

RESEARCH ARTICLE | MARCH 20 2025

Adaptive-precision potentials for large-scale atomistic simulations

Special Collection: [Molecular Dynamics, Methods and Applications 60 Years after Rahman](#)

David Immel ; Ralf Drautz ; Godehard Sutmann  



J. Chem. Phys. 162, 114119 (2025)

<https://doi.org/10.1063/5.0245877>



Articles You May Be Interested In

Nanostructural interpretation for elastic softening of amorphous carbon induced by the incorporation of silicon and hydrogen atoms

J. Appl. Phys. (June 2010)

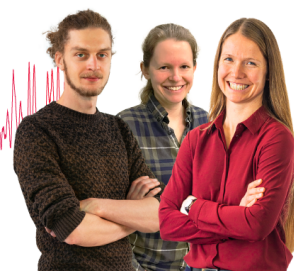
Webinar From Noise to Knowledge

May 13th – Register now



Zurich
Instruments

Universität
Konstanz



Adaptive-precision potentials for large-scale atomistic simulations

Cite as: J. Chem. Phys. 162, 114119 (2025); doi: 10.1063/5.0245877

Submitted: 29 October 2024 • Accepted: 9 February 2025 •

Published Online: 20 March 2025



David Immel,¹  Ralf Drautz,²  and Codehard Sutmann^{1,2,a)} 

AFFILIATIONS

¹Jülich Supercomputing Centre (JSC), Institute for Advanced Simulation (IAS), Forschungszentrum Jülich, Jülich, Germany

²Interdisciplinary Centre for Advanced Materials Simulations (ICAMS), Ruhr Universität Bochum, Bochum, Germany

Note: This paper is part of the JCP Special Topic on Molecular Dynamics, Methods and Applications 60 Years After Rahman.

a) Author to whom correspondence should be addressed: g.sutmann@fz-juelich.de

ABSTRACT

Large-scale atomistic simulations rely on interatomic potentials, providing an efficient representation of atomic energies and forces. Modern machine-learning (ML) potentials provide the most precise representation compared to electronic structure calculations, while traditional potentials provide a less precise but computationally much faster representation and, thus, allow simulations of larger systems. We present a method to combine a traditional and a ML potential into a multi-resolution description, leading to an adaptive-precision potential with an optimum of performance and precision in large, complex atomistic systems. The required precision is determined per atom by a local structure analysis and updated automatically during simulation. We use copper as demonstrator material with an embedded atom model as classical force field and an atomic cluster expansion (ACE) as ML potential, but, in principle, a broader class of potential combinations can be coupled by this method. The approach is developed for the molecular-dynamics simulator LAMMPS and includes a load-balancer to prevent problems due to the atom dependent force-calculation times, which makes it suitable for large-scale atomistic simulations. The developed adaptive-precision copper potential represents the ACE-forces with a precision of 10 me V/Å and the ACE-energy exactly for the precisely calculated atoms in a nanoindentation of 4×10^6 atoms calculated for 100 ps and shows a speedup of 11.3 compared with a full ACE simulation.

© 2025 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>). <https://doi.org/10.1063/5.0245877>

I. INTRODUCTION

Molecular dynamics (MD) simulations have developed into a powerful tool to get insight into complex atomistic systems and to study their evolution in phase space. The length and time scale, which can be covered, are strongly dependent on the cost of energy- and force-computations between the particles. The length and size of systems have been continuously extended over the years using parallel computing. However, the accuracy of the interactions between particles in long time simulations has been limited in the past due to the approximative nature of classical force-field descriptions. In recent years, the development and application of machine learning (ML) potentials [such as the atomic-cluster expansion (ACE),¹ moment tensor potentials,² Gaussian approximation potentials,³ spectral neighbor analysis potentials,⁴ and neural network

potentials⁵] have received a lot of attention for the combination of computational performance and accuracy. Accurate ML potentials are often based on large sets of density functional theory (DFT) data,^{4,6–8} used as a reference, and therefore translate the accuracy of DFT calculations⁹ into dynamic simulations. However, there is still a considerable performance gap between the evaluation of classical fields and ML potentials, e.g., the force calculation for a classical EAM potential and the ML-based ACE method differ by a factor of 100–1000,⁸ which still limits the accessibility of time and length scales of systems, completely described by highly accurate descriptions, and therefore poses a conflict between system size and precision.

Due to the high computational demands of accurate interaction models, there have been several attempts to couple low- to high-accurate descriptions adaptively within a simulation. In QM/MM

simulations, quantum mechanical descriptions based on DFT are coupled to classical force fields.^{10–12} Due to the large computational demand, the QM region is limited to a small sub-system. On a coarser level, e.g., in soft matter,^{12–15} an adaptive description has been introduced to reduce the number of degrees of freedom in a system to speed up the computations. On even larger scales, where individual particles can be combined to groups of particles in specific regions of the system, coupling between classical atomistic simulations and continuum simulations (such as the material point method¹⁶ or finite element simulations¹⁷ or with coarse-grained systems as in Refs. 18 and 19) is performed. The commonality of the given examples is that one can separate different subsystems, i.e., one subsystem, which has to be described with high precision because of high sensitivity of results on the underlying method, and one subsystem, which serves as a material background, which is not strongly perturbed and can be described by cheaper, less accurate methods. Since the perturbation state of a material might show temporal and spatial dependence, such a distinction between subsystems ideally works dynamically and adaptively.

In this article, we present such an adaptive-precision approach for MD simulations as there is a broad range of applications: In chemistry, e.g., catalytic reactions at surfaces or interfaces require a precise treatment of reacting atoms and surface atoms, but not for atoms far below the surface or far away from reaction centers. In physics, surface-growth processes, for example, for semiconductors, require a precise treatment of the surface, but with evolving growth and agglomeration of many new atoms, the atoms belonging to the original surface layer become less important and can be calculated with less precision, i.e., they do not contribute actively any more to the physical growth process. In materials science, e.g., dynamical crack propagation requires only a precise simulation of the crack tip.

Due to the change of energy and force description in the two regions (high-/low-accuracy), a spatial zone is usually introduced, which provides a smooth transition from one to the other region.^{20,21} As a first approach, one can weight both energy and force per atom with a factor that changes from 0 to 1.²¹ However, this first approach violates both energy and momentum conservation and is thus limited to the canonical ensemble. For consistency, either the energy or force can be weighted by a factor.²⁰ Since the force is entering directly into the integration of motion, several approaches^{14,22,23} are based on a weighting of forces. However, it was shown in Ref. 20 that this approach does not result in energy conservation since the underlying potential cannot be properly reconstructed from the weighted forces. An approach to generate energy conserving dynamics has been taken in Refs. 18, 24, and 25, where the weighting has been included in the Hamiltonian description.

With the advent of ML potentials, coupling of different interaction models shows strong potential for combining a high accurate description in small-to-medium sized subsystems with acceptable computational costs and a low accurate description in large subsystems, opening the path to long time and large length scale simulations, including high accuracy, where it is needed. Such a coupling between a ML-potential and a classical force-field description has been proposed in Ref. 23 using a force-mixing approach to simulate a grand-canonical ensemble in the thermodynamic limit.

In contrast, in the present article, we use an energy-mixing approach that is consistent with a Hamiltonian description and allows, in principle, the simulation of a microcanonical ensemble. In so doing, we introduce a method to couple a precise ML-potential for subregions of atoms of interest and a fast traditional potential for the remaining system components to overcome the conflict between system size and precision for classical atomistic simulations. The atoms of interest are thereby automatically detected by a customizable detection mechanism, and therefore, the method works autonomously and self-adaptively in space and time. To further reduce the execution time, we implemented the method into the parallel simulation engine LAMMPS.²⁶

The present paper is organized as follows: We first introduce our adaptive-precision model in Sec. II, whereas we present the energy-model in Sec. II A, discuss the group of interest detection in Sec. II B, and the integration of motion in Sec. II C. With adaptive-precision and dynamic precision-selection in a parallel simulation comes the need for dynamic load balancing as the compute time changes over magnitudes between atoms. Therefore, we present our dynamic load-balancing method in Sec. II D. We applied the introduced adaptive-precision mechanism and combined an EAM potential with an ACE potential for copper. The input potentials are presented in Sec. III, and the EAM potential is improved to be used in combination with the ACE potential. Finally, we demonstrate in Sec. IV the capabilities in terms of precision and efficiency of the adaptive-precision copper potential for a nanoindentation of 4×10^6 atoms calculated for 100 ps with LAMMPS.²⁶ In Sec. V, we conclude.

II. METHOD

A. Representation of energy and force

The total energy of the system is given as $H = \sum_i (T_i + E_i) + E_{\text{ext}}$, where T_i is the kinetic energy and E_i is the potential energy of atom i and E_{ext} are possibly existing external fields. The energy of our adaptive-precision approach combines a precise and a fast energy, $E_i^{(p)}$ and $E_i^{(f)}$, per atom i . We will use the atomic cluster expansion (ACE)¹ as precise and the embedded atom model (EAM)²⁷ as fast interatomic potential. Precise and fast energies are combined with a continuous switching parameter $\lambda_i \in [0, 1]$ per atom i , namely, to the adaptive-precision potential energy,

$$E_i = \lambda_i E_i^{(f)} + (1 - \lambda_i) E_i^{(p)}. \quad (1)$$

During the course of a simulation, the energy will be switched automatically from fast to precise or vice versa, as required by the local atomic environment. The switching parameter λ_i needs to change continuously to prevent shocks by instant energy changes. One can save computation time by using the adaptive-precision energy E_i instead of the precise energy $E_i^{(p)}$ since the precise calculation is only required for a subset of atoms in the adaptive-precision calculation. One needs to make sure that the energies $E_i^{(f)}$ and $E_i^{(p)}$ are as similar as possible for the atoms, which are calculated with the fast potential, to prevent a systematic energy change due to the change of the switching parameter. The adaptive-precision energy, according

to Eq. (1), gives the total potential energy $\sum_k E_k = \sum_k (\lambda_k E_k^{(f)} + (1 - \lambda_k) E_k^{(p)})$ from where the adaptive-precision force $F_i = -\nabla_i \sum_k E_k$ can be calculated as

$$F_i = \sum_k \left(-\lambda_k (\nabla_i E_k^{(f)}) - (1 - \lambda_k) (\nabla_i E_k^{(p)}) + (\nabla_i \lambda_k) (E_k^{(p)} - E_k^{(f)}) \right). \quad (2)$$

It is important to note that $\lambda_i = 0$ is not the only requirement for a precise force on atom i since the force F_i depends, according to Eq. (2), on the switching parameters λ_k of all atoms k within the force cutoffs of the interatomic potentials. For visualization, we define the force $f_{k \rightarrow i}^{(M)}$ from atom k on atom i according to the interatomic model M as $F_i^{(M)} = \sum_{k \neq i} f_{k \rightarrow i}^{(M)} \nabla_i r_{ik}$. In contrast to the total force, the individual pair force contributions of two different interatomic potentials are not necessarily correlated, as shown in Fig. 1 for an EAM and an ACE copper potential. Figure 1(a) is point-symmetrical to the origin due to Newton's third law of motion. The Pearson correlation coefficients (PCCs) of the individual pair force contributions $\{f_{k \rightarrow i}^{(M)} \nabla_i r_{ik}\}$ between EAM and ACE calculated for the first four atom neighbor shells are, in ascending order of shells, -0.530 , -0.747 , -0.984 , and 0.954 . Due to the smaller force-cutoff, the atoms of the 5th neighbor shell are only partially evaluated by EAM. Thus, we only calculate the PCCs for the first four neighbor shells. In contrast, the Pearson correlation coefficient of the total forces, which are visualized in Fig. 1(b), is 0.998 . The PCCs and the visualization in Fig. 1(a) show that both potentials have qualitatively different mechanisms to construct a quantitatively only slightly different total force. Therefore, a smooth transition zone between fast and precise energy description is essential to smoothly transfer the force-construction mechanism of the precise potential into the one of the fast potential.

B. Adiabatic switching

The switching function λ changes the energy-precision [cf. Eq. (1)] and thereby also the force-precision during simulation dependent on the atomic environment. This change of precision is ideally quasi-adiabatic to prevent perturbations of the system. The switching function itself generally has no physical meaning; it is only an auxiliary function to determine the precision. However, the force on atom i depends, according to Eq. (2), on $(\nabla_i \lambda_k)$ and, thus, not only on the value of the switching parameter but also on the calculation mechanism. Therefore, an adiabatically slow change of the switching parameter is important to minimize the unphysical force contribution of $(\nabla_i \lambda_k)$.

The switching function needs a detection mechanism for particles that require a precise calculation. This detection mechanism strongly depends on the simulation. The centro-symmetry parameter²⁹ (CSP) detects, e.g., defects and surface atoms, whereas the common neighbor analysis (CNA)³⁰ can characterize grain boundary structures.³¹ The common neighborhood parameter (CNP)³² combines CNA and CSP. CSP, CNA, and CNP are implemented in LAMMPS^{33–35} and, therefore, are accessible to a wider community. Depending on the underlying physical system and simulation, other distinctions are possible. To detect, for example, stress-related events, the von-Mises stress can be used as an indicator. For dynamic

simulations of atom bombardment, as in Ref. 36, the particle velocity can be used, and in simulations of damages, as for crack propagation, the strain energy can serve as an indicator.²¹ For a coating, as in Ref. 37, one can include only atoms of the same element like the central atom i in the calculation of CSP_i to make the interface look like a surface.

The switching parameters $\{\lambda_k\}$ of all atoms k within the force cutoff of atom i determine the force-precision of atom i , as discussed in Sec. II A. Therefore, when an atom i is detected for a precise calculation, the switching function needs to change also the switching parameters λ_k of the neighboring atoms k to ensure a precise force calculation of atom i .

