 and covers particle types as well as bond, angle, dihedral, and improper types.

4. OPENKIM SUPPORT FOR BONDED FORCE FIELDS

Usage of SMs to deploy OpenKIM bonded force fields in LAMMPS is already described in Section 3.2. Here, we detail the other aspects of the OpenKIM cyberinfrastructure as they pertain to the new type labels support for bonded force fields.

At the time of publication, openkim.org hosts nearly 650 IMs, which can be automatically downloaded, built, and installed for use with compatible codes by the "`kim-api-collections-management`" utility installed with KIM. Every model on openkim.org is run through compatible verification checks (VCs) and tests, which (respectively) check the model's coding integrity and predictions for material properties. Every model is accompanied by thorough citations, provenance, and version control.

In a sense, the OpenKIM cyberinfrastructure comprises a repository, a package manager, and very extensive continuous integration for IMs. The advantages of using a curated and tested model archived in OpenKIM have already been present for reactive interatomic potentials and are now available to users of bonded force fields.

4.1. Archiving of Bonded Force Fields in OpenKIM.org. As mentioned in Section 3.2, the PCFF-IFF force field that was previously archived in OpenKIM with preset numerical atom and interaction types has been upgraded to the new type labels SM scheme.[20,48−52] Additionally, the subset of CHARMM-IFF available in CHARMM-GUI,[20,34,41,50,62−67] which has different functional forms from the PCFF version and supports a slightly different set of materials, has been archived in OpenKIM. Adiabatic core−shell models for NaCl,[68,69] MgO,[70] and CaF[71] have been archived as well. These models allow polarization of ions by representing them as a pair of particles—a core and a shell—connected by a harmonic bond with an equilibrium length of zero. Unlike massless shell models, the shell has a small, but finite mass. These models, therefore, take the form of a simple bonded force field.

Because of the diverse nature of bonded force fields and their application, it is imperative that the user read the description of the model on openkim.org to understand which interactions, parameters and fixes are included in the model, and which must be specified by the user. For example, for the core−shell models, it is up to the user to assign the atoms to core and shell groups and to issue the required commands that use those groups in the simulation.

LAMMPS bonded SMs archived in OpenKIM contain a dictionary metadata field `atom-type-labels` that designates what atom type labels the model is compatible with and what chemical species each atom type label represents. This is an additional ease-of-use feature provided by OpenKIM for bonded force fields.

4.2. OpenKIM Testing of Bonded Force Fields. Prior to this work, all OpenKIM VCs and tests were written to support "Standard Models", which, briefly, means that the information passed to the model consists of only atomic positions and chemical species. Here, we introduce the concept of "Special Purpose" VCs and tests, which are understood to run with models that accept additional information beyond atomic positions and chemical species (in this case, atom type labels and bonding topology). An up-to-date detailed description of the schema defining these concepts is found on the OpenKIM Web site.[72] Special Purpose VCs and tests work with a specific subset of KIM models, in this case, meaning ones that share a common set of atom types.

The new developments in archiving and testing bonded force fields in OpenKIM are timely, as we recently published the first component of the new "Crystal Genome" testing framework. Until recently, OpenKIM tests supported material property computation only for homonuclear solids. Using the crystal symmetry tools and databases in AFLOW,[73,74] ASE,[75] and spglib,[76] Crystal Genome extends the OpenKIM property testing framework to arbitrary multicomponent bulk crystals. The first set of tests available in this new framework computes

https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282−3297

the equilibrium crystal structure and energy of these crystals at zero temperature and pressure.[73,74,77,78] These are the aforementioned tests for Standard Models, which are automatically created from the AFLOW DFT repository and work with the reactive interatomic potentials in OpenKIM.

In this work, we present the creation and comparison of analogous tests for the core/shell and IFF models described in Section 4.1, using a single OpenKIM test driver that computes the equilibrium crystal structure of any material specified using type labels.[79] As more bonded models are added to OpenKIM, this test driver will be expanded to create tests that support these models.

For simple models with limited bonding, such as core−shell models, the process for creating tests is analogous to that for reactive interatomic potentials and can be fully automated by obtaining crystal structures from the aflow.org database. In order to ensure compatibility, all current and future core−shell models archived in OpenKIM are required to follow a uniform convention for atom type labels. For polarized atoms, the cores are named after the element suffixed with "C", and shells with "S" (e.g., "NaC" and "NaS" for sodium). Unpolarized atoms are simply labeled by their element name.

For more complicated force-fields that involve topology, such as IFF, fully automated creation of special-purpose tests is outside the scope of OpenKIM. However, the process can be made semiautomated using automatic input generator tools such as the CHARMM-GUI Nanomaterial Modeler,[41] which can output LAMMPS input files for the myriad of structures it is able to produce with comments indicating atom types. This makes it straightforward to create input structures for OpenKIM tests that run IFF models with CHARMM-GUI atom types.

