Hybrid quantum-classical approach for atomistic simulation of metallic systems

Jacek Dziedzic,[1,*] Maciej Bobrowski,[1,2] and Jarosław Rybicki[1,2,3]

[1]*Faculty of Technical Physics and Applied Mathematics, Gdansk University of Technology, Narutowicza 11/12, 80-952 Gdańsk, Poland*

[2]*TASK Computer Centre, Gdansk University of Technology, Narutowicza 11/12, 80-952 Gdańsk, Poland*

[3]*Institute of Mechatronics, Nanotechnology and Vacuum Techniques, Koszalin University of Technology, Racławicka 5-17, 75-620 Koszalin, Poland*

(Received 4 October 2010; revised manuscript received 6 March 2011; published 29 June 2011)

The learn-on-the-fly (LOTF) method [G. Csànyi *et al.*, Phys. Rev. Lett. **93**, 175503 (2004)] serves to seamlessly embed quantum-mechanical computations within a molecular-dynamics framework by continual local retuning of the potential's parameters so that it reproduces the quantum-mechanical forces. In its current formulation, it is suitable for systems where the interaction is short-ranged, such as covalently bonded semiconductors. We propose a substantial extension of the LOTF scheme to metallic systems, where the interaction range is longer and the many-body nature of the potential prevents a straightforward application of the original LOTF technique. We propose to realize the force optimization stage in a divide-and-conquer fashion and give detailed analysis of the difficulties encountered and the means to overcome them. We show how the technique, which we have termed divide and conquer learn-on-the-fly, can be parallelized to utilize several tens of processors. Finally, we present the results of an application of the proposed scheme (utilizing tight binding for the quantum-mechanical part) to nanoindentation and nanoscratching of single-crystal Cu.

DOI: 10.1103/PhysRevB.83.224114 PACS number(s): 02.70.Ns, 71.15.Pd, 62.25.−g, 62.20.−x

I. INTRODUCTION

After the inception of computers, it became possible to complement experimental and theoretical research in materials science with computer simulation. The ever-increasing computational power allows us not only to simulate larger systems and extend the time scales of study, but also to employ more advanced and more accurate methods. Yet, the computational effort of quantum-based methods rises prohibitively with increasing system size, especially for the most accurate approaches, such as configuration interaction (CI). Even moderately accurate quantum-based techniques, such as density functional theory (DFT) or tight binding (TB), require computational effort that scales with the cube of the system size, preventing one from simulating systems larger than 10^1–10^4 atoms (depending on the exact choice of method) even with today's most powerful computers. Although recent years have seen an interest in linear-scaling approaches (cf., e.g., Refs. 1,2), it should be pointed out that these rely on the exponential decay of the density matrix,[3] which is not the case in metallic systems.[4]

For larger systems, not out of choice but out of necessity, one resorts to empirical methods, which offer less accuracy and transferability, but boast better scaling properties. Among these, a well-established method is molecular dynamics (MD), which has proven its usefulness in materials science over the course of several decades. For metallic systems, the computational effort of MD scales linearly with the number of atoms, and today systems of millions of atoms at time scales of a nanosecond can be simulated. Molecular dynamics treats atoms as classical particles, neglecting electronic structure. Time is assumed to be discrete and the velocities and positions of all atoms are obtained by numerical integration of the equations of motion, given the forces acting on the atoms. These, in turn, are derived by differentiating the interatomic potential. The functional form of the potential, while guided by physical intuition, is assumed *ad hoc*, with its parameters tuned so as to agree with certain experimental quantities, such as bulk modulus, elastic constants, density, or cohesive energy. The empirical nature of the potential, while allowing it to perform well for the system under study, usually leads to poor accuracy for systems that are sufficiently different from the one for which it was parametrized (the potential is then said to be poorly transferable).

Ab initio MD (AIMD) is a class of methods in which the forces driving the atoms in an MD computation are obtained from a quantum-based calculation, rather than from an empirical potential. This radically improves accuracy and transferability, yet the cubic (or worse) scaling prevents the application of AIMD to metallic systems larger than several hundreds of atoms.

The so-called cross-scaling methods offer one solution to this problem, by conceptually separating the system under study into two parts—an "interesting" part, which is treated with the quantum-based method, and the "remainder," where classical calculations are employed. The underlying assumption is that oftentimes most of the system is sufficiently close to its equilibrium structure and thus can be reasonably described with the empirical potential. The quantum-based computation is then only performed for the region in which it is deemed necessary (e.g., where chemical bonds are broken and accurate treatment necessitates the inclusion of the electronic structure into the picture). Cross-scaling methods, however, suffer from their own problems, the most prominent of which is the difficulty of devising a physically sound interface between the two computational methodologies, across the two parts of the system.[5]

In 2004, Csànyi *et al.*[6] devised a different approach, in which they propose, instead of directly driving the atoms with the quantum-based forces, to pass this knowledge to the empirical potential. They have termed the method learn-on-the-fly (LOTF), since the potential learns from the results of the quantum-based computation during the course

of the simulation by locally adjusting its parameters. In Ref. 6, encouraging results are given for vacancy diffusivity and brittle fracture (both for Si), employing the three-body Stillinger-Weber potential, which successfully "learns" from tight-binding (TB) and DFT calculations, respectively.