To detect defects and surfaces, we assume here the CSP as detection mechanism. However, it can be changed to other criteria without big implementation overhead. The CSP is given as

$$\text{CSP}_i(t) = \sum_{j=1}^{N/2} \left(\tilde{r}_{ij}(t) + \tilde{r}_{i,j+N/2}(t) \right)^2, \quad (3)$$

whereas N is the number of nearest neighbors and j and $j + N/2$ correspond to a pair of opposite nearest neighbors of the central atom i . How to identify these pairs is discussed in detail in Appendix A.

The construction of our switching function, taking into account the requirements discussed, is described in Appendix B.

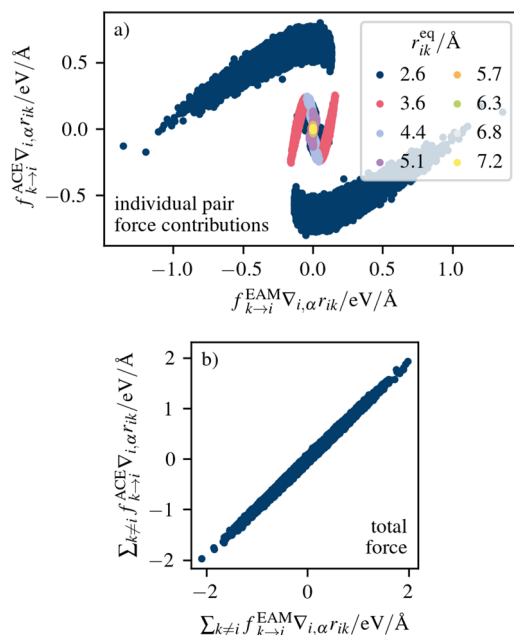


FIG. 1. (a) Comparison of the individual pair force contributions $f_{k \rightarrow i}^{(M)}$ for a system of 13 500 copper atoms at 300 K with periodic-boundary conditions in all spatial dimensions α . The force contributions are colored according to the equilibrium distance r_{ik}^{eq} of the corresponding atom pair ik . The compared potentials are a modified EAM2 of Ref. 28 and ACE of Ref. 8, both of which are described in Sec. III. (b) Total force F_i on an atom i , which is given by the sum over the individual pair force contributions of all neighbors k .

C. Integration of motion

1. Velocity-Verlet integration

With a known velocity $v_i(t)$, position $x_i(t)$, force $F_i(t)$, and switching parameter $\lambda_i(t)$ of the last time step t , one can propagate an atom in time. A time step of Δt according to the velocity-Verlet integration³⁸ updated for use with an adaptive-precision potential is given as

$$1. \quad v_i\left(t + \frac{\Delta t}{2}\right) = v_i(t) + \frac{F_i(t)}{m_i} \frac{\Delta t}{2}, \quad (4a)$$

$$x_i(t + \Delta t) = x_i(t) + v_i\left(t + \frac{\Delta t}{2}\right) \Delta t, \quad (4b)$$

$$2. \quad \lambda_i(t + \Delta t) = \lambda_i(\lambda_i(t), \{x(t + \Delta t)\}), \quad (4c)$$

$$3. \quad F_i(t + \Delta t) = F_i(\{x_i(t + \Delta t)\}, \{\lambda_i(t + \Delta t)\}), \quad (4d)$$

$$E_i(t + \Delta t) = E_i(\{x_i(t + \Delta t)\}, \lambda_i(t + \Delta t)), \quad (4e)$$

$$4. \quad v_i(t + \Delta t) = v_i\left(t + \frac{\Delta t}{2}\right) + \frac{1}{2} \frac{F_i(t + \Delta t)}{m_i} \Delta t, \quad (4f)$$

whereas step 2 in Eq. (4) is new and required due to the adaptive-precision potential. The first velocity and position calculations for $v_i(t + \Delta t/2)$ and $x_i(t + \Delta t)$ use the given $\lambda_i(t)$. A new switching parameter $\lambda_i(t + \Delta t)$ has to be calculated in the second step of the integration since the positions of the particles have changed and, thus, also their switching parameter. Therefore, the afterward calculated quantities $F_i(t + \Delta t)$, $E_i(t + \Delta t)$, and $v_i(t + \Delta t)$ depend on the newly calculated switching parameter $\lambda_i(t + \Delta t)$.

2. Avoiding unphysical force-contributions by $(\nabla_i \lambda_k)$

The force calculation, according to Eq. (4d), requires, according to Eq. (2), the knowledge of $(\nabla_i \lambda_k)$. The force contribution through this gradient of the switching function should be small as a slowly changing switching function is used. However, as discussed in Sec. II B, the switching function generally has no physical meaning but only influences the required precision. Furthermore, the need to change the switching parameter adiabatically slowly implies the usage of time averages whose gradient cannot be calculated consistently. Thus, we want to neglect $(\nabla_i \lambda_k)$ in Eq. (2), but this would violate energy-conservation. The adaptive-precision potential is conservative for a constant set of $\{\lambda_i\}$ as the calculation of $\nabla_i \lambda_k$ is not required in this case. This conservative reference system offers the possibility of neglecting the unphysical force contribution of $(\nabla_i \lambda_k)$ in an energy and momentum-conserving way. We want to stress that if λ is kept constant per atom over time, we have a conservative dynamics and, therefore, conserve energy and momentum naturally. Thus, we can use the system with constant switching parameters as reference system for each atom to apply a momentum-conserving correction for energy conservation, which is described in the following. Note that the presented correction approach is independent of the calculation of λ and can be used for any switching function.

The energy difference ΔH_i of atom i at the end of a time step between the conservative reference system and the system with updated switching parameters is caused by the change of the switching parameter. To calculate this energy error ΔH_i , one has to finish the calculation of an integration step for both sets of switching parameters. As the forces of the reference system and the dynamic- λ

system depend on the same atomic positions, one can easily calculate both forces during the force calculation by just summing up two forces weighted with the corresponding switching parameter. This energy difference affects both the kinetic energy change ΔT_i and the potential energy change ΔE_i , namely,

$$\Delta H_i = \Delta T_i + \Delta E_i. \quad (5)$$

The potential energy change ΔE_i affects only atoms whose switching parameter has changed [cf. Eq. (1)]. The kinetic energy change ΔT_i affects atoms in whose force cutoff a switching parameter has changed [cf. Eq. (2)]. The calculation of ΔE_i and ΔT_i is described in Appendix C 1.

We want to add or remove energy to compensate for the work ΔH_i performed on the system when the switching parameters $\{\lambda_i\}$ are changed. Since one cannot simply change the potential energy in a predictable way, we change the kinetic energy of particles. The energy errors are calculated per particle i and should also be corrected locally. Therefore, we need a local thermostat that changes the kinetic energy of atoms by the measured energy error ΔH_i . The local thermostat rescales the momenta of a group of atoms Ω_i relative to the center-of-mass velocity by the factor $\beta(\Omega_i, \Delta H_i)$. The rescaling conserves the momentum as the rescaling is relative to the center-of-mass velocity. The rescaling, including the calculation of the rescaling factor $\beta(\Omega_i, \Delta H_i)$, is described in detail in Appendix C 2. The integrator changes the velocity of particles of mass m according to $\Delta v = \Delta t F/m$ due to a force F . One can understand the energy-error correction as an additional force that is just used at the end of a time step but not in the next integrator step. The application of this local thermostat contains a stochastic component, as the direction of the effectively applied force is in the direction of the relative momentum and, thus, stochastic. Note that this local thermostat is not considered to provide a NVT ensemble. It can be considered as an energy correction but not primarily as a temperature correction, i.e., the objective function is not the temperature but the energy. Thus, non-equilibrium simulations, non-energy conserving simulations such as a NVT ensemble, and simulations with external forces can be performed with the local thermostat.

D. Load balancing

Time integrating the equations of motion for a set of atoms described by a short-range interatomic potential requires per atom only information from neighboring atoms within a cutoff distance. Thus, LAMMPS uses distributed-memory parallelism via MPI.²⁶ The simulation box is divided into disjoint domains that cover the whole simulation box according to a spatial domain-decomposition approach.³⁹ Each MPI task, in the following denoted as processor, is assigned to one domain and administers all particles located in this domain. Our adaptive-precision potential requires a highly different compute time per atom dependent on the used potential, and that causes load-imbalances between the domains. Hence, we present a load-balancing approach to dynamically change the domain sizes during a simulation.

We calculate the centro-symmetry parameter $\text{CSP}_i(t)$ according to Eq. (3) and the switching parameter λ according to Eq. (B4) within the force-calculation routine to include both in the load balancing. Thus, we perform four independent calculations during the force-calculation routine. We execute first the calculation

of the fast potential (FP), second the precise potential (PP), third CSP_{*i*} (CSP), and fourth $\lambda_i(\lambda)$. Since all four calculations are required only for a subset of particles, as discussed in Appendix G in detail, the code inherently needs load balancing. Required adjustments of the interatomic-potentials to allow effective load-balancing for adaptive-precision potentials are discussed in detail in Appendix H.

We assume per processor p and subroutine $\mathcal{R}$ an average work per particle,

$$\rho_p^{(\mathcal{R})} = \tau_p^{(\mathcal{R})} / N_p^{(\mathcal{R})}, \quad (6)$$

whereas the total duration $\tau_p^{(\mathcal{R})}$ and the number of calculations $N_p^{(\mathcal{R})}$ are measured within the force-calculation subroutine $\mathcal{R}$. We rescale the work $\rho_p^{(\mathcal{R})}$ with a constant factor per processor p in order to match the total work of our model with the also measured total calculation time τ_p of the processor. Thereby, one can assign an individual load ρ_i^{atom} to atom i with possible contributions of $\rho_{p(i)}^{\text{FP}}$, $\rho_{p(i)}^{\text{PP}}$, $\rho_{p(i)}^{\text{CSP}}$, and $\rho_{p(i)}^{\lambda}$. The load ρ_i^{atom} is used as input for a load-balancing algorithm. We use a staggered grid,⁴⁰ which is used for particle simulations⁴¹ and which is implemented in the load-balancing library ALL.⁴²

Note that the measured force-calculation times can be used to estimate on-the-fly the calculation time $\tau_{\text{system}}^{\text{PP}}$ that it would take to calculate the entire system with the precise potential. As load-imbalances are negligible when the whole system is calculated only with the same potential, $\tau_{\text{system}}^{\text{PP}}$ is given as

$$\tau_{\text{system}}^{\text{PP}} = \frac{1}{P} \sum_p \rho_p^{\text{PP}} N_p, \quad (7)$$

where N_p denotes the number of atoms administered by processor p and P denotes the number of processors. The estimated speedup S of the adaptive-precision potential is then

$$S = \frac{\tau_{\text{system}}^{\text{PP}}}{\max_p(\tau_p)}. \quad (8)$$

For the adaptive-precision to be beneficial, $S > 1$ must apply. Therefore, the on-the-fly speedup estimation can be used to detect on-the-fly whether the by simplified calculations saved calculation time outweighs the overhead of calculating detection mechanism and switching parameter, i.e., whether $S > 1$ applies. For $S \leq 1$, one can then switch off the adaptive-precision potential and use the precise potential instead.

III. ADAPTIVE-PRECISION POTENTIAL

A. Atomic cluster expansion

We use the atomic cluster expansion (ACE) of Ref. 8 as precise potential, as it is a modern ML-potential with a good representation of its DFT reference data. The energy E_i of an atom i described by the atomic cluster expansion¹ is

$$E_i^{\text{ACE}} = \phi_i^{(1)} + \sqrt{\phi_i^{(2)}}, \quad (9)$$

whereas the functions $\phi_i^{(p)}$ are expanded as

$$\phi_i^{(p)} = \sum_{\mathbf{v}} c_{\mathbf{v}}^{(p)} B_{i\mathbf{v}}, \quad (10)$$

where $B_{i\mathbf{v}}$ are product basis functions that describe the atomic environment and $c_{\mathbf{v}}^{(p)}$ are fitted expansion coefficients with the multi-indices $\mathbf{v}$. We use the copper ACE potential from Ref. 8.