The development of the tools to generate core−shell tests or convert CHARMM-GUI outputs to OpenKIM tests was greatly simplified by the type label framework. For example, it was not necessary to consider the numerical labels of the atom and interaction types at all, as the coordinates and topology could be entirely specified using type labels. The type labels simply had to be declared. The test generation scripts are available with the test driver and can be easily used to create new tests for new materials (e.g., when new options are added to the CHARMM-GUI Nanomaterial Modeler) and/or models.

At the time of writing, 79 new tests using the test driver are available on openkim.org. To the authors' knowledge, this represents the first effort to systematically compare material property predictions between different bonded force fields and reactive interatomic potentials as well as experimental and DFT reference data. A sample of comparisons is given in Table 1. In these results, the IFF models reproduce the experimental structure very well, especially considering its low computational cost, performing better than DFT. Some reactive interatomic potentials are very accurate, while others show significant deviation from experimental values. Full results for all the tests can be accessed by viewing the test driver on openkim.org[79] and clicking on one of the tests under "Tests using this Test Driver."

Table 1. Sample of Equilibrium Crystal Structure Results for Bonded Force Fields and Reactive Interatomic Potentials Computed in OpenKIM, as well as Experimental and DFT Reference Data (Full Results Available on openkim.org[79])[a]

crystal structure	rock salt	α-quartz	
material	MgO	SiO_2	
lattice parameter	a	a	c/a
expt (298 K)	4.211[80]	4.913[81]	1.100[81]
expt (0 K)	4.199[80]	4.901[81]	1.101[81]
PAW−PBE (0 K)[78]	4.253	4.992	1.100
CHARMM-GUI IFF (0 K)[20,50,62−67]	4.187	4.916	1.084
PCFF-IFF (0 K)[20,49−52]		4.919	1.084
core−shell (0 K)[69,70]	4.225		
Buckingham (0 K)[82,83]	4.187		
Buckingham (0 K)[84,85]		5.039	1.095
Buckingham (0 K)[86,87]		4.941	1.103
ReaxFF (0 K)[88−90]		4.892	1.104
ReaxFF (0 K)[91,92]		4.898	1.093
Vashishta (0 K)[93,94]		4.959	1.072
Vashishta (0 K)[95,96]		5.046	1.063
Vashishta (0 K)[97,98]		4.829	1.099
Tersoff (0 K)[99−101]		5.082	1.088

[a]Excluding the experimental data, all the values can be directly queried from the OpenKIM repository. The DFT calculations are obtained from the AFLOW-ICSD repository and archived in OpenKIM. Lattice constants are in Angstroms or unitless. All materials are characterized in terms of conventional unit cells. All computational values are obtained by static relaxation, and therefore correspond to 0 K. The cited experimental references provide both a room temperature crystal structure and an extrapolation to 0 K.

5. SUMMARY AND FUTURE WORK

The type label framework introduced in this work allows atom types and higher-order interactions to be represented as strings rather than as numbers within the LAMMPS molecular simulation package. This simple extension increases the versatility with which LAMMPS can interface with the supporting software and manipulate information associated with bonded force fields. For example, the original atom types assigned to each atom, which are typically unique to a given force field, can now be retained throughout each step of the modeling workflow. Useful aspects of this framework include increased portability of input files, direct reference of atom types and higher-order interactions between atoms (bond types, angle types, and so forth) in input scripts, and simulation output files that can be parsed in a standard manner and, therefore, postprocessed more easily. In particular, preprocessing tools that include type assignment and modeling workflows that include multiple input files or high-throughput analysis can benefit from adopting type labels. Type labels are currently available in LAMMPS as a stable release.

To support the new type label functionality, extensions were made to the OpenKIM repository of interatomic potentials and the REACTER package for molecular reactions in bonded force fields. OpenKIM now supports the archiving and testing of bonded force fields through its simulator model (SM) interface. Property predictions of bonded force fields archived in OpenKIM are available within the LAMMPS input files through the OpenKIM Web Query interface. This extends the benefits of using curated, tested, reproducible, and accessible models from OpenKIM, which were formerly only reactive interatomic potentials, to bonded force fields. For REACTER, type labels enable the automatic assignment of atom types to new bonded interactions created during reactions, which can greatly reduce the number of reaction templates required to describe a given mechanism. The increased universality of LAMMPS input files greatly simplifies workflows that involve

https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282−3297

both data files and molecule template files or manipulate interaction types, as required by the REACTER protocol.

The type label framework within LAMMPS remains under development to eventually extend all functionality that references numeric types to also support alphanumeric type labels. In addition, opportunities exist for additional new features that build on the type label framework. For example, the symmetries of bonded interactions could be accounted for when processing type labels. If a bond type label "C1−C2" was previously defined, a subsequent bond with type "C2−C1" could be assigned the label and parameters associated with the previous symmetry-equivalent type label. This operation may be doable natively within LAMMPS, or in a more limited manner, for OpenKIM models only.