Our aim was to extend the LOTF formalism to metallic systems, where the lack of a band gap and the resulting slow decay of the elements of the density matrix[4] increase the interaction range beyond several Å. Such systems are particularly difficult for cross-scaling approaches, which usually rely on bonding locality in constructing the quantum-classical interface. The significantly longer range of interactions and the fact that many-body potentials such as the Sutton-Chen[7] potential have to be used to realistically model metals rendered our attempt anything but straightforward. Since the many-body nature of the potential is not unique to metals, it is conceivable that the approach we propose would also be applicable to systems other than d-band metals, described by many-body potentials other than the Sutton-Chen potential. In this paper, however, we focus solely on d-band metallic systems, as exemplified by the presented application of the approach to the nanoindentation and nanoscratching of copper.

The paper is organized as follows. In Sec. II, we briefly describe the original LOTF formulation and show why it cannot be directly applied to metallic systems. Following this, we give a detailed explanation of our proposed extension, along with a prescription for parallelizing it. Section III deals with an example application of the new scheme. The final section contains conclusions and a summary of the paper.

II. THE COMPUTATIONAL TECHNIQUE

A. The original learn-on-the-fly method

The learn-on-the-fly technique[6] attempts to alleviate the following problem, typical for cross-scaling methods. The quantum-based computation is limited to atoms within a certain spatial region. This isolation results in the truncation of chemical bonds across the region's boundary. Because the boundary atoms are now undercoordinated, the obtained forces become disrupted, with the magnitude of the disruption being largest in the vicinity of the boundary. Traditionally this problem is amended, to a certain degree, by adding virtual link atoms, which serve to terminate the broken bonds.[8–10] This approach is not suitable for systems with delocalized bonding, which require special treatment. For these, we have proposed a moderately successful approach in Refs. 11,12.

In the learn-on-the-fly method, instead of performing one quantum-based computation for the whole region, one independently calculates the forces on each of the atoms within the region by performing successive quantum computations for small clusters centered on subsequent atoms, discarding all the forces except on the central atom in each cluster. This assumes that truncated bonds and resulting force disruptions several Å away from a central atom have a negligible impact on the force acting upon it.

This switch from a large computation for N atoms to N independent computations for k atoms each (N being the number of atoms within the region, and k the number of atoms within a cluster) has two advantages—one obtains accurate forces on all the atoms within the region, and, if k is sufficiently smaller than N, or, equivalently, if N is large enough, the $O(N^3)$ bottleneck of even the computationally cheapest quantum-based methods, such as TB, is avoided, the new approach then scaling with k^3N. The drawback is that, since the forces are now calculated independently, Newton's third law of motion is not strictly satisfied and the total force in the region is not zero, although its magnitude is expected to be small if the clusters are large enough.

The core idea of the LOTF method is that the accurate forces obtained on the atoms within the quantum region are used to reparametrize the MD potential, instead of driving the atoms. The notion of global potential parameters is abandoned and each atom now has its own set of parameters. These then undergo optimization, using the quantum-based forces as input, which aims to minimize the square difference between the desired ("target") forces and the forces obtained from the application of the optimized potential. This optimization is, in principle, applied to all the atoms in the system, yet in practice the parameters are shown to vary appreciably only in the vicinity of the quantum region. For the atoms outside the region, the original MD force is used as the target for optimization. This approach has two direct advantages—first, the MD engine can be applied to all the atoms in the system, and second, the quantum-based computation need not be performed at every MD step, e.g., it may suffice to reparametrize the potential every 10 steps, which clearly translates to a shorter simulation wall time. The drawback of this approach is that, since the parameters of the potential vary with time, the system Hamiltonian is not conservative and total energy conservation is lost, with the system energy changing abruptly after every reparametrization of the potential. Although these jumps in the total energy are small in magnitude (on the order of 0.2%), their exact impact on the physicality of the system is not clear.

The LOTF method has been demonstrated to work well for semiconductor systems, and it has been successfully used to simulate the fracture of silicon[6,13,14] and silica.[15] An application to water molecules has also been attempted.[16] The approach has since been reformulated—first, to utilize spline-based potentials in fitting,[14,16] then to introduce a system of virtual springs connecting selected atoms, whose stiffnesses are optimized instead of optimizing the potential parameters.[5,14] This has the advantage of decoupling the optimization from the specifics of the potential used, lifting the requirement to recompute MD forces during every step of the the optimization stage, and making the optimization linear, which drastically simplifies this stage. The reformulated approach has its own disadvantages. One inconvenience stems from the need to choose the pairs of atoms to be connected by the springs.[17] Atoms whose neighbors are close to being coplanar become pathological, as additional springs need to be attached to them to exert out-of-plane forces.[14] Also, the success of the reformulated approach has, at least in part, been attributed to the fact that the interatomic distances (which are used as the springs' displacements) serve as very good coordinates for *covalent* systems.[5] For these reasons, we have decided to pursue the original, not the reformulated, LOTF approach as the basis for the extension we propose. We do not preclude the possibility of the reformulated approach performing adequately for metallic systems, however there

are no indications of any prior application of LOTF to metallic systems in the literature.