B. Embedded atom model

We use the EAM2 potential of Ref. 28 as a starting point for the fast potential since it is widely used.^{43–46} The energy of an atom i described with the embedded atom model²⁷ (EAM) is

$$E_i^{\text{EAM}} = \xi \left(\sum_{j \neq i} \zeta(r_{ij}) \right) + \frac{1}{2} \sum_{j \neq i} \Phi(r_{ij}), \quad (11)$$

where ξ is the embedding function, ζ is the electron density, and Φ is a pair potential. We use a copper potential from Ref. 28, which is described in Appendix E. We need to optimize the EAM potential to minimize systematic energy differences between the used EAM and ACE potentials. Furthermore, the optimization gives the possibility to improve the forces as well. The EAM potential is fitted for applications with surfaces, vacancies, and interstitials that are detected for a precise calculation and, therefore, do not need to be described by the fast and less accurate EAM potential. To correct an energy offset, we introduce ξ_0^{Fit} to the original embedding function $\xi(x)$ [Eq. (E4)], namely, $\xi^{\text{Fit}}(x) = \xi(x) + \xi_0^{\text{Fit}}$ and use $\xi^{\text{Fit}}(x)$ instead of $\xi(x)$. The set of parameters $\mathcal{A}$ (cf. Appendix E) is optimized by minimizing a loss function $\mathcal{L}$ with atomicrex.⁵¹ The loss function includes atomic energies $E_{s,i}$ and forces $F_{s,i}$ of different structures s with N_s atoms as well as scalar properties A_o . We use the lattice parameter a_0^{FCC} , the cohesive energy, and elastic constants as the scalar properties. The target values denoted with “targ” are the corresponding values calculated with the highly accurate ACE potential. The predicted values denoted with “pred” are calculated with the current set of parameters $\mathcal{A}$. The differences between predicted and target values are per quantity weighted with a tolerance δ^{tol} . The used target values and tolerances of the loss function are listed in Table I. The minimized loss function is

$$\begin{aligned} \mathcal{L}(\mathcal{A}) = & \sum_{s \in \mathcal{S}} \frac{1}{N_s} \left(\sum_{i=1}^{N_s} \left(\frac{E_{s,i}^{\text{targ}} - E_{s,i}^{\text{pred}}}{\delta_{E_{\text{md}}}^{\text{tol}}} \right)^2 + \sum_{i=1}^{N_s} \left(\frac{\|F_{s,i}^{\text{targ}} - F_{s,i}^{\text{pred}}\|}{\delta_{F_{\text{md}}}^{\text{tol}}} \right)^2 \right) \\ & + \sum_{o \in \mathcal{O}} \left(\frac{A_o^{\text{targ}} - A_o^{\text{pred}}}{\delta_o^{\text{tol}}} \right)^2, \end{aligned} \quad (12)$$

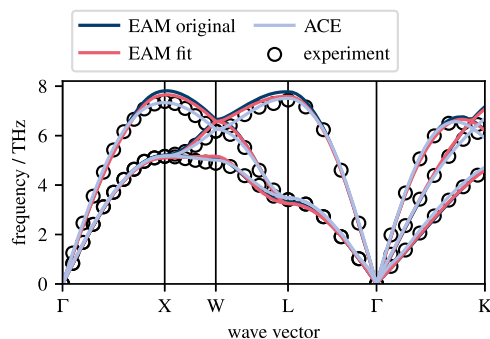
TABLE I. Target values with tolerance used in the loss function Eq. (12) for the optimization of the EAM potential.

Bulk property	Target value	Tolerance δ^{tol}
Lattice parameter a_0^{FCC}	3.6309 Å	0.001 Å
Cohesive energy E_{coh}	−3.6995 eV/atom	0.01 eV/atom
Bulk modulus B	138.2 GPa	1 GPa
Elastic constant C_{11}	173.8 GPa	1 GPa
Elastic constant C_{12}	120.5 GPa	1 GPa
Elastic constant C_{44}	77.8 GPa	1 GPa
Force F_{md}	MD simulation	0.01 eV Å ^{−1}
Atomic energy E_{md}	MD simulation	0.01 eV

TABLE II. Comparison of quantities calculated with EAM, optimized EAM, and ACE with DFT and experimental reference values.

	ACE	EAM Original	EAM Fit	DFT	Exp.
Lattice constant/Å	3.6309	3.6149	3.6305		
Cohesive energy/eV	−3.6995	−3.5579	−3.6957		
Vacancy formation energy/eV	1.1285	1.2735	1.0589	1.07 ⁸	1.27 ⁴⁷
Surface energy 111/Jm ^{−2}	1.3566	1.2443	0.8802	1.36 ⁸	
Surface energy 100/Jm ^{−2}	1.5111	1.3515	0.9765	1.51 ⁸	
Surface energy 110/Jm ^{−2}	1.5889	1.4822	1.1225	1.57 ⁸	
Interstitial formation energy/eV					
100-dumbbell	3.1025	3.0809	2.9662	3.10 ⁴⁸	2.8–4.2 ⁴⁹
Octahedral	3.3392	3.2559	3.1303	3.35 ⁴⁸	
Tetrahedral	3.6217	3.5655	3.4240	3.64 ⁴⁸	
Elastic constant C ₁₁ /GPa	173.8	171.5	171.3	177 ⁸	177 ⁵⁰
Elastic constant C ₁₂ /GPa	120.5	124.2	120.1	132 ⁸	125 ⁵⁰
Elastic constant C ₄₄ /GPa	77.8	76.2	80.6	82 ⁸	81 ⁵⁰

with the set of scalar observables $\mathcal{O}$ and the set of structures $\mathcal{S}$. We use two snapshots of a NVE simulation of copper with $10 \times 10 \times 10$ unit cells at 300 K and periodic boundary conditions as structures for the fit. These structures are sufficient since we start with an already tested and validated EAM potential, and the optimized EAM potential is for use only with atomic configurations similar to the used structures. We start the fitting with the parameters of EAM2 from Ref. 28 and $\xi_0^{\text{Fit}} = 0$ eV. In a first run, only the energy offset ξ_0^{Fit} is fitted. Afterward, all parameters, including ξ_0^{Fit} , are fitted. The optimized parameters are listed in Appendix E in Table III. Scalar properties calculated with the optimized EAM potential are shown in Table II. Lattice constant and cohesive energy of the optimized EAM potential match through the optimization. The elastic constants of ACE are in good agreement with both EAM and optimized EAM potential. The phonon spectra are compared in Fig. 2 and are in good agreement.

**FIG. 2.** Phonon spectra calculated with ASE⁵² of the fitted EAM potential, original EAM potential, the ACE potential, and experimental values⁵³ measured by neutron diffraction at the temperature of 80 K.

C. Combined EAM and ACE potential

We combine the ACE potential and the optimized EAM potential into an adaptive-precision copper potential according to Eq. (1) with the switching function λ , which is described in Sec. II (b) and Appendix B. There are parameters in three components: first, in the mechanism used to detect atoms that require a precise calculation; second, in the switching function; and third, in the local thermostat used to avoid unphysical force contributions by $(\nabla_i \lambda_k)$. We discuss how to select these parameters in Appendix F, whereas the parameter set is listed in Table IV. We refer to the adaptive-precision potential with this parameter set as Hyb1.

IV. DEMONSTRATION

We calculated a nanoindentation with 4×10^6 atoms and a (100)-surface at 300 K for 100 ps on JURECA-DC⁵⁴ with the adaptive-precision copper potential Hyb1 (cf. Table IV) to analyze precision and saved computation time compared to a full ACE simulation.

A. Precision

The force difference $\Delta F^{\text{Hyb1}} = \text{RMSE}(F_{i\alpha}^{\text{Hyb1}} - F_{i\alpha}^{\text{ACE}})$ of rescaled atoms, including the theoretical rescaling force according to Eq. (F1), of the adaptive-precision potential Hyb1 compared to ACE forces during the nanoindentation is shown in Fig. 3(a). The force difference ΔF^{Hyb1} is calculated from force components $F_{i\alpha}$ in the spatial dimension α . For defects and surface atoms, the forces are within a tolerance of up to 10 meV/Å compared to the precise ACE forces for most cases, while the exact potential energy of ACE is used according to Eq. (1).

We changed single parameters of the parameter set Hyb1 and performed simulations with otherwise identical parameters to perform a sensitivity analysis on the results of the selected parameters. The influence of the changed parameters is shown in Fig. 3(b). Increasing the thresholds CSP_{lo} and CSP_{hi} by 0.5 Å^2 decreases the

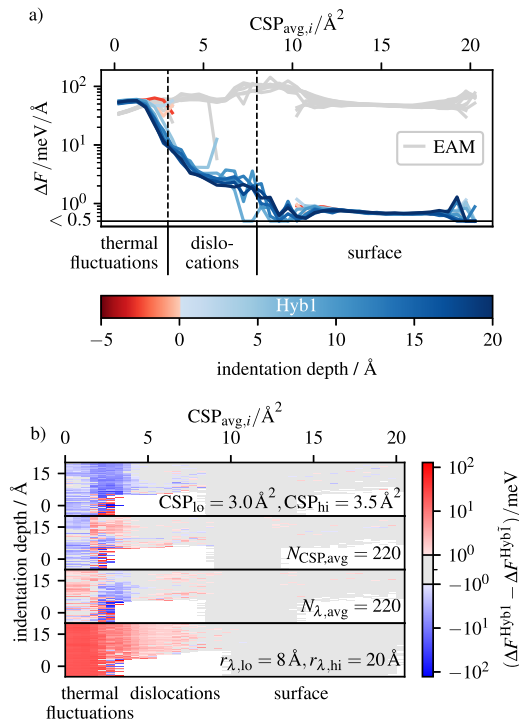


FIG. 3. (a) Force error $\Delta F^{\text{Hyb1}} = \text{RMSE}(F_{\text{Hyb1}}^{\text{Hyb1}} - F_{\text{ACE}}^{\text{ACE}})$ of rescaled atoms dependent on $\text{CSP}_{\text{avg},i}$ and indentation depth of a nanoindentation. (b) Force error ΔF^{Hyb1} compared with the force error $\Delta F^{\text{Hyb1}\hat{}}$ of another simulation with the parameter set Hyb1, whereas the parameters in the lower right corner of the corresponding plot are different than in Hyb1. The force error difference $\Delta F^{\text{Hyb1}} - \Delta F^{\text{Hyb1}\hat{}}$ is color-coded, whereas blue and red colors correspond to a smaller force error of Hyb1 and Hyb1, respectively. The simulations for (a) and the lowest plot of (b) are calculated in a cubic system with 100^3 unit cells; 68^3 unit cells are used for the remaining simulations.

force precision for thermally fluctuating atoms and dislocations with $\text{CSP}_{\text{avg},i} < 4.5 \text{\AA}^2$. The decreased force precision for the dislocations is expected as these are the atoms that are simulated with the fast potential due to the increased CSP thresholds. Thus, the only advantage of increasing the CSP thresholds is performance. Doubling the averaging time of the centro-symmetry parameter $N_{\text{CSP,avg}}$ decreases the force precision on thermally fluctuating atoms. The force precision of dislocations increases in most snapshots. As dislocations are in contrast to the surface, not stationary, increasing the CSP averaging time may reduce the detection possibility for moving dislocations. Thus, one can adjust $N_{\text{CSP,avg}}$ dependent on the expected dislocation dynamics. Doubling the averaging time of the switching parameter $N_{\lambda,\text{avg}}$ decreases the force precision of dislocations with $\text{CSP}_{\text{avg},i} < 4 \text{\AA}^2$, but increases the force precision of dislocations with a larger time-averaged CSP. The force precision for dislocations decreases with an increasing $\text{CSP}_{\text{avg},i}$ according to Fig. 3(a) and this effect is enhanced by doubling $N_{\lambda,\text{avg}}$. Thus, increasing $N_{\lambda,\text{avg}}$ is not beneficial. Increasing the cutoffs $r_{\lambda,\text{lo}}$ and $r_{\lambda,\text{hi}}$ of $\lambda_{\text{min},i}$ by 4 and 8 $\text{\AA}$, respectively, increases the number of precise calculations and, thus, the force precision for all atoms apart from surface atoms, which

are already treated with the precise potential. The only disadvantage is the increased compute time due to more precise calculations. Therefore, the cutoffs of $\lambda_{\text{min},i}$ can be adjusted depending on the performance requirements.

B. Computational efficiency

The processor p with the highest work τ_p^{max} restricts the speed of a simulation since faster processors need to wait due to communication and synchronization. Thus, the imbalance, as defined in LAMMPS,⁵⁵

$$I = \tau_p^{\text{max}} / \langle \tau_p \rangle_p \quad (13)$$

of the force calculation time τ_p gives a measure for the quality of the load balancing, whereas an imbalance of 1 corresponds to a perfectly balanced system. The mean imbalance in the nanoindentation is 1.41 with a standard deviation of 0.26. Dynamic load-balancing of an adaptive-precision potential is challenging as the work distribution changes drastically when the precision of atoms is switched from fast to precise or vice versa, as discussed in Appendix J in more detail. The average work per atom for the four sub-processes is $\langle \rho_{p,t}^{\text{FP}} \rangle_{p,t} = 9.6 \mu\text{s}$, $\langle \rho_{p,t}^{\text{PP}} \rangle_{p,t} = 1.7 \text{ms}$, $\langle \rho_{p,t}^{\lambda} \rangle_{p,t} = 48.6 \mu\text{s}$, and $\langle \rho_{p,t}^{\text{CSP}} \rangle_{p,t} = 5.6 \mu\text{s}$, whereas the processor-dependency is discussed in Appendix I.

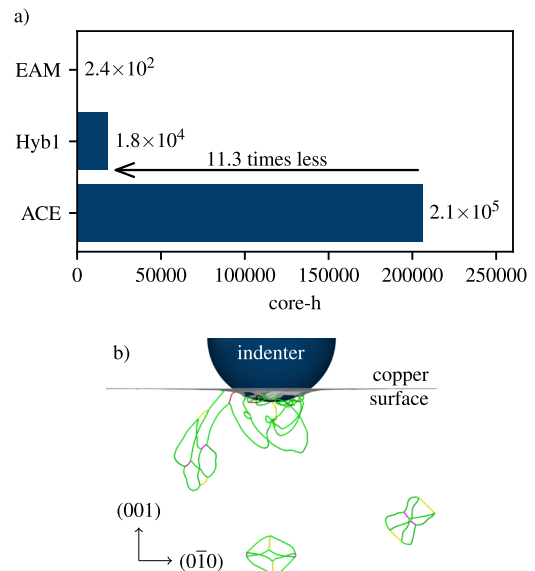


FIG. 4. (a) Total computation time of a nanoindentation with 4×10^6 atoms (100^3 unit cells) simulated for 100 ps with the adaptive-precision potential Hyb1 compared to ACE and EAM nanoindentations. EAM, hybrid, and ACE simulations are calculated on 128, 384, and 2048 cores of JURECA-DC,⁵⁴ respectively. (b) Visualization of the dislocation line defects that occur in a nanoindentation. The visualization is done with OVITO,⁵⁶ and the dislocations are identified by the dislocation analysis⁵⁷ of OVITO.