Further work within the OpenKIM framework involves the development of new tests (property calculations) for bonded force fields. In particular, tests for arbitrary crystals under the Crystal Genome framework will be developed in parallel for bonded and reactive force fields. Immediate plans include tests for elastic constants, thermal expansion, and surfaces. In the future, tests for more complex properties of interest to the soft matter community, such as hydration energy, binding energy, and catalysis, will be developed. Finally, to enable many realistic usage scenarios, it will be necessary to develop a mechanism for automatically combining multiple bonded force fields archived in OpenKIM in a user-friendly fashion.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jpcb.3c08419.

An example of how to use the TopoTools plugin in VMD with and without type labels (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Jacob R. Gissinger − *Department of Chemical Engineering and Materials Science, Stevens Institute of Technology, Hoboken, New Jersey 07030, United States;* orcid.org/0000-0003-0031-044X; Email: jgissing@stevens.edu

Ellad B. Tadmor − *Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, Minnesota 55455, United States;* orcid.org/0000-0003-3311-6299; Email: tadmor@umn.edu

Authors

Ilia Nikiforov − *Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, Minnesota 55455, United States*

Yaser Afshar − *Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, Minnesota 55455, United States; Intel Corporation, Hillsboro, Oregon 97124, United States*

Brendon Waters − *Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, Minnesota 55455, United States*

Moon-ki Choi − *Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, Minnesota 55455, United States*

Daniel S. Karls − *Department of Aerospace Engineering and Mechanics, University of Minnesota, Minneapolis, Minnesota 55455, United States*

Alexander Stukowski − *OVITO GmbH, Darmstadt 64293, Germany*

Wonpil Im − *Departments of Biological Sciences, Chemistry, Bioengineering, Lehigh University, Bethlehem, Pennsylvania 18015, United States;* orcid.org/0000-0001-5642-6041

Hendrik Heinz − *Department of Chemical and Biological Engineering, University of Colorado at Boulder, Boulder, Colorado 80301, United States;* orcid.org/0000-0002-6776-7404

Axel Kohlmeyer − *Institute for Computational Molecular Science, Temple University, Philadelphia, Pennsylvania 19122, United States;* orcid.org/0000-0001-6204-6475

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jpcb.3c08419

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors thank Ryan Elliott and Ron Miller for their early contributions to this effort and Steve Plimpton for helpful discussion and guidance. J.R.G. thanks Kristopher E. Wise for helpful discussions. Resources supporting this work were provided by the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing (NAS) Division at Ames Research Center. In addition, the authors acknowledge partial support by the National Science Foundation, USA (NSF), under grants no. DMR-1834251, OAC-1931304, OAC-1931587, and OAC-1931343. A.K. acknowledges financial support by Sandia National Laboratories under POs 2149742 and 2407526.

■ ADDITIONAL NOTES

[a]In eq 2 and following, the labels i, j, k, l refer to atom numbers and range from 1 to the number of atoms N. Each atom also has an atom type depending on the molecules under consideration. The force field parameters depend on the atom types of the related atoms, but this is not indicated explicitly. A more precise notation would be, e.g., $r_{0,\mathrm{at}(i)\mathrm{at}(j)}$ instead of $r_{0,ij}$, or $K_{\theta,\mathrm{at}(i)\mathrm{at}(j)\mathrm{at}(k)}$ instead of $K_{\theta,ijk}$, where $\mathrm{at}(i)$ denotes that atom type of atom i. For simplicity, we use the abbreviated notations $r_{0,ij}$ and $K_{\theta,ijk}$ wherein i, j, k, l are indices for respective atoms with the atom type implied.

[b]Recall that i and j refer to atom numbers, but the LJ parameters ϵ and σ depend on the types of these atoms.

[c]Because OpenKIM archives force fields and not structure topologies, charges are only provided for models where they are fixed for a given type, such as simple core−shell models. For any force field with variable atom charges, charge assignment is considered to be part of topology building as described in Section 2.4.

■ REFERENCES

(1) Sprenger, K. G.; Jaeger, V. W.; Pfaendtner, J. The General AMBER Force Field (GAFF) Can Accurately Predict Thermodynamic and Transport Properties of Many Ionic Liquids. *J. Phys. Chem. B* **2015**, *119*, 5882−5895.

(2) Boyd, P. G.; Moosavi, S. M.; Witman, M.; Smit, B. Force-Field Prediction of Materials Properties in Metal-Organic Frameworks. *J. Phys. Chem. Lett.* **2017**, *8*, 357−363.

(3) Ponder, J. W.; Case, D. A. *Advances in Protein Chemistry; Protein Simulations*; Academic Press, 2003; Vol. *66*, p 27.