The motivation to use an adaptive-precision potential is to save computation time compared to a precise potential while preserving accuracy in regions of the simulation where it is required. The nanoindentation with 4×10^6 atoms calculated for 100 ps on JURECA-DC⁵⁴ with the adaptive-precision potential Hyb1 (cf. Table IV) shows a speedup of 11.3 compared to a full ACE simulation as visualized in Fig. 4(a). We calculated the surface and dislocations, which developed during the nanoindentation as visualized in Fig. 4(b), precisely with the ACE-potential and saved computation time for the remaining atoms by using the fast EAM-potential. Note that the amount of saved computation time strongly depends on the number of not required evaluations of the precise and expensive potential, as discussed in Appendix K, using the equilibration of the nanoindentation as an example.

V. CONCLUSION

We introduced a method that allows us to compute adaptive-precision interatomic potentials to speedup large-scale atomistic simulations with a region or atoms of special interest. The atoms of special interest are automatically detected by a customizable detection mechanism and simulated with the precise potential, while the fast potential is used for the remaining atoms. The energy model [Eq. (1)] ensures precise energies for the detected atoms, while the switching function (cf. Sec. II B) is used to get precise forces. We presented a load-balancing method with subroutine specific per-atom works per processor using a staggered grid to prevent the load-balancing issues arising from combining two potentials of different costs. We demonstrated the presented adaptive-precision method by creating the copper potential Hyb1 (Table IV) from an EAM and an ACE potential. We explained how the fast EAM potential is optimized to match bulk properties with the precise ACE potential.

A nanoindentation with 4×10^6 atoms calculated for 100 ps was used to demonstrate the capabilities of the method. The achieved accuracy of Hyb1 for the detected atoms of interest during the nanoindentation is for forces 10 meV/Å, while the exact potential energies of the ACE potential are obtained by design. The nanoindentation showed a speedup of 11.3 compared to a full ACE simulation.

Note that the amount of computation time that can be saved heavily depends on the number of not required precise calculations and, therefore, on the simulation setup and the detection mechanism for atoms of interest. Hence, the expected speedup cannot be generalized. Our adaptive-precision method includes overhead in the form of the detection mechanism and the calculation of the switching parameter. Therefore, in both limits, when all or no atoms require a precise calculation, using an adaptive-precision potential is not beneficial. Between both limits, however, there will be a range in the fraction of precise atoms, where the adaptive-precision potential becomes beneficial, as demonstrated with the nanoindentations in which the majority of precise calculations could be accelerated. Since the force-calculation time of the precise potential is measured to enable dynamic load-balancing, one can, as discussed in Sec. II D, decide on-the-fly whether or not to apply the adaptive precision, i.e., whether the overhead dominates the computational benefit. In this case, it is a simple switch to compute all atoms in the simulation via the accurate potential.

The force on an atom depends, according to Eq. (2), on the switching parameters of all neighboring atoms. Therefore, we use a spatial switching-zone to ensure a smooth transition of the potential from the precisely to the fast calculated atoms. The size of this spatial switching zone, 8 or 12 Å in this paper, should be small in comparison to the system size. Otherwise, a single atom that is identified for a precise potential description would cause the computation of a majority of the system atoms with the precise potential since atoms in the transition zone require the computation of both the precise and fast potential in order to be weighted according to Eqs. (1) and (2). As mentioned earlier, an optimization of compute time would then suggest applying the precise potential for the complete system in order to reduce or avoid computational overhead. This case study shows that an adaptive-precision potential is best suited for large-scale atomistic simulations where a decision can be made about which atoms require a high precision description. In many simulations, a large speedup can be achieved then by skipping most of the precise calculations. Furthermore, an adaptive-precision potential allows us to embed precisely calculated atoms of interest at low cost into millions of quickly calculated atoms and thereby allows larger simulations, for which nanoindentations serve here as an example. The dislocation loops nucleate below the indenter and glide away from there [cf. Fig. 4(b)]. This type of system, considered on a longer time scale, definitely requires a very big system since dislocation loops evolve in each spatial dimension and should not artificially interact with themselves via periodic boundary conditions, which is quite likely if the system size is too small.

As discussed, our method needs a detection mechanism for atoms of interest that need to be calculated precisely. We selected the centro-symmetry parameter (CSP) as detection mechanism as we were interested in defects and surface atoms for the nanoindentation demonstration. However, one can replace this detection mechanism easily depending on one's own needs, as discussed in Sec. II B, where we suggest detection mechanisms for grain-boundary structures, coatings, interfaces, and cracks, among others. However, our proposed dynamical adjustment of the group of atoms is not appropriate for spontaneous phenomena that cannot be anticipated by the detection mechanism, as one cannot ensure a smooth transition from the fast to the precise potential, which would result in artifacts.

The application of an adaptive-precision potential for other situations where the highest precision is required only locally but not globally, such as interfaces, cracks, and grain boundaries, is straightforward, while the generalization of the detection mechanism for non-crystalline systems such as amorphous solids is a natural next question. Automated training of the fast potential would improve the usability of adaptive-precision potentials. The present work is based on a CPU-version of our method. A GPU-version is planned for the future.

ACKNOWLEDGMENTS

We would like to thank Y. Lysogorskiy for the help with the ML-PACE package in LAMMPS and for the valuable comments on the integration of the switching parameter within the ACE calculation. The authors gratefully acknowledge the computing time on

the supercomputer JURECA⁵⁴ at Forschungszentrum Jülich under Grant No. 28990 (hybridace).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

David Immel: Formal analysis (lead); Investigation (lead); Methodology (equal); Resources (equal); Software (lead); Validation (lead); Visualization (lead); Writing – original draft (lead). **Ralf Drautz:** Conceptualization (equal); Methodology (equal); Supervision (equal); Writing – review & editing (equal). **Godehard Sutmann:** Conceptualization (equal); Methodology (equal); Resources (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this study are available from D.I. upon reasonable request.

APPENDIX A: CENTRO-SYMMETRY PARAMETER

To calculate the CSP according to Ref. 29, initially reference vectors $\vec{r}_{i,j,\text{ref}}$ from the central atom i to its closest neighboring atoms j are determined in an undistorted lattice. In the distorted lattice, one uses the distorted vectors $\vec{r}_{i,j}$ closest in distance to the undistorted reference vectors to calculate the centro-symmetry parameter $\text{CSP}_i^{\text{ref}} = \sum_{j=1}^{N/2} (\vec{r}_{i,j} + \vec{r}_{i,j+N/2})^2$, whereas j and $j + N/2$ correspond to a pair of opposite nearest neighbors of the central atom i . The set of distorted vectors used to calculate $\text{CSP}_i^{\text{ref}}$ may contain a neighbor multiple times and non-nearest neighbors.

LAMMPS³³ uses a different definition for the centro-symmetry parameter CSP^{imp} , which does not require a reference system. At first, one searches the set $\mathcal{N}_i$ of the N nearest neighboring atoms of the central atom i , whereas N is the number of nearest neighbors in the undistorted lattice. Afterward, the $N/2$ pairs with the smallest $|\vec{r}_{i,j_1} + \vec{r}_{i,j_2}|$ with $j_1, j_2 \in \mathcal{N}_i$ are used to calculate the centro-symmetry parameter. This definition may also contain duplicates,

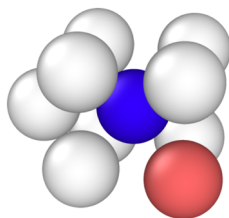


FIG. 5. Snapshot of BCC tungsten at 300 K. The eight nearest neighbors (white) of the central atom (blue) do not contain all neighboring atoms required to calculate the centro-symmetry parameter CSP^{ref} . The ninth-nearest neighbor (red) is missing. Thus, one needs to evaluate more than the 8 nearest neighbors for the CSP^{imp} -calculation.

but relevant non-nearest neighbors may be neglected. This neglect of non-nearest neighbors is relevant for the computed CSP^{imp} of tungsten in a BCC lattice at 300 K, as shown in Fig. 5. The effect of this neglect is reduced by searching $N + N_{\text{buffer}}$ rather than N nearest neighbors for $\mathcal{N}_i$. Thus, we calculate CSP^{imp} according to the LAMMPS-definition but use $N + N_{\text{buffer}}$ atoms for the identification of the $N/2$ relevant opposite-neighbor pairs denoted as $j, j + N/2$.

The switching parameter needs to detect atoms for the precise calculation precisely to prevent too many evaluations of the precise and expensive potential. This requirement is fulfilled by the CSP since only neighboring atoms similar to the expected neighbors in an undistorted lattice are used for the calculation. Missing or displaced atoms in the second neighbor-shell of the central atom do not influence the CSP. Hence, the nearest neighbor-shell of a defect, vacancy, or surface atom is detected, but the second-nearest neighbor-shell is not. The latter behavior might not be useful when low angle symmetric tilt grain boundaries, for example, are of interest, as only a subset of atoms may be detected.⁵⁸ However, one can use the CSP to detect some grain boundaries.⁵⁹

APPENDIX B: CONSTRUCTION OF THE SWITCHING FUNCTION

The centro-symmetry parameter, according to Eq. (3), detects defects and surface atoms, which need to be calculated by the precise potential. Therefore, we use the CSP as starting point to construct the switching function for a nanoindentation. The centro-symmetry parameter fluctuates over time as atoms fluctuate, dependent on their temperature. Since the thermal fluctuations of atoms should not be detected, we introduce a moving average,

$$\text{CSP}_{\text{avg},i}(t) = \frac{1}{N_{\text{CSP,avg}}} \sum_{n=1}^{N_{\text{CSP,avg}}} \text{CSP}_i(t - n\Delta t), \quad (\text{B1})$$

of $N_{\text{CSP,avg}}$ time steps Δt . The time-averaged centro-symmetry parameter can be used to calculate a switching parameter $\lambda_{0,i}$ as

$$\lambda_{0,i}(t) = f^{(\text{cut})} \left(\frac{\text{CSP}_{\text{avg},i}(t) - \text{CSP}_{\text{lo}}}{\text{CSP}_{\text{hi}} - \text{CSP}_{\text{lo}}} \right), \quad (\text{B2})$$

where we use the radial function,

$$f^{(\text{cut})}(x) = \begin{cases} 1 & \text{for } x \leq 0, \\ 0.5 + 0.9375y - 0.625y^3 & \text{for } 0 < x < 1, \\ +0.1875y^5 \text{ with } y = 1 - 2x & \\ 0 & \text{for } 1 \leq x. \end{cases} \quad (\text{B3})$$

$f^{(\text{cut})}$ is taken from Ref. 60 as the first and second derivatives of $f^{(\text{cut})}$ are smooth at 0 and 1. $\lambda_{0,i}(t) = 1$ applies for all atoms with $\text{CSP}_{\text{avg},i}(t) \leq \text{CSP}_{\text{lo}}$. The use of a continuous switching function for $\text{CSP}_{\text{avg},i} \in [\text{CSP}_{\text{lo}}, \text{CSP}_{\text{hi}}]$ ensures a smooth transition from the fast interatomic potential to the precise one and vice versa for all atoms.

One can change the argument of $f^{(\text{cut})}$ in Eq. (B2) to use a different detection mechanism and use the following definitions of the switching function as presented without further changes. The argument must be 0 for atoms to be calculated exactly and 1 for atoms to be calculated quickly.

Since the force on an atom depends on all switching parameters within its force cutoff, as discussed in Sec. II A, we introduce

$$\lambda_{\min,i}(t) = \min \left(\left\{ 1 - (1 - \lambda_{0,j}(t)) f^{(\text{cut})} \left(\frac{r_{ij}(t) - r_{\lambda,\text{lo}}}{r_{\lambda,\text{hi}} - r_{\lambda,\text{lo}}} \right) : j \in \Omega_{\lambda,i} \right\} \right), \quad (\text{B4})$$

where $\Omega_{\lambda,i} = \{j : r_{ij} < r_{\lambda,\text{hi}}\}$ is the set of neighboring atoms. As $\lambda_{0,i} \in [0, 1]$ applies, $\lambda_{\min,i} \in [0, 1]$ follows. For both the calculation and understanding of $\lambda_{\min,i}(t)$, it is useful to start with $\lambda_{0,i}(t)$ as initial value for the minimum search. Atoms j that were not detected for a precise calculation have $\lambda_{0,j}(t) = 1$. Their contribution to the minimum search is 1 independent of r_{ij} and, thus, negligible. Only neighbors j that were detected for a precise calculation and, thus, have $\lambda_{0,j}(t) < 1$ and $\lambda_{0,i}(t)$ can be the searched minimum. Atoms with $\lambda_{0,i}(t) = 0$ set $\lambda_{\min,i}(t) = 0$ for all neighboring atoms j with $r_{ij} \leq r_{\lambda,\text{lo}}$ and may set $\lambda_{\min,i} \in (0, 1)$ for neighboring atoms with $r_{ij} \in (r_{\lambda,\text{lo}}, r_{\lambda,\text{hi}})$. Thus, Eq. (B4) decreases the switching parameter for neighbors of atoms, which needs to be calculated precisely to ensure a higher fraction of precise force contributions for these atoms. Since the radial function $f^{(\text{cut})}(r_{ij})$ used in Eq. (B4) also fluctuates due to atomic fluctuations, we use a moving average,

$$\lambda_{\text{avg},i}(t) = \frac{1}{N_{\lambda,\text{avg}}} \sum_{n=1}^{N_{\lambda,\text{avg}}} \lambda_{\min,i}(t - n\Delta t), \quad (\text{B5})$$

of $N_{\lambda,\text{avg}}$ time steps. To prevent unnecessary fluctuations of the switching function further, we introduce a minimum step size $\Delta\lambda_{\min}$ and neglect changes smaller than this step size in the form of

$$\lambda_{\text{avg},\text{msz},i}(t) = \begin{cases} \lambda_{\text{avg},i}(t) & \text{for } |\lambda_{\text{avg},i}(t) - \lambda_{\text{avg},\text{msz},i}(t - \Delta t)| \\ & \geq \Delta\lambda_{\min} \text{ or } \lambda_{\text{avg},i}(t) \in \{0, 1\}, \\ \lambda_{\text{avg},\text{msz},i}(t - \Delta t), & \text{otherwise.} \end{cases} \quad (\text{B6})$$

When an atom is detected for a precise calculation according to $\lambda_{\min,i}(t)$, it takes $N_{\lambda,\text{avg}}$ time steps until $\lambda_{\text{avg},i}$ is changed completely due to the time average. Therefore, a change of $\lambda_{\text{avg},i}$ in the order of magnitude of $1/N_{\lambda,\text{avg}}$ is expected. Changes of $\lambda_{\text{avg},\text{msz},i}$ smaller than $1/N_{\lambda,\text{avg}}$ can be neglected without disturbing the smooth transition from the fast interatomic potential to the precise one or vice versa in a relevant way. Thus, the use of an appropriately small $\Delta\lambda_{\min}$ prevents only λ changes due to atomic fluctuations. We use $\lambda_{\text{avg},\text{msz},i}$ according to Eq. (B6) as switching function for energy- and force-calculations as the CSP precisely detects defects and surface atoms, and the switching function ensures precise forces on these detected atoms.

APPENDIX C: LOCAL THERMOSTAT

The local thermostat needs to change the kinetic energy of particles in order to correct the energy error ΔH_i according to Eq. (5) introduced by changed switching parameters and thereby conserve the energy.

1. Error calculation

ΔH_i is the energy difference of atom i at the end of a time step between the conservative reference system with constant switching parameters and the system with updated switching parameters. Therefore, one has to finish the calculation of an integration step for both sets of switching parameters. The quantities of the conservative reference system calculated with the constant switching parameter $\lambda_i^c(t + \Delta t) = \lambda_i(t)$ are denoted with $\dots^c$. In contrast, the quantities calculated with the dynamically updated switching parameter $\lambda_i^d(t + \Delta t)$ are denoted with $\dots^d$. The non-energy conserving force on an atom with the updated switching parameter under neglect of $\nabla_i \lambda_k$ in Eq. (2) is

$$F_i^d = \sum_k \left(-\lambda_k^d (\nabla_i E_k^{(\text{f})}) - (1 - \lambda_k^d) (\nabla_i E_k^{(\text{p})}) \right). \quad (\text{C1})$$

With the same force equation like Eq. (C1) but λ_k^c instead of λ_k^d , we get the conservative reference force F_i^c . Since the form of the equation is the same and both forces depend on the same atomic positions $x_i(t + \Delta t)$, one can easily calculate F_i^d and F_i^c during the force calculation by just summing up two forces weighted with the corresponding switching parameter.

To measure the violation of energy conservation [cf. Eq. (5)] by neglecting $\nabla_i \lambda_k$ and using F_i^d according to Eq. (C1), we calculate both the potential energy difference $\Delta E_i(t + \Delta t) = E_i^c(t + \Delta t) - E_i^d(t + \Delta t)$ and the kinetic energy difference $\Delta T_i(t + \Delta t) = (|\vec{v}_i^c(t + \Delta t)|^2 - |\vec{v}_i^d(t + \Delta t)|^2) m_i / 2$ at the end of the time step. The potential energy difference is given as

$$\Delta E_i(t + \Delta t) = \left(\lambda_i^c(t + \Delta t) - \lambda_i^d(t + \Delta t) \right) \times \left(E_i^{(\text{f})}(t + \Delta t) - E_i^{(\text{p})}(t + \Delta t) \right). \quad (\text{C2})$$

One should note that Eq. (C2) only uses already calculated potential energies and does not introduce further overhead. The switching parameter is known for all atoms, and one does not need any potential energies for atoms i with $\lambda_i^c(t + \Delta t) = \lambda_i^d(t + \Delta t)$. The kinetic energy difference is given as

$$\Delta T_i(t + \Delta t) = \sum_{\alpha=1}^3 \left(\frac{(\Delta t)^2}{8m_i} \left((F_{i\alpha}^c(t + \Delta t))^2 - (F_{i\alpha}^d(t + \Delta t))^2 \right) + \left(F_{i\alpha}^c(t + \Delta t) - F_{i\alpha}^d(t + \Delta t) \right) \times v_{i\alpha} \left(t + \frac{\Delta t}{2} \right) \frac{\Delta t}{2} \right), \quad (\text{C3})$$

whereas α denotes the spatial component of a vector. It is important to use only the forces of the interatomic potential as F_i^c and F_i^d and to neglect any present external forces since such additional forces may not be energy- and momentum-conserving.

2. Error correction

Coupling particles to a local heat bath through collisions of two particles, according to Lowe-Andersen,⁶¹ for example, is, therefore, impractical since the direction of the momentum change of the two particles is in the direction of the relative distance of both particles. This predefined direction of a momentum change

for two particles may result in very high and unphysical forces when the relative momentum of the two considered particles is not parallel to the direction of the momentum change. Thus, one needs to apply a correction to the set Ω_i of atoms rather than to a pair of atoms to distribute the additional force over multiple atoms. We get a minimum additional force when a particle's momentum is rescaled since the direction of force and momentum change is equal in this case, but this would violate momentum conservation. Thus, we need to rescale with respect to the center-of-mass velocity $\vec{v}_{\text{cm}}(\Omega_i) = (\sum_{j \in \Omega_i} m_j \vec{v}_j) / (\sum_{j \in \Omega_i} m_j)$ of the rescaled atoms. We want to rescale only the relative momentum $\vec{p}_{\text{rel},j}(\Omega_i) = \vec{p}_j(\Omega_i) - m_j \vec{v}_{\text{cm}}(\Omega_i)$ of all particles $j \in \Omega_i$ by the factor $\beta(\Omega_i)$, namely,

$$\vec{p}_{j,\text{resc}}(\Omega_i) = m_j \vec{v}_{\text{cm}}(\Omega_i) + \beta(\Omega_i) \vec{p}_{\text{rel},j}(\Omega_i). \quad (\text{C4})$$

The total relative momentum $\sum_{j \in \Omega_i} \vec{p}_{\text{rel},j} = \vec{0}$ vanishes, and the rescaling, according to Eq. (C4), therefore conserves the momentum. The kinetic energy change in Ω_i due to the rescaling is

$$\Delta T_{\text{resc}}(\Omega_i) = (\beta(\Omega_i)^2 - 1) \sum_{j \in \Omega_i} \frac{1}{2m_j} p_{\text{rel},j}(\Omega_i)^2. \quad (\text{C5})$$

To correct the detected energy error ΔH_i according to Eq. (5), we need to select $\beta(\Omega_i)$ to satisfy $\Delta H_i = \Delta T_{\text{resc}}(\Omega_i)$, namely,

$$\beta(\Omega_i) = \sqrt{\frac{\Delta H_i}{\sum_{j \in \Omega_i} \frac{1}{2m_j} p_{\text{rel},j}(\Omega_i)^2} + 1}. \quad (\text{C6})$$

We neglect $\beta(\Omega_i) = -\sqrt{\dots}$ although it fulfills $\Delta H_i = \Delta T_{\text{resc}}(\Omega_i)$, because it alters the momentum direction for $\Delta H_i = 0$. Rescaling, according to Eq. (C6), conserves the momentum for $\Delta H_i = 0$. It is important to note that the rescaling factor β depends on the momentum before the rescaling. Therefore, our method presents challenges in terms of parallelization, as one needs a sub-domain decomposition with sub-domains that can be treated independently. We did not develop and implement such a sub-domain decomposition as it is not the focus of our work. Instead, we only rescale locally administered particles and no ghost particles. We use a random selection of locally administered particles of the neighbor list of particle i and the particle i itself as Ω_i . Thereby, the energy correction by rescaling relative momenta can be applied per processor without communication for all particles with an energy error of $\Delta H_i \neq 0$. Note that the application of this local thermostat contains essentially two stochastic components. First, the selection of neighboring atoms that are rescaled is stochastic, namely, Ω_i . Second, the direction of the effective force is the direction of the relative momentum $\vec{p}_{\text{rel},j}(\Omega_i)$ [cf. Eq. (C4)], and thus stochastic.

The maximum possible decrease in the relative momentum per particle is to decrease the relative momentum for all particles $j \in \Omega_i$ to 0 kgm/s with $\beta(\Omega_i) = 0$. Thus, according to Eq. (C6), energy errors $\Delta H_i < -\sum_{j \in \Omega_i} p_{\text{rel},j}(\Omega_i)^2 / (2m_j)$ cannot be corrected as they correspond to a negative radicand in Eq. (C6) and imply $\beta \notin \mathbb{R}$. Hence, one needs to include more particles in the rescaling in this case. Concrete prevention strategies and workarounds in LAMMPS for negative radicands are discussed in Appendix D. However, the occurrence of negative radicands depends on the

simulation setup and the combined potentials and is, by our observation, an extremely rare event that does not occur at all in many simulations.

The presented rescaling approach fulfills momentum conservation as only relative momenta are rescaled and results in energy conservation for the forces according to Eq. (C1). One can then use $F_i^d(t + \Delta t)$ as $F_i(t + \Delta t)$ and the rescaled velocities as $v_i(t + \Delta t)$ at the beginning of the next integration step.

This local thermostat is a tool to correct locally a known energy gain or loss of an atom i during the integration step by adjusting the kinetic energy in the local environment of this atom. We know exactly the energy error due to the neglect of $\nabla_i \lambda_k$ in Eq. (2) and correct this error to conserve the energy within numerical precision.

APPENDIX D: HANDLING NEGATIVE RADICANDS IN LAMMPS

Energy errors $\Delta H_i < -\sum_{j \in \Omega_i} p_{\text{rel},j}(\Omega_i)^2 / (2m_j)$ cannot be corrected by the local thermostat as they correspond to a negative radicand in Eq. (C6) and imply $\beta \notin \mathbb{R}$. Hence, one needs a larger neighbor list to randomly draw atoms as Ω_i in case of a negative radicand. As a dynamic neighbor-list cutoff is not implemented in LAMMPS, one needs to use the last restart file to run the simulation with a larger neighbor-list cutoff. Requesting a larger neighbor-list cutoff as a precautionary measure is possible, but it would reduce the performance of the whole simulation rather than just of a small subset of time steps. As we use a parallel-computing approach, including a load-balancing scheme, problems due to the domain geometry are possible. When the load balancer reduces the size of a domain, it also reduces the number of local particles that are available for rescaling. Thus, in case of a negative radicand, one can also restart the simulation with temporarily disabled load balancing to avoid the existence of small domains. For details regarding the load balancing, see Sec. II D.

APPENDIX E: OPTIMIZING THE EAM POTENTIAL

A EAM potential is given according to Eq. (11) by an embedding function ξ , an electron density ζ , and a pair potential Φ . We use a copper potential from Ref. 28, which is given by the pair potential,

$$\Phi(x) = \left(E_1 M(x, r_0^{(1)}, \alpha_1) + E_2 M(x, r_0^{(2)}, \alpha_2) + \delta \right) \times \psi\left(\frac{x - r_c}{h}\right) - \sum_{n=1}^3 \Theta(r_s^{(n)} - x) S_n(r_s^{(n)} - x)^4, \quad (\text{E1})$$

where

$$\psi(x) = x^4 / (1 + x^4) \Theta(-x) \quad (\text{E2})$$

and

$$M(x, r_0, \alpha) = \exp(-2\alpha(x - r_0)) - 2 \exp(-\alpha(x - r_0)) \quad (\text{E3})$$

TABLE III. Optimized parameters of the EAM Copper potential compared with the original values of EAM2 of Ref. 28.