(4) van Duin, A. C. T.; Dasgupta, S.; Lorant, F.; Goddard, W. A. ReaxFF: A Reactive Force Field for Hydrocarbons. *J. Phys. Chem. A* **2001**, *105*, 9396−9409.
(5) Behler, J. Perspective: Machine learning potentials for atomistic simulations. *J. Chem. Phys.* **2016**, *145*, 170901.
(6) Tadmor, E. B.; Elliott, R. S.; Sethna, J. P.; Miller, R. E.; Becker, C. A. The Potential of Atomistic Simulations and the Knowledgebase of Interatomic Models. *JOM* **2011**, *63*, 17.
(7) Features of the OpenKIM system. https://openkim.org/doc/overview/kim-features (accessed Nov 19, 2023).
(8) Gissinger, J. R.; Jensen, B. D.; Wise, K. E. Modeling chemical reactions in classical molecular dynamics simulations. *Polymer* **2017**, *128*, 211−217.
(9) Gissinger, J. R.; Jensen, B. D.; Wise, K. E. REACTER: A heuristic method for reactive molecular dynamics. *Macromolecules* **2020**, *53*, 9953−9961.
(10) Huang, J.; MacKerell, A. D. CHARMM36 all-atom additive protein force field: Validation based on comparison to NMR data. *J. Comput. Chem.* **2013**, *34*, 2135−2145.
(11) Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and testing of a general AMBER force field. *J. Comput. Chem.* **2004**, *25*, 1157−1174.
(12) Dauber-Osguthorpe, P.; Roberts, V. A.; Osguthorpe, D. J.; Wolff, J.; Genest, M.; Hagler, A. T. Structure and energetics of ligand binding to proteins: Escherichia coli dihydrofolate reductase-trimethoprim, a drug-receptor system. *Proteins: Struct., Funct., Bioinf.* **1988**, *4*, 31−47.
(13) Mayo, S. L.; Olafson, B. D.; Goddard, W. A. DREIDING: a generic force field for molecular simulations. *J. Phys. Chem.* **1990**, *94*, 8897−8909.
(14) Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *J. Am. Chem. Soc.* **1996**, *118*, 11225−11236.
(15) Price, M. L.; Ostrovsky, D.; Jorgensen, W. L. Gas-phase and liquid-state properties of esters, nitriles, and nitro compounds with the OPLS-AA force field. *J. Comput. Chem.* **2001**, *22*, 1340−1352.
(16) Ewig, C. S.; Berry, R.; Dinur, U.; Hill, J.-R.; Hwang, M.-J.; Li, H.; Liang, C.; Maple, J.; Peng, Z.; Stockfisch, T. P.; et al. Derivation of class II force fields. VIII. Derivation of a general quantum mechanical force field for organic compounds. *J. Comput. Chem.* **2001**, *22*, 1782−1800.
(17) Sun, H.; Mumby, S. J.; Maple, J. R.; Hagler, A. T. An ab initio CFF93 all-atom force field for polycarbonates. *J. Am. Chem. Soc.* **1994**, *116*, 2978−2987.
(18) Sun, H. COMPASS: an ab initio force-field optimized for condensed-phase applications overview with details on alkane and benzene compounds. *J. Phys. Chem. B* **1998**, *102*, 7338−7364.
(19) Halgren, T. A. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *J. Comput. Chem.* **1996**, *17*, 490−519.
(20) Heinz, H.; Lin, T.-J.; Kishore Mishra, R.; Emami, F. S. Thermodynamically consistent force fields for the assembly of inorganic, organic, and biological nanostructures: the INTERFACE force field. *Langmuir* **2013**, *29*, 1754−1765.
(21) Tadmor, E. B.; Miller, R. E. *Modeling Materials: Continuum, Atomistic and Multiscale Techniques*; Cambridge University Press: Cambridge, 2011.
(22) Shi, Y.; Xia, Z.; Zhang, J.; Best, R.; Wu, C.; Ponder, J. W.; Ren, P. Polarizable Atomic Multipole-Based AMOEBA Force Field for Proteins. *J. Chem. Theory Comput.* **2013**, *9*, 4046−4063.
(23) Vanommeslaeghe, K.; Hatcher, E.; Acharya, C.; Kundu, S.; Zhong, S.; Shim, J.; Darian, E.; Guvench, O.; Lopes, P.; Vorobyov, I.; et al. CHARMM general force field: A force field for drug-like molecules compatible with the CHARMM all-atom additive biological force fields. *J. Comput. Chem.* **2010**, *31*, 671−690.
(24) Robertson, M. J.; Tirado-Rives, J.; Jorgensen, W. L. Improved peptide and protein torsional energetics with the OPLS-AA force field. *J. Chem. Theory Comput.* **2015**, *11*, 3499−3509.
(25) Heinz, H.; Ramezani-Dakhel, H. Simulations of inorganic−bioorganic interfaces to discover new materials: insights, comparisons to experiment, challenges, and opportunities. *Chem. Soc. Rev.* **2016**, *45*, 412−448.
(26) Geada, I. L.; Ramezani-Dakhel, H.; Jamil, T.; Sulpizi, M.; Heinz, H. Insight into induced charges at metal surfaces and biointerfaces using a polarizable Lennard−Jones potential. *Nat. Commun.* **2018**, *9*, 716.
(27) Wang, S.; Zhu, E.; Huang, Y.; Heinz, H. Direct correlation of oxygen adsorption on platinum-electrolyte interfaces with the activity in the oxygen reduction reaction. *Sci. Adv.* **2021**, *7*, No. eabb1435.
(28) Pramanik, C.; Gissinger, J. R.; Kumar, S.; Heinz, H. Carbon nanotube dispersion in solvents and polymer solutions: mechanisms, assembly, and preferences. *ACS Nano* **2017**, *11*, 12805−12816.
(29) Zhu, C.; Hoff, S. E.; Hémadi, M.; Heinz, H. Accurate and Ultrafast Simulation of Molecular Recognition and Assembly on Metal Surfaces in Four Dimensions. *ACS Nano* **2023**, *17*, 9938−9952.
(30) Dick, B. G.; Overhauser, A. Theory of the dielectric constants of alkali halide crystals. *Phys. Rev.* **1958**, *112*, 90−103.
(31) Lemkul, J. A.; Huang, J.; Roux, B.; MacKerell, A. D. An empirical polarizable force field based on the classical drude oscillator model: development history and recent applications. *Chem. Rev.* **2016**, *116*, 4983−5013.
(32) *Materials Studio 7 Program Suite and User Guide*; BIOVIA, Dassault Systèmes: Cambridge, UK, 2015−2019.
(33) Kohlmeyer, A.; Vermaas, J.; Braun, E. *VMD TopoTools Plugin*: version 1.9, 2022. https://doi.org/10.5281/zenodo.7071898 (accessed Nov 19, 2023).
(34) Jo, S.; Kim, T.; Iyer, V. G.; Im, W. CHARMM-GUI: A web-based graphical user interface for CHARMM. *J. Comput. Chem.* **2008**, *29*, 1859−1865.
(35) *VMD psfgen Plugin*, version 2.0. https://www.ks.uiuc.edu/Research/vmd/plugins/psfgen/ (accessed Nov 19, 2023).
(36) *INTERFACE Force Field*, version 1.5. https://bionanostructures.com/interface-md/ (accessed Nov 19, 2023).
(37) *interfaceff2gro*. https://github.com/kolmank/interfaceff2gro (accessed Nov 19, 2023).
(38) *LAMMPS Tools*. https://docs.lammps.org/Tools.html (accessed Nov 19, 2023).
(39) Cummings, P. T.; MCabe, C.; Iacovella, C. R.; Ledeczi, A.; Jankowski, E.; Jayaraman, A.; Palmer, J. C.; Maginn, E. J.; Glotzer, S. C.; Anderson, J. A.; et al. Open-source molecular modeling software in chemical engineering focusing on the Molecular Simulation Design Framework. *AIChE J.* **2021**, *67*, No. e17206.
(40) Heinz, H.; Suter, U. W. Atomic charges for classical simulations of polar systems. *J. Phys. Chem. B* **2004**, *108*, 18341−18352.
(41) Choi, Y. K.; Kern, N. R.; Kim, S.; Kanhaiya, K.; Afshar, Y.; Jeon, S. H.; Jo, S.; Brooks, B. R.; Lee, J.; Tadmor, E. B.; et al. CHARMM-GUI Nanomaterial Modeler for Modeling and Simulation of Nanomaterial Systems. *J. Chem. Theory Comput.* **2022**, *18*, 479−493.
(42) Ewald, P. P. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Ann. Phys.* **1921**, *369*, 253−287.
(43) Hockney, R. W.; Eastwood, J. W. *Computer Simulation Using Particles*; CRC Press: Boca Raton, 2021.
(44) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A smooth particle mesh Ewald method. *J. Chem. Phys.* **1995**, *103*, 8577−8593.
(45) Winetrout, J. J.; Kanhaiya, K.; Sachdeva, G.; Pandey, R.; Damirchi, B.; van Duin, A.; Odegard, G.; Heinz, H. Implementing Reactivity in Molecular Dynamics Simulations with the Interface Force Field (IFF-R) and Other Harmonic Force Fields. *arXiv* **2021**, arXiv:2107.14418.
(46) Thompson, A. P.; Aktulga, H. M.; Berger, R.; Bolintineanu, D. S.; Brown, W. M.; Crozier, P. S.; in't Veld, P. J.; Kohlmeyer, A.; Moore, S. G.; Nguyen, T. D.; et al. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comput. Phys. Commun.* **2022**, *271*, 108171.
(47) *KIM Command in LAMMPS*. https://docs.lammps.org/kim_commands.html (accessed Nov 19, 2023).