Parameter	Value original	Value fit 300K	Parameter	Value original	Value fit 300 K
E_1/eV	$2.01\,458 \times 10^2$	$2.01\,486 \times 10^2$	$\delta/\text{\AA}$	$8.62\,250 \times 10^{-3}$	$2.18\,016 \times 10^{-2}$
E_2/eV	$6.59\,288 \times 10^{-3}$	$6.44\,057 \times 10^{-3}$	$F_0^{\text{Fit}}/\text{eV}$	$0.00\,000 \times 10^0$	$-1.50\,036 \times 10^{-1}$
$\xi^{(0)}/\text{eV}$	$-2.30\,000 \times 10^0$	$-2.30\,089 \times 10^0$	$h/\text{\AA}$	$5.00\,370 \times 10^{-1}$	$5.00\,370 \times 10^{-1}$
$\xi^{(2)}/\text{eV}$	$1.40\,000 \times 10^0$	$1.39\,912 \times 10^0$	q_1/eV	$-1.30\,000 \times 10^0$	$-1.29\,894 \times 10^0$
Q_1	$4.00\,000 \times 10^{-1}$	$3.99\,178 \times 10^{-1}$	q_2/eV	$-9.00\,000 \times 10^{-1}$	$-8.99\,672 \times 10^{-1}$
Q_2	$3.00\,000 \times 10^{-1}$	$3.00\,455 \times 10^{-1}$	q_3/eV	$1.80\,000 \times 10^0$	$1.79\,940 \times 10^0$
$S_1/\text{eV}\text{\AA}^{-4}$	$4.00\,000 \times 10^0$	$4.00\,000 \times 10^0$	q_4/eV	$3.00\,000 \times 10^0$	$2.99\,944 \times 10^0$
$S_2/\text{eV}\text{\AA}^{-4}$	$4.00\,000 \times 10^1$	$4.00\,000 \times 10^1$	$r_0^{(1)}/\text{\AA}$	$8.35\,910 \times 10^{-1}$	$8.36\,347 \times 10^{-1}$
$S_3/\text{eV}\text{\AA}^{-4}$	$1.15\,000 \times 10^3$	$1.15\,000 \times 10^3$	$r_0^{(2)}/\text{\AA}$	$4.46\,867 \times 10^0$	$4.46\,690 \times 10^0$
a	$3.80\,362 \times 10^0$	$3.80\,541 \times 10^0$	$r_0^{(3)}/\text{\AA}$	$-2.19\,885 \times 10^0$	$-2.19\,834 \times 10^0$
$\alpha_1/\text{\AA}^{-1}$	$2.97\,758 \times 10^0$	$2.96\,972 \times 10^0$	$r_0^{(4)}/\text{\AA}$	$-2.61\,984 \times 10^2$	$-2.61\,984 \times 10^2$
$\alpha_2/\text{\AA}^{-1}$	$1.54\,927 \times 10^0$	$1.55\,510 \times 10^0$	$r_s^{(1)}/\text{\AA}$	$2.24\,000 \times 10^0$	$2.24\,000 \times 10^0$
$\beta_1/\text{\AA}^{-2}$	$1.73\,940 \times 10^{-1}$	$1.66\,589 \times 10^{-1}$	$r_s^{(2)}/\text{\AA}$	$1.80\,000 \times 10^0$	$1.80\,000 \times 10^0$
$\beta_2/\text{\AA}^{-1}$	$5.35\,661 \times 10^2$	$5.35\,661 \times 10^2$	$r_s^{(3)}/\text{\AA}$	$1.20\,000 \times 10^0$	$1.20\,000 \times 10^0$
$r_c/\text{\AA}$	$5.50\,679 \times 10^0$	$5.50\,679 \times 10^0$			

with the Heaviside step function $\Theta(x) = 1$ for $x \geq 0$, 0 for $x < 0$. The embedding function is

$$\xi(x) = \begin{cases} \xi^{(0)} + \frac{1}{2}\xi^{(2)}(x-1)^2 & \text{if } x \leq 1, \\ \xi^{(0)} + \frac{1}{2}\xi^{(2)}(x-1)^2 + q_1(x-1)^3 & \text{otherwise.} \end{cases} \quad (\text{E4})$$

The electron density is

$$\zeta(x) = \left(a \exp\left(-\beta_1(x-r_0^{(3)})^2\right) + \exp\left(-\beta_2(x-r_0^{(4)})\right) \right) \times \psi((x-r_c)/h). \quad (\text{E5})$$

The EAM potential is optimized as described in Sec. III with atom-icrex. The target values and tolerances for the minimized loss function (12) are shown in Table I. The fitted parameters are shown in Table III.

APPENDIX F: PARAMETER SELECTION FOR AN ADAPTIVE-PRECISION POTENTIAL

The parameters of the hybrid potential are listed in Table IV and will be explained in more detail in this appendix.

1. Centro-symmetry parameter

N_{buffer} is used to ensure all relevant neighboring atoms are used in the calculation of the centro-symmetry parameter according to Eq. (3). For copper, we have not observed the case of unexpected high centro-symmetry parameters we discussed in Fig. 5 for tungsten. Therefore, we use $N_{\text{buffer}} = 0$. An unexpected high CSP is less likely for copper as the distance in lattice constants between the first

TABLE IV. Parameters of the adaptive-precision model and values of the parameter set Hyb1 for copper at 300 K.

Parameter	Value	Parameter	Value	Parameter	Value
N_{buffer}	0 atoms	CSP_{hi}	$3.0\,\text{\AA}^2$	$N_{\lambda,\text{avg}}$	110
$N_{\text{CSP,avg}}$	110	$r_{\lambda,\text{lo}}$	$4.0\,\text{\AA}$	$\Delta\lambda_{\text{min}}$	$1.0/N_{\lambda,\text{avg}}$
CSP_{lo}	$2.5\,\text{\AA}^2$	$r_{\lambda,\text{hi}}$	$12.0\,\text{\AA}$	$ \Omega_i $	800 atoms

and second neighbor shell is for BCC with $1 - \sqrt{3/4} \approx 0.13$ smaller than for FCC with $1 - \sqrt{1/2} \approx 0.29$.

The centro-symmetry parameter of an atom i fluctuates due to atomic fluctuations. To suppress these fluctuations, we use a time average of $N_{\text{CSP,avg}}$ time steps in Eq. (B1). To determine $N_{\text{CSP,avg}}$, we measured autocorrelation functions (acf) for 5 ps near a (100)-copper surface, as shown in Fig. 6, as a mean per (100) atom layer dependent on the distance to the surface. The acf of the CSP has a peak at about 110 fs for all layers apart from the surface layer. As the CSP depends on the local environment, a different acf for the surface atoms is expected. Since the time averaging of the centro-symmetry parameter is used to average out thermal fluctuations, we want to average the CSP up to the peak of the acf of non-surface atoms at 110 fs. With the used time step $\Delta t = 1$ fs, $N_{\text{CSP,avg}} = 110$ follows.

The distribution of the centro-symmetry parameter CSP_i in the training data is used to set the thresholds CSP_{lo} and CSP_{hi} for Eq. (B2). The fast potential is optimized for usage at a target temperature with MD data of the precise and expensive potential. The training data contain only thermal fluctuations of atoms with observed $\text{CSP}_i \in [0\,\text{\AA}^2, 3\,\text{\AA}^2]$. Therefore, we distinguish between a small CSP of the thermal fluctuations and all other atoms with a higher CSP. As atoms with $\text{CSP}_{\text{avg},i} \geq \text{CSP}_{\text{hi}}$ are detected for a precise calculation according to Eq. (B2), we select $\text{CSP}_{\text{hi}} = 3\,\text{\AA}^2$.

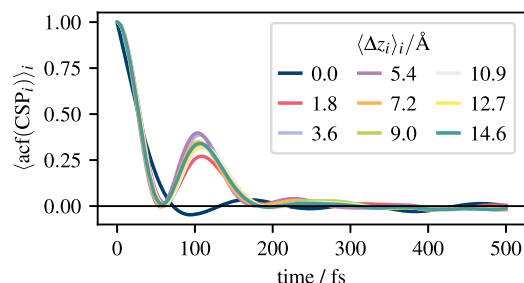


FIG. 6. Mean autocorrelation functions of the centro-symmetry parameter CSP_i according to Eq. (3). Each shown graph is the mean of the autocorrelation functions (acf) of one atom layer in the z -direction. The mean distance $\langle \Delta z_i \rangle_i$ of the corresponding atom layer to the (100)-copper surface is given in the legend. The autocorrelation functions are calculated from data of a 5 ps simulation with an adaptive-precision potential.

Thereby, we do not detect atoms with expected thermal fluctuations but all other atoms. We use $\text{CSP}_{10} = 2.5 \text{ \AA}^2$ since the difference of $0.5 \text{ \AA}^2$ between the thresholds allows a smooth transition for atoms from the fast to the precise potential. One might use a slightly different value for CSP_{10} , but far lower values limit the possible performance gain since thermally fluctuating atoms are treated partially with the precise potential. It is essential to treat the majority of the thermally fluctuating atoms completely with EAM, as one can only save compute time when ACE is not evaluated for most atoms at all.

2. Switching function

The cutoff radii $r_{\lambda,lo}$ and $r_{\lambda,hi}$ are used in the calculation of $\lambda_{min,i}$ according to Eq. (B4) to decrease the switching parameters for neighboring atoms of atoms, which require a precise calculation.

The force on an atom depends on all switching parameters within the force cutoff. Thus, $r_{\lambda,lo}$ and $r_{\lambda,hi}$ increase the force precision for precisely calculated atoms. To determine the cutoff radii $r_{\lambda,lo}$ and $r_{\lambda,hi}$, we calculate energies and forces according to Eqs. (C1) and (1) for one time step with $\lambda_{min,i}$ as switching function under the neglect of $(\nabla_i \lambda_k)$ as discussed in Sec. II C. However, we do not apply a local thermostat to prevent random influence in this parameter study and calculate only one time step. We use a snapshot of a nanoindentation, including dislocations. The energy error compared to the precise ACE energy, as shown in Fig. 7(a), vanishes by design for precisely calculated atoms independently of the varied parameters since it depends only on $\lambda_{min,i}$ of the corresponding precisely calculated atom i . The force error compared to the precise ACE forces is shown in Fig. 7(b) and depends for the case of thermal fluctuations mainly on the width $r_{\lambda,hi} - r_{\lambda,lo}$ of the switching zone between precise and fast atoms but is independent of $r_{\lambda,lo}$. The force error for thermal fluctuations can be larger than for the EAM reference force since only the total forces rather than the force contributions of EAM and ACE are fitted, as shown in Fig. 1. Therefore, a switching zone is needed to change smoothly between EAM and ACE atoms. The larger the switching zone, the smaller the force error on thermal fluctuations, but the more computationally expensive the calculation becomes since more precise calculations are required. We selected a switching zone width of $r_{\lambda,hi} - r_{\lambda,lo} = 8 \text{ \AA}$. The force error for precisely calculated atoms depends primarily on $r_{\lambda,lo}$. We selected $r_{\lambda,lo} = 4 \text{ \AA}$.

The switching parameter $\lambda_{min,i}$, according to Eq. (B4), of atom i fluctuates as it depends on the distance r_{ij} to neighboring atoms j , which require a precise calculation. Thus, we use a time average of $N_{\lambda,avg}$ time steps in Eq. (B5) to average out these fluctuations of the switching parameter. As this is the same motivation as for the time averaging of the centro-symmetry parameter in Eq. (B1), we use the same number $N_{\lambda,avg} = N_{\text{CSP},avg} = 110$ of averaged time steps.

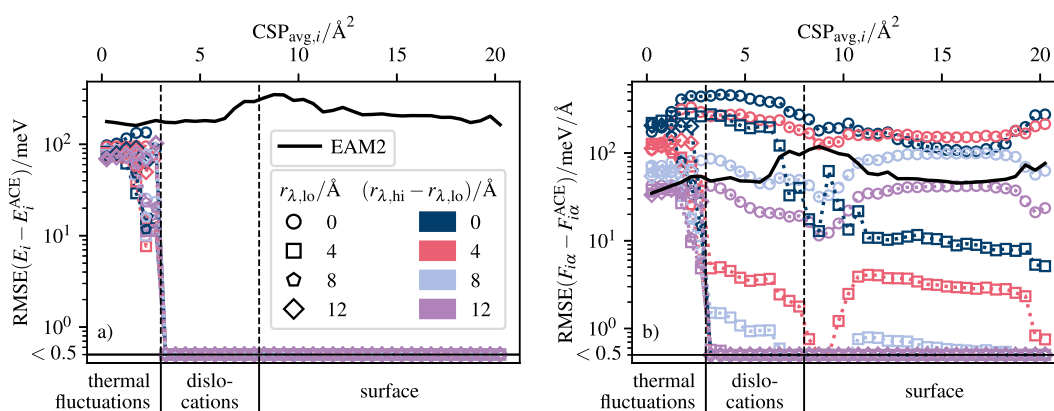


FIG. 7. Root mean square error (RMSE) of the (a) potential energy and (b) forces of the adaptive-precision potential compared with the precise ACE potential for a snapshot of a nanoindentation dependent on $r_{\lambda,lo}$ and $r_{\lambda,hi}$. All atoms whose forces are influenced by both EAM and ACE are grouped dependent on $\text{CSP}_{avg,i}$ and the RMSE is calculated per group. Atoms with pure EAM or ACE force-contributions are not considered since their energies and forces are good. The RMSE of ACE compared to EAM²⁸ is shown as a reference. The cutoff $r_{\lambda,lo}$ is represented by different symbols, and the width $r_{\lambda,hi} - r_{\lambda,lo}$ of the switching zone is color-coded.