(48) Heinz, H.; Emami, F. S.; Lin, T.-J.; Mishra, R. K. *LAMMPS PCFF bonded force-field combined with IFF non-bonded 9−6 Lennard-Jones potentials for metal interactions v000*; OpenKIM, 2019. https://doi.org/10.25950/9bb85c6e
(49) Hill, J.-R.; Sauer, J. Molecular Mechanics Potential for Silica and Zeolite Catalysts Based on ab Initio Calculations. 2. Aluminosilicates. *J. Phys. Chem.* **1995**, *99*, 9536−9550.
(50) Heinz, H.; Vaia, R. A.; Farmer, B. L.; Naik, R. R. Accurate Simulation of Surfaces and Interfaces of Face-Centered Cubic Metals Using 12−6 and 9−6 Lennard-Jones Potentials. *J. Phys. Chem. C* **2008**, *112*, 17281−17290.
(51) Liu, J.; Tennessen, E.; Miao, J.; Huang, Y.; Rondinelli, J. M.; Heinz, H. Understanding Chemical Bonding in Alloys and the Representation in Atomistic Simulations. *J. Phys. Chem. C* **2018**, *122*, 14996−15009.
(52) Heinz, H.; Emami, F. S.; Lin, T.-J.; Mishra, R. K. *LAMMPS PCFF bonded force-field combined with IFF non-bonded 9−6 Lennard-Jones potentials for metal interactions v001*; OpenKIM, 2023. DOI: 10.25950/2ea4fcb9
(53) Pisani, W. A.; Radue, M. S.; Patil, S. U.; Odegard, G. M. Interfacial modeling of flattened CNT composites with cyanate ester and PEEK polymers. *Composites, Part B* **2021**, *211*, 108672.
(54) Alzate-Vargas, L.; Blau, S. M.; Spotte-Smith, E. W. C.; Allu, S.; Persson, K. A.; Fattebert, J.-L. Insight into SEI Growth in Li-Ion Batteries using Molecular Dynamics and Accelerated Chemical Reactions. *J. Phys. Chem. C* **2021**, *125*, 18588−18596.
(55) Yeganeh, M. S.; Jusufi, A.; Deighton, S. P.; Ide, M. S.; Siskin, M.; Jaishankar, A.; Maldarelli, C.; Bertolini, P.; Natarajan, B.; Vreeland, J. L.; et al. Solid with infused reactive liquid (SWIRL): A novel liquid-based separation approach for effective CO2 capture. *Sci. Adv.* **2022**, *8*, No. eabm0144.
(56) Kawagoe, Y.; Surblys, D.; Matsubara, H.; Kikugawa, G.; Ohara, T. Construction of polydisperse polymer model and investigation of heat conduction: A molecular dynamics study of linear and branched polyethylenimine. *Polymer* **2019**, *180*, 121721.
(57) Humphrey, W.; Dalke, A.; Schulten, K. VMD − Visual Molecular Dynamics. *J. Mol. Graphics* **1996**, *14*, 33−38.
(58) VMD Homepage. https://www.ks.uiuc.edu/Research/vmd (accessed Nov 19, 2023).
(59) Stukowski, A. Visualization and analysis of atomistic simulation data with OVITO-the Open Visualization Tool. *Modell. Simul. Mater. Sci. Eng.* **2010**, *18*, 015012.
(60) Phillips, J. C.; Hardy, D. J.; Maia, J. D. C.; Stone, J. E.; Ribeiro, J. V.; Bernardi, R. C.; Buch, R.; Fiorin, G.; Hénin, J.; Jiang, W.; et al. Scalable molecular dynamics on CPU and GPU architectures with NAMD. *J. Chem. Phys.* **2020**, *153*, 044130.
(61) Vermaas, J. V.; Hardy, D. J.; Stone, J. E.; Tajkhorshid, E.; Kohlmeyer, A. TopoGromacs: Automated Topology Conversion from CHARMM to GROMACS within VMD. *J. Chem. Inf. Model.* **2016**, *56*, 1112−1116.
(62) Heinz, H.; Emami, F. S.; Lin, T.-J.; Mishra, R. K.; Liu, J.; Pramanik, C.; Kanhaiya, K.; Wang, S. *Interface Force Field (IFF) parameters due to Heinz et al. as used in the CHARMM-GUI input generator v000*; OpenKIM, 2023. https://doi.org/10.5281/zenodo.707189810.25950/544c334f
(63) Emami, F. S.; Puddu, V.; Berry, R. J.; Varshney, V.; Patwardhan, S. V.; Perry, C. C.; Heinz, H. Force Field and a Surface Model Database for Silica to Simulate Interfacial Properties in Atomic Resolution. *Chem. Mater.* **2014**, *26*, 2647−2658.
(64) Lin, T.-J.; Heinz, H. Accurate Force Field Parameters and pH Resolved Surface Models for Hydroxyapatite to Understand Structure, Mechanics, Hydration, and Biological Interfaces. *J. Phys. Chem. C* **2016**, *120*, 4975−4992.
(65) Mishra, R. K.; Kanhaiya, K.; Winetrout, J. J.; Flatt, R. J.; Heinz, H. Force field for calcium sulfate minerals to predict structural, hydration, and interfacial properties. *Cem. Concr. Res.* **2021**, *139*, 106262.