3. Local thermostat

An energy error ΔH_i according to Eq. (5) of atom i is corrected by rescaling momenta of the set of random neighboring atoms Ω_i according to Eq. (C6). The distribution of ΔH_i in Fig. 8(a) for a snapshot of a copper nanoindentation at 20 Å indentation depth and 300 K shows that energy corrections need to be applied up to an order of magnitude of 1 meV due to changes in the potential energy. The energy difference between ACE and

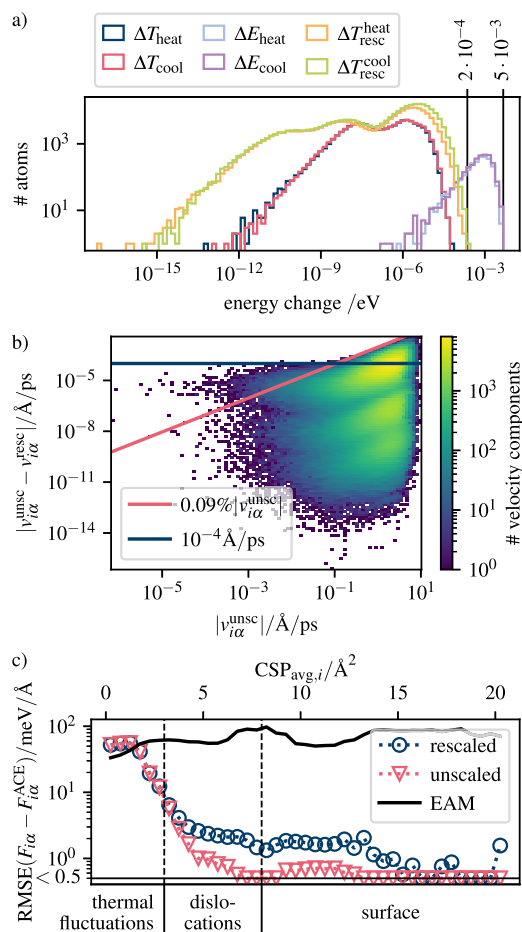


FIG. 8. Rescaling-related quantities of a nanoindentation at 300 K and 20 Å indentation depth. (a) Histogram of potential energy change ΔE_i according to Eq. (C2), kinetic energy change ΔT_i according to Eq. (C3), and kinetic energy change ΔT_{resc} due to the applied rescaling according to Eq. (C4) separated for positive values denoted with “heat” and negative values denoted with “cool.” (b) 2D-Histogram of the velocity $|v_{i\alpha}^{\text{unsc}}|$ before the rescaling step compared with the velocity change $|v_{i\alpha}^{\text{unsc}} - v_{i\alpha}^{\text{resc}}|$ during the rescaling step, where i denotes a particle and α denotes a spatial dimension. (c) Root mean square error (RMSE) of different forces compared with the precise ACE potential. All rescaled atoms are grouped dependent on $\text{CSP}_{\text{avg},i}$ and the RMSE is calculated per group. The applied forces, according to Eq. (C1), are denoted with “unscaled.” The graph denoted with “rescaled” corresponds to the applied forces plus theoretical forces according to Eq. (F1) due to the rescaling. The forces of the optimized EAM potential used for the adaptive-precision potential are denoted with “EAM” and given as a reference.

EAM is in the order of 100 meV according to Fig. 7(a), and the minimum change of the switching parameter is $\Delta\lambda_{\min} = 1/N_{\lambda,\text{avg}} \approx 10^{-2}$. Thus, the energy fluctuations due to the change of λ_i in the order of 1 meV are expected. The kinetic energy T_i of an atom at 300 K is $T_i = (3/2)k_B T \approx 39$ meV, where k_B is the Boltzmann constant and T is the temperature. The kinetic energy changes only slightly with an appropriate time step, and the influence of a λ update on the forces and, therefore, on the kinetic energy change is even lower. Therefore, the contribution of the kinetic energy to the energy error is magnitudes smaller than the potential-energy contribution, as shown in Fig. 8(a). Nevertheless, a kinetic energy change due to rescaling in the order of 1 meV is not negligible compared to an average kinetic energy of 39 meV at 300 K. Hence, we need to distribute the effect of rescaling onto a larger number of particles in order to minimize effects on single atoms. In simulations, we use a maximum number $|\Omega_i| = 800$ of rescaled neighbors. The velocity change due to rescaling is shown in Fig. 8(b) for the snapshot of the nanoindentation. The absolute velocity change $|v_{i\alpha}^{\text{unsc}} - v_{i\alpha}^{\text{resc}}|$ is smaller than 0.1% $v_{i\alpha}^{\text{unsc}}$ for $v_{i\alpha}^{\text{unsc}} \geq 0.1$ Å/ps and smaller than 0.1 mÅ/ps for about $v_{i\alpha}^{\text{unsc}} \leq 0.1$ Å/ps. Thus, the velocity changes are small but, nevertheless, may influence the dynamics of the system. To characterize this effect further, we translate the velocity changes into effective forces. Velocities are updated by the integrator according to $\Delta v = (F/m)\Delta t$ [cf. Eqs. (4a) and (4f)]. Hence, the additional force contribution is given as

$$\Delta F_{i\alpha} = (v_{i\alpha}^{\text{unsc}} - v_{i\alpha}^{\text{resc}})m_i/\Delta t \quad (\text{F1})$$

for all particles i and spatial dimensions α . The force $\Delta F_{i\alpha}$ according to Eq. (F1), however, is a result of the neglect of $\nabla_i \lambda_k$ in Eq. (2) as discussed in Sec. II C. The velocity updates in Fig. 8(b) correspond to the theoretical forces in Fig. 8(c). These forces, including rescaling, on precisely calculated atoms are within a tolerance of up to 10 meV/Å compared to the precise ACE forces. The potential energy E_i depends, according to Eq. (1), only on the switching parameter λ_i of atom i . Therefore, the precision of the potential energies E_i is unaffected by the rescaling.

APPENDIX G: ATOM SUBGROUPS FOR LOAD BALANCING

We execute first the calculation of the fast potential (FP), second the precise potential (PP), third CSP_i (CSP), and fourth $\lambda_{\min,i}$ from $\lambda_{0,j}$ (λ). Which potential needs to be calculated depends on the switching parameter λ . The centro-symmetry parameter is calculated for the group of atoms with a changeable switching parameter. The CSP is not calculated for atoms with a constant switching parameter, which is useful for static atoms used at an open boundary. The fourth subroutine is decreasing, if possible, the switching parameter $\lambda_{\min,i}$ compared to $\lambda_{0,i}$ for neighboring atoms of atoms, which need to be calculated precisely by calculating $\lambda_{\min,i}$ according to Eq. (B4). Atoms i with $\lambda_{0,i} = 1$, which do not require a precise calculation, do not influence $\lambda_{\min,i}$ of neighboring atoms. As $\lambda_{0,i} = 1$ should apply for most of the atoms, these atoms should not cause a load for the calculation of $\lambda_{\min,i}$, which is achieved by iterating over the neighboring atoms of atoms with $\lambda_{0,i} \in [0, 1)$.

APPENDIX H: ADJUSTING POTENTIAL CALCULATION
FOR LOAD BALANCING

For all four force subroutines to be executed independently one after the other on all processors, it is important that none of the subroutines contains communication with other processors, as communication is a synchronization point within a force-calculation subroutine and may include high additional waiting times. Thus, avoiding communication during the force-calculation subroutines is essential to allow effective load balancing. The EAM calculation in LAMMPS includes communication of the derivative of the embedding function and the electron density within the force-calculation routine. In order to avoid communication during the force calculation subroutines, we compute the corresponding quantities on all processors that require them. This results in double calculations, but it is acceptable in our case to achieve proper load balancing. The ACE calculation did not require any adjustments due to communication.

APPENDIX I: PROCESSOR DEPENDENCY
OF SUBPROCESS WORK

Histograms of the work $\rho_p^{(\mathcal{X})}$ according to Eq. (6) used at all load balancing steps of a nanoindentation calculated on 384 processors on JURECA-DC⁵⁴ are shown in Fig. 9. $\rho_{p,t}^\lambda$ with $\langle \rho_{p,t}^\lambda \rangle_{p,t} = 48.6 \mu\text{s}$ is distributed over four orders of magnitude since the calculation time required on atom i also depends on $\lambda_{0,j}$ of neighboring atoms. Since the number of atoms with $\lambda_{0,i} \neq 1$ should be small compared with the number of all atoms, we iterate over all atoms with $\lambda_{0,i} \neq 1$ and calculate $\lambda_{\min,j}$ for all neighbors j of i . As we search a minimum switching parameter $\lambda_{\min,j}$ per neighbor, we can abort the calculation for $\lambda_{\min,j} \leq \lambda_{0,i} \neq 1$ without calculating r_{ij} and $f^{(\text{cut})}$. As the calculation of $\lambda_{\min,j}$ is only required on the processor that administers particle j , it is not required to compute the parameter $\lambda_{\min}$ for ghost particles. Thus, the work caused by an atom i depends on the neighboring atoms and is not constant for all processors. The histograms show $\langle \rho_{p,t}^{\text{FP}} \rangle_{p,t} = 9.6 \mu\text{s}$, $\langle \rho_{p,t}^{\text{PP}} \rangle_{p,t} = 1.7 \text{ ms}$, and $\langle \rho_{p,t}^{\text{CSP}} \rangle_{p,t} = 5.6 \mu\text{s}$, whereas there are two peaks next to each other since the calculation times are dependent on the number of neighbors, which is about a factor of two smaller for surface atoms. Therefore, one cannot use one constant processor-independent time per atom. The

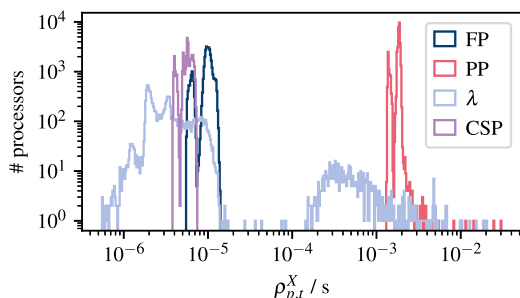


FIG. 9. Histogram of the average work $\rho_p^{(\mathcal{X})}$ for all load-balancing steps of a nanoindentation in a cubic system of 100^3 unit cells and load balancing with a staggered grid domain decomposition.

histograms show the need to measure the required work per atom per processor. Furthermore, the histograms show that the fraction of surface atoms is non-negligible. The atoms at the surface are detected by the centro-symmetry parameter and, thus, require a precise calculation; hence, the domains administered by a processor are small. Note that excluding surface atoms from the precise calculation is possible by comparing the CSP to a given reference configuration, for example, initial or equilibrium configuration, instead of using

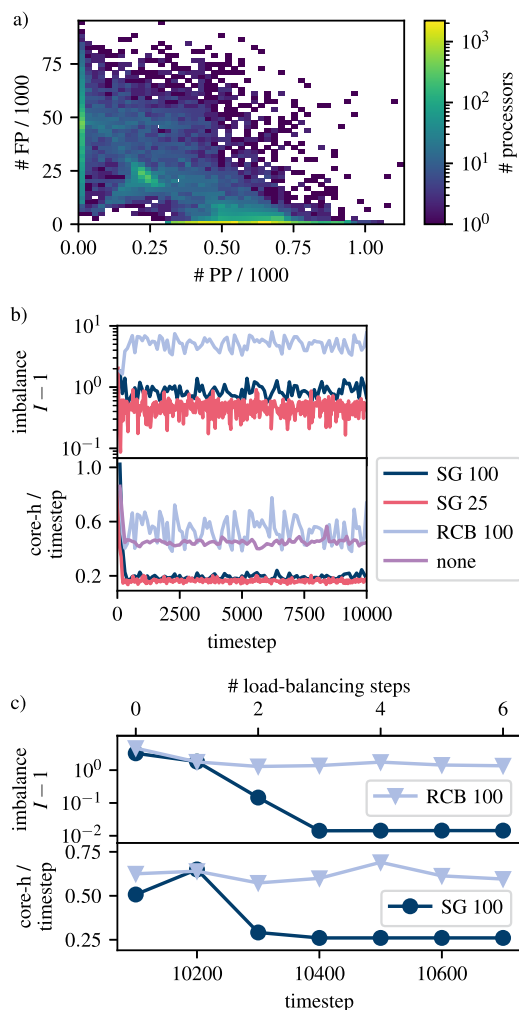


FIG. 10. (a) Number of atoms per processor, which require the calculation of the fast potential (FP) and the precise potential (PP) for all load-balancing steps of a nanoindentation in a cubic system of 100^3 unit cells and load balancing with a staggered grid domain decomposition. (b) Imbalance I according to Eq. (13) and core-h per time step during equilibration of a surface in a NVT ensemble compared for different load-balancing methods. The staggered grid method (SG) uses per processor p the work $\rho_p^{(\mathcal{X})}$ of all four force subroutines $\mathcal{X}$, whereas the recursive coordinate bisectioning (RCB) method only uses one work ρ_p per processor and is thus worse balanced. The number behind the load balancing method in the legend is the number of time steps, after which the corresponding load balancer is called. (c) Load balancing in the system of (b) but with static atoms only to test the behavior of the load balancers.

the absolute value of the CSP as input for the switching function in Eq. (B2).

APPENDIX J: DYNAMIC-LOAD BALANCING DETAILS

We use a nanoindentation to discuss the challenges of dynamic load-balancing of a simulation with an adaptive-precision potential in the following. The numbers of precise and fast calculations per processor are shown in Fig. 10(a) as a 2D-histogram of all load-balancing steps, whereas load-balancing is done as described in Sec. II D. The 2D-histogram shows that balancing a few precise calculations on some processors with many more fast calculations on other processors works. More precisely, the domains with the fewest fast calculations perform, on average, 568 precise calculations, whereas the domains with the fewest precise calculations perform, on average, 44 245 fast calculations.

To analyze dynamic load balancing further and also include non-performant load-balancing strategies, we only use the start of the equilibration instead of a whole nanoindentation. The required core-hours per time step during the equilibration of a surface in a NVT ensemble and the imbalance are shown in Fig. 10(b) for different load-balancing strategies. Our load-balancing method applying a staggered grid (SG) and estimating the work per particle with the four force-subroutine times $\rho_p^{(\mathcal{X})}$ [cf. Eq. (6)] results in an imbalance of 1.88 when used every 100 time steps. Not estimating the work per particle but per processor p with only one time measurement of the total force-calculation time τ_p means that the actual load distribution on the processor is unknown to the load balancer. Thus, load balancing with only one average work ρ_p [cf. Eq. (6)] per processor using the recursive coordinate bisectioning (RCB) method of LAMMPS results in an imbalance of 6.17. The imbalance difference between SG and RCB demonstrates the benefit of using one average work $\rho_p^{(\mathcal{X})}$ per individual force subroutine $\mathcal{X}$. The imbalance 1 of a perfectly balanced system is not reached since the system is dynamic and changes the workload at each time step.