(66) Liu, J.; Zeng, J.; Zhu, C.; Miao, J.; Huang, Y.; Heinz, H. Interpretable molecular models for molybdenum disulfide and insight into selective peptide recognition. *Chem. Sci.* **2020**, *11*, 8708−8722.
(67) Kanhaiya, K.; Nathanson, M.; in't Veld, P. J.; Zhu, C.; Nikiforov, I.; Tadmor, E. B.; Choi, Y. K.; Im, W.; Mishra, R. K.; Heinz, H. Accurate Force Fields for Atomistic Simulations of Oxides, Hydroxides, and Organic Hybrid Materials up to the Micrometer Scale. *J. Chem. Theory Comput.* **2023**, *19*, 8293−8322.
(68) Mitchell, P. J.; Fincham, D. *LAMMPS adiabatic core-shell model for the Na-Cl system developed by Mitchell and Fincham (1993) v000*; OpenKIM, 2023. https://doi.org/10.25950/453a6777
(69) Mitchell, P. J.; Fincham, D. Shell model simulations by adiabatic dynamics. *J. Phys.: Condens. Matter* **1993**, *5*, 1031−1038.
(70) Mitchell, P. J.; Fincham, D. *LAMMPS adiabatic core-shell model for the Mg-O system developed by Mitchell and Fincham (1993) v000*; OpenKIM, 2023. https://doi.org/10.25950/9626dfd7
(71) Mitchell, P. J.; Fincham, D. *LAMMPS adiabatic core-shell model for the Ca-F system developed by Mitchell and Fincham (1993) v000*; OpenKIM, 2023. https://doi.org/10.25950/8944bd95
(72) OpenKIM Special-Purpose Test Drivers and Verification Checks. https://openkim.org/doc/schema/special-purpose-tests/ (accessed Nov 19, 2023).
(73) Esters, M.; Oses, C.; Divilov, S.; Eckert, H.; Friedrich, R.; Hicks, D.; Mehl, M. J.; Rose, F.; Smolyanyuk, A.; Calzolari, A.; et al. aflow.org: A web ecosystem of databases, software and tools. *Comput. Mater. Sci.* **2023**, *216*, 111808.
(74) Oses, C.; Esters, M.; Hicks, D.; Divilov, S.; Eckert, H.; Friedrich, R.; Mehl, M. J.; Smolyanyuk, A.; Campilongo, X.; van de Walle, A.; et al. aflow++: A C++ framework for autonomous materials design. *Comput. Mater. Sci.* **2023**, *217*, 111889.
(75) Hjorth Larsen, A.; Jørgen Mortensen, J.; Blomqvist, J.; Castelli, I. E.; Christensen, R.; Dułak, M.; Friis, J.; Groves, M. N.; Hammer, B.; Hargus, C.; et al. The atomic simulation environment—a Python library for working with atoms. *J. Phys.: Condens. Matter* **2017**, *29*, 273002.
(76) Togo, A.; Tanaka, I. Spglib: a software library for crystal symmetry search. *arXiv* **2018**, arXiv:1808.01590. (accessed Nov 19, 2023)
(77) Nikiforov, I.; Tadmor, E. B. *Equilibrium structure and energy for a crystal structure at zero temperature and pressure v000*; OpenKIM, 2023. DOI: 10.25950/53ef2ea4
(78) Curtarolo, S.; Setyawan, W.; Wang, S.; Xue, J.; Yang, K.; Taylor, R. H.; Nelson, L. J.; Hart, G. L.; Sanvito, S.; Buongiorno-Nardelli, M.; et al. AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations. *Comput. Mater. Sci.* **2012**, *58*, 227−235.
(79) Nikiforov, I.; Tadmor, E. B.; ki Choi, M.; Karls, D. S. *Equilibrium structure and energy for a crystal specified using LAMMPS type labels at zero temperature and pressure v000*; OpenKIM, 2023. DOI: 10.25950/86dce899
(80) Hazen, R. M. Effects of temperature and pressure on the cell dimension and X-ray temperature factors of periclase. *Am. Mineral.* **1976**, *61*, 266.
(81) Barron, T.; Collins, J.; Smith, T.; White, G. Thermal expansion, Gruneisen functions and static lattice properties of quartz. *J. Phys. C: Solid State Phys.* **1982**, *15*, 4311−4326.
(82) Voyiatzis, E.; Afantitis, A.; Sun, J.; Stirner, T.; Hagston, W. E.; Leyland, A.; Matthews, A. *LAMMPS Buckingham potential for MgO developed by Sun et al. (2006) v000*; OpenKIM, 2021. DOI: 10.25950/5cb45fc8
(83) Sun, J.; Stirner, T.; Hagston, W. E.; Leyland, A.; Matthews, A. A simple transferable interatomic potential model for binary oxides applied to bulk a-Al_2O_3 and the (0001) a-Al_2O_3 surface. *J. Cryst. Growth* **2006**, *290*, 235−240.
(84) Voyiatzis, E.; Afantitis, A.; Carré, A.; Horbach, J.; Ispas, S.; Kob, W. *LAMMPS Buckingham potential for SiO2 developed by Carré et al. (2008) v000*; OpenKIM, 2021. DOI: 10.25950/dbac5d53