An atom i in a bulk of only thermally fluctuating atoms can change its value $\lambda_{0,i}$ to a value smaller than 1 due to a spontaneous fluctuation, which might be reversed in one of the next time steps and, therefore, produces workload fluctuations due to changes in the neighboring $\lambda_{\min,j}$ parameters according to Eq. (B4). More concretely, the atom changes the switching parameter of all neighboring atoms within the cutoff $r_{\lambda,hi}$ to a value smaller than 1, which implies a precise and expensive ACE calculation. Within the cutoff $r_{\lambda,hi}^{\text{Hyb1}} = 12 \text{ \AA}$ are 626 atoms at 0 K, which is more than the 568 precise calculations for the processors with the fewest fast calculations in Fig. 10(a). Thus, one atom with $\lambda_{\min,i} < 1$ can double the work of its processor, and one cannot expect an imbalance of 1 according to Eq. (13) for such a dynamic system. However, the staggered-grid load balancer reaches a better imbalance with 1.47 when called every 25 time steps since it can better follow the dynamics of the system. When we freeze all atoms in the NVT equilibration of the surface, we get a static force calculation after $N_{\lambda,avg}^{\text{Hyb1}} + N_{\text{CSP},avg}^{\text{Hyb1}} = 220$ time steps. The SG method reduces the imbalance of this static system up to 1.02, as shown in Fig. 10(c). Hence, the staggered-grid-load balancer works, but the system dynamics is challenging. In contrast, the RCB-load balancer cannot balance the static system due to the missing per atom and subroutine $\mathcal{X}$ force-calculation times $\rho_p^{(\mathcal{X})}$ [cf. Eq. (6)].

APPENDIX K: SIMULATION DEPENDENCY OF THE SAVED COMPUTATION TIME

A nanoindentation with 4×10^6 atoms calculated on JURECA-DC⁵⁴ with the adaptive-precision potential Hyb1 (cf. Table IV) requires 11 and 13 times less core-h for equilibration and nanoindentation itself, as visualized in Fig. 11. The amount of saved computation time depends on the system itself. Dislocations develop during a nanoindentation, and therefore there are more precisely calculated atoms at the end of a nanoindentation than at the beginning or during the equilibration of the surface. Hence, one requires 59 and 33 times less core-h for the simulations with periodic

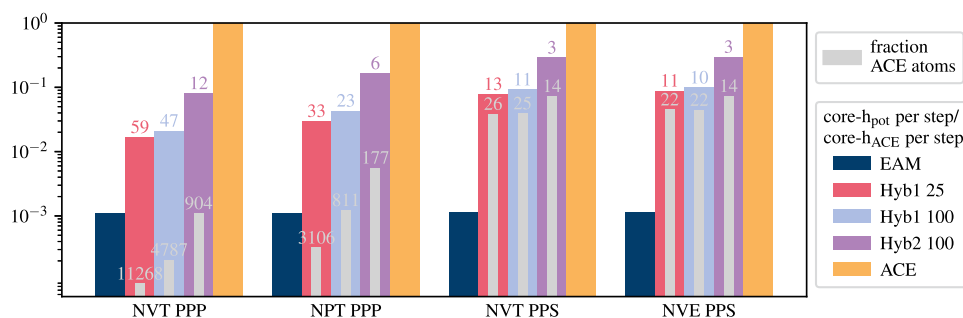


FIG. 11. Total computation time of nanoindentations with 4×10^6 atoms (100^3 unit cells) simulated with adaptive-precision potentials compared to ACE and EAM nanoindentations. The number behind the potential in the legend is the number of time steps, after which the staggered-grid load balancer is called. EAM, hybrid, and ACE simulations are calculated on 128, 384, and 2048 cores of JURECA-DC, respectively. The fraction of used core-h compared to ACE is given on top of the bars. The approximated fraction of ACE atoms is given in gray to visualize the overhead induced by the method. The times are given for all four parts of a simulation separately. NVT and NPT ensembles are simulated with periodic boundaries (PPP) to prepare a surface (PPS). The surface is equilibrated in an NVT ensemble. The nanoindentation itself is calculated in an NVE ensemble.

boundaries in all spatial directions since there are no precisely calculated atoms at the surface. One saves even computation time for all simulation parts for Hyb2, where we increased the cutoffs $r_{\lambda,lo} = 8 \text{ \AA}$ and $r_{\lambda,hi} = 20 \text{ \AA}$ and used the remaining parameters of Hyb1.

REFERENCES

- ¹R. Drautz, *Phys. Rev. B* **99**, 014104 (2019).
- ²A. V. Shapeev, *Multiscale Model. Simul.* **14**, 1153 (2016).
- ³A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, *Phys. Rev. Lett.* **104**, 136403 (2010).
- ⁴A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker, *J. Comput. Phys.* **285**, 316 (2015).
- ⁵J. Behler, *J. Chem. Phys.* **134**, 074106 (2011).
- ⁶A. Seko, A. Togo, and I. Tanaka, *Phys. Rev. B* **99**, 214108 (2019).
- ⁷C. van der Oord, G. Dussan, G. Csányi, and C. Ortner, *Mach. Learn.: Sci. Technol.* **1**, 015004 (2020).
- ⁸Y. Lysogorskiy, C. van der Oord, A. Bochkarev, S. Menon, M. Rinaldi, T. Hamerschmidt, M. Mrovec, A. Thompson, G. Csányi, C. Ortner, and R. Drautz, *npj Comput. Mater.* **7**, 97 (2021).
- ⁹P. Verma and D. G. Truhlar, *Trends Chem.* **2**, 302 (2020).
- ¹⁰A. Warshel and M. Levitt, *J. Mol. Biol.* **103**, 227 (1976).
- ¹¹G. Moras, R. Choudhury, J. R. Kermode, G. Csányi, M. C. Payne, and A. De Vita, “Hybrid quantum/classical modeling of material systems: The ‘learn on the fly’ molecular dynamics scheme,” in *Trends in Computational Nanomechanics: Transcending Length and Time Scales*, edited by T. Dumitrica (Springer, Netherlands, Dordrecht, 2010), pp. 1–23.
- ¹²J. R. Gołębowski, J. R. Kermode, P. D. Haynes, and A. A. Mostofi, *Phys. Chem. Chem. Phys.* **22**, 12007 (2020).
- ¹³C. Peter and K. Kremer, *Soft Matter* **5**, 4357 (2009).
- ¹⁴M. Praprotnik, L. D. Site, and K. Kremer, *Annu. Rev. Phys. Chem.* **59**, 545 (2008).
- ¹⁵R. Cortes-Huerto, M. Praprotnik, K. Kremer, and L. Delle Site, *Eur. Phys. J. B* **94**, 189 (2021).
- ¹⁶N. He, Y. Liu, and X. Zhang, *Int. J. Numer. Methods Eng.* **112**, 380 (2017).
- ¹⁷A. Tabarraei, X. Wang, A. Sadeghirad, and J. Song, *Finite Elem. Anal. Des.* **92**, 36 (2014).
- ¹⁸R. Potestio, S. Fritsch, P. Español, R. Delgado-Buscalioni, K. Kremer, R. Everaers, and D. Donadio, *Phys. Rev. Lett.* **110**, 108301 (2013).
- ¹⁹U. Alekseeva, R. G. Winkler, and G. Sutmann, *J. Comput. Phys.* **314**, 14 (2016).
- ²⁰L. Delle Site, *Phys. Rev. E* **76**, 047701 (2007).
- ²¹M. J. Buehler, A. C. T. van Duin, and W. A. Goddard, *Phys. Rev. Lett.* **96**, 095505 (2006).
- ²²M. Praprotnik, L. Delle Site, and K. Kremer, *J. Chem. Phys.* **123**, 224106 (2005).
- ²³L. Zhang, H. Wang, and W. E, *J. Chem. Phys.* **149**, 154107 (2018).
- ²⁴A. Heyden, H. Lin, and D. G. Truhlar, *J. Phys. Chem. B* **111**, 2231 (2007).
- ²⁵P. Español, R. Delgado-Buscalioni, R. Everaers, R. Potestio, D. Donadio, and K. Kremer, *J. Chem. Phys.* **142**, 064115 (2015).
- ²⁶A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in ‘t Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, *Comput. Phys. Commun.* **271**, 108171 (2022).
- ²⁷M. W. Finnis and J. E. Sinclair, *Philos. Mag. A* **50**, 45 (1984).
- ²⁸Y. Mishin, M. J. Mehl, D. A. Papaconstantopoulos, A. F. Voter, and J. D. Kress, *Phys. Rev. B* **63**, 224106 (2001).
- ²⁹C. L. Kelchner, S. J. Plimpton, and J. C. Hamilton, *Phys. Rev. B* **58**, 11085 (1998).
- ³⁰J. D. Honeycutt and H. C. Andersen, *J. Phys. Chem.* **91**, 4950 (1987).
- ³¹E. R. Homer, *Comput. Mater. Sci.* **161**, 244 (2019).
- ³²H. Tsuzuki, P. S. Branicio, and J. P. Rino, *Comput. Phys. Commun.* **177**, 518 (2007).
- ³³LAMMPS, LAMMPS documentation - compute centro/atom command, 2024, https://docs.lammps.org/compute_centro_atom.html.
- ³⁴LAMMPS, LAMMPS documentation - compute cna/atom command, 2024, https://docs.lammps.org/compute_cna_atom.html.
- ³⁵LAMMPS, LAMMPS documentation - compute cnp/atom command, 2024, https://docs.lammps.org/compute_cnp_atom.html.
- ³⁶M. Li, J. Cui, J. Wang, and Q. Hou, *J. Nucl. Mater.* **433**, 17 (2013).
- ³⁷H.-T. Luu, S. Raugel, F. Dencker, M. Wurz, and N. Merkert, *Surf. Coat. Technol.* **437**, 128342 (2022).
- ³⁸W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, *J. Chem. Phys.* **76**, 637 (1982).
- ³⁹S. Plimpton, *J. Comput. Phys.* **117**, 1 (1995).
- ⁴⁰R. Halver, “Adaptives Lastbalance-Verfahren für Gebietszerlegung in der Molekulardynamik,” M.S. thesis, FH Aachen University of Applied Sciences, 2010.
- ⁴¹H. Iliev, M.-A. Hermanns, J. H. Göbber, R. Halver, C. Terboven, B. Mohr, and M. S. Müller, in *High-Performance Scientific Computing: First JARA-HPC Symposium, JHPCS 2016, Aachen, Germany, October 4–5, 2016, Revised Selected Papers 1* (Springer, 2017), pp. 187–199.
- ⁴²R. Halver, S. Schulz, and G. Sutmann, “ALL—A loadbalancing library,” C++/Fortran library, <https://gitlab.version.fz-juelich.de/SLMS/loadbalancing/-/releases>
- ⁴³A. Hernandez, A. Balasubramanian, F. Yuan, S. A. Mason, and T. Mueller, *npj Comput. Mater.* **5**, 112 (2019).
- ⁴⁴H. S. Huang, L. Q. Ai, A. C. T. van Duin, M. Chen, and Y. J. Lü, *J. Chem. Phys.* **151**, 094503 (2019).
- ⁴⁵D. Geissler, J. Freudenberger, A. Kauffmann, S. Martin, and D. Rafaja, *Philos. Mag.* **94**, 2967 (2014).
- ⁴⁶E. M. Bringa, J. U. Cazamias, P. Erhart, J. Stölken, N. Tanushev, B. D. Wirth, R. E. Rudd, and M. J. Caturla, *J. Appl. Phys.* **96**, 3793 (2004).
- ⁴⁷R. W. Siegel, *J. Nucl. Mater.* **69–70**, 117 (1978).
- ⁴⁸P.-W. Ma and S. L. Dudarev, *Phys. Rev. Mater.* **5**, 013601 (2021).
- ⁴⁹*Atomic Defects in Metals*, Landolt Bornstein, New Series, Group III, Vol. 25: Crystal and Solid State Physics, edited by H. Ullmaier (Springer, Berlin, 1991), p. 88.
- ⁵⁰H. Ledbetter, *Physica Status Solidi A* **66**, 477 (1981).
- ⁵¹A. Stukowski, E. Fransson, M. Mock, and P. Erhart, *Modell. Simul. Mater. Sci. Eng.* **25**, 055003 (2017).
- ⁵²A. Hjorth Larsen, J. Jørgen Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Bjerre Jensen, J. Kermode, J. R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, *J. Phys.: Condens. Matter* **29**, 273002 (2017).
- ⁵³G. Nilsson and S. Rolandson, *Phys. Rev. B* **7**, 2393 (1973).
- ⁵⁴D. Krause and P. Thörnig, *J. Large-Scale Res. Facil.* **4**, A132 (2018).
- ⁵⁵LAMMPS, LAMMPS documentation - fix balance command, 2024, https://docs.lammps.org/fix_balance.html.
- ⁵⁶A. Stukowski, *Modell. Simul. Mater. Sci. Eng.* **18**, 015012 (2009).
- ⁵⁷A. Stukowski, V. V. Bulatov, and A. Arsenlis, *Modell. Simul. Mater. Sci. Eng.* **20**, 085007 (2012).
- ⁵⁸S. P. Coleman, D. E. Spearot, and L. Capolungo, *Modell. Simul. Mater. Sci. Eng.* **21**, 055020 (2013).
- ⁵⁹S. Li, L. Yang, and C. Lai, *Comput. Mater. Sci.* **161**, 330 (2019).
- ⁶⁰A. Bochkarev, Y. Lysogorskiy, S. Menon, M. Qamar, M. Mrovec, and R. Drautz, *Phys. Rev. Mater.* **6**, 013804 (2022).
- ⁶¹C. P. Lowe, *Europhys. Lett.* **47**, 145 (1999).