(85) Carré, A.; Horbach, J.; Ispas, S.; Kob, W. New fitting scheme to obtain effective potential from Car-Parrinello molecular-dynamics simulations: Application to silica. *Europhys. Lett.* **2008**, *82*, 17001.
(86) Freitas, A. A.; Santos, R. L.; Colaço, R.; Lopes, J. N. C. *LAMMPS Buckingham potential for $CaO-Al_2O_3-SiO_2$ systems developed by Freitas et al. (2015) v000*; OpenKIM, 2023. DOI: 10.25950/de76655f
(87) Freitas, A. A.; Santos, R. L.; Colaço, R.; Bayão Horta, R.; Canongia Lopes, J. N. From lime to silica and alumina: systematic modeling of cement clinkers using a general force-field. *Phys. Chem. Chem. Phys.* **2015**, *17*, 18477−18494.
(88) Brugnoli, L.; Miyatani, K.; Akaji, M.; Urata, S.; Menziani, M. C.; Pedone, A. *LAMMPS ReaxFF potential for Ceria/Silica/Water/NaCl developed by Brugnoli et al. (2023) v000*; OpenKIM, 2023. DOI: 10.25950/89e866e6
(89) Brugnoli, L.; Menziani, M. C.; Urata, S.; Pedone, A. *Development and Application of a ReaxFF Reactive Force Field for Cerium Oxide/Water Interfaces. The Journal of Physical Chemistry A*; American Chemical Society, 2021.
(90) Brugnoli, L.; Miyatani, K.; Akaji, M.; Urata, S.; Pedone, A. *New Atomistic Insights on the Chemical Mechanical Polishing of Silica Glass with Ceria Nanoparticles. Langmuir*; American Chemical Society, 2023; Vol. *39*, p 5527.
(91) Manzano, H.; Moeini, S.; Marinelli, F.; van Duin, A. C. T.; Ulm, F.-J.; Pellenq, R. J.-M. *LAMMPS ReaxFF potential for Ca-Si-O-H systems developed by Manzano et al. (2012) v000*; OpenKIM 2022. DOI: 10.25950/af9b5456
(92) Manzano, H.; Moeini, S.; Marinelli, F.; van Duin, A. C. T.; Ulm, F.-J.; Pellenq, R. J.-M. Confined Water Dissociation in Microporous Defective Silicates: Mechanism, Dipole Distribution, and Impact on Substrate Properties. *J. Am. Chem. Soc.* **2012**, *134*, 2208−2215.
(93) Broughton, J. Q.; Meli, C. A.; Vashishta, P.; Kalia, R. K. *LAMMPS Vashishta potential for the Si-O system developed by Broughton et al. (1997) v000*; OpenKIM, 2019. DOI: 10.25950/9522b8ae
(94) Broughton, J. Q.; Meli, C. A.; Vashishta, P.; Kalia, R. K. Direct atomistic simulation of quartz crystal oscillators: Bulk properties and nanoscale devices. *Phys. Rev. B* **1997**, *56*, 611−618.
(95) Vashishta, P.; Kalia, R. K.; Nakano, A. *LAMMPS Vashishta potential for the Si-O system developed by Nakano et al. (1994) v000*; OpenKIM, 2019. https://doi.org/10.25950/933b35ad
(96) Nakano, A.; Kalia, R. K.; Vashishta, P. First sharp diffraction peak and intermediate-range order in amorphous silica: finite-size effects in molecular dynamics simulations. *J. Non-Cryst. Solids* **1994**, *171*, 157−163.
(97) Rino, J. P.; Vashishta, P.; Kalia, R. K.; Ebbsjö, I. *LAMMPS Vashishta potential for the Si-O system developed by Vashishta et al. (1990) v000*; OpenKIM, 2019. https: //doi.org/10.25950/c958fbaf
(98) Vashishta, P.; Kalia, R. K.; Rino, J. P.; Ebbsjö, I. Interaction potential for SiO2: A molecular-dynamics study of structural correlations. *Phys. Rev. B* **1990**, *41*, 12197−12209.
(99) Munetoh, S.; Motooka, T.; Moriguchi, K. *Tersoff-style three-body potential for SiO developed by Munetoh et al. (2007) v000*; OpenKIM, 2022. https://doi.org/10.25950/59df4e98
(100) Munetoh, S.; Motooka, T.; Moriguchi, K.; Shintani, A. Interatomic potential for Si−O systems using Tersoff parameterization. *Comput. Mater. Sci.* **2007**, *39*, 334−339.
(101) Brink, T.; Thompson, A. P.; Farrell, D. E.; Wen, M.; Tersoff, J.; Nord, J.; Albe, K.; Erhart, P.; Nordlund, K.; Ziegler, J. F.; et al. *Model driver for Tersoff-style potentials ported from LAMMPS v005*; OpenKIM, 2021. DOI: 10.25950/9a7dc96c

https://doi.org/10.1021/acs.jpcb.3c08419
J. Phys. Chem. B 2024, 128, 3282−3297