B. The nanoindentation process for metals

Our interest in the original LOTF scheme was fueled by the desire to accurately simulate the nanoindentation and nanoscratching processes for d-band metals. Nanoindentation is a process in which a hard indenter (tool) penetrates a soft workpiece to a depth of several to several hundred nanometers. During nanoscratching, the tool moves parallel to the surface of the workpiece. Significant costs associated with experimental analysis of nanoindentation motivate the computational approach. Several MD studies of nanoindentation[18,19] and similar processes (ultraprecision machining,[20,21] nanoscratching,[18] and nanocutting[21–23]) have been undertaken, yet it is unclear how realistic the predictions of the empirical potential are if one keeps in mind the heavy bond breaking and bond reconstruction that takes place at the point of contact between the tool and the work material. Purely AIMD approaches are not feasible, because the size of the simulated system is on the order of 10^4–10^6 atoms. One of the key challenges encountered by computational approaches to the study of nanotribological processes is the discrepancy between the length and time scales corresponding to the numerical and laboratory experiments. Atomic-scale computer simulations deal with indentation depths of several Å to several nanometers, whereas in the laboratory the typical indentation depths are measured in hundreds of nanometers. Typical velocities used in simulations are up to nine orders of magnitude larger than corresponding velocities in laboratory conditions,[24] an unfortunate consequence of the limited computing power. This makes a comparison between simulation and experiment particularly difficult, but it also attaches significance to the performance aspect—any proposed model, apart from being physically sound, needs to be efficient (and, likely, parallelizable) to be successfully used to study realistic systems.

We have previously simulated nanoindentation of copper with an infinitely hard indenter, employing a cross-scaling scheme, where a cylindrical region directly below the tip of the indenter was treated with tight binding and the rest of the system with the Sutton-Chen many-body potential,[7] with moderate success.[11,12] The main difficulty of our approach stemmed from the $O(N^3)$ scaling of the TB method [so-called $O(N)$ TB variants[1] rely on short-rangedness of forces and could not be applied], especially since our technique of embedding the quantum-based computation within the classical system relied on ignoring a large fraction of the (distorted) quantum-based forces, which in turn necessitated using a quantum region of a size that bordered on being prohibitive. The promising scaling properties of the LOTF method have enticed us to try to extend it to metallic systems.

C. Failure of direct parameter optimization for large fitting regions with many-body potentials

Metallic systems are poorly characterized by pairwise potentials, which comes as no surprise considering the distinct character of metallic bonding. To arrive at a reasonable description, the potential must include local-volume- or local-density-dependent terms.[25] The Sutton-Chen potential[7]

$$U_{\mathrm{SC}} = \sum_i \varepsilon \left[\sum_{j \neq i} \frac{1}{2} \left(\frac{a}{r_{ij}} \right)^n - c \sqrt{\sum_{j \neq i} \left(\frac{a}{r_{ij}} \right)^m} \right] \qquad (1)$$

is a well-known many-body potential that gives reasonable results for d-band metals, such as Cu, Ni, or Ag (r_{ij} is the distance between atoms i and j, while a, ε, m, n, and c are parameters). In practice, the lattice sums in the above equation are restricted to atoms j within a certain cutoff radius r_{cut} from atom i; a long-range correction to energy is often used to account for this truncation. This cutoff radius is typically two to three lattice constants, which, in the case of Cu, translates to r_{cut} in the range of 7–11 Å. Note how this compares with the Stillinger-Weber potential for Si, which decays to zero at $r_{\mathrm{cut}} = 3.77$ Å.[6]

In an MD simulation, one is mostly interested in forces rather than the potential itself. After differentiating the above formula, one comes to the realization that the force on atom i depends on the lattice sum on atom j, and thus on the positions of not only all atoms within r_{cut} from i, but also on all atoms within the same radius from j. This is a direct result of the local density dependence of the potential.

Up to now, the discussion concerned monatomic systems. In the LOTF formalism, every atom has its own set of parameters, and as a consequence, from the computational standpoint, the situation resembles an alloy system. The Sutton-Chen potential can be extended to alloys, where, in general, parameters of atom i may differ from the parameters of atom j. Rafii-Tabar[26] and Sutton give convincing mixing formulas to calculate the potential parameters and the resulting force contribution on atom i from atom j in such a case. The important realization here is that the force on any atom i will now depend not only on the positions, but also on the parameters of all atoms j within r_{cut}, and—since it depends on the local density of j—on parameters of all atoms k within r_{cut} of j. To realize the impact of this, consider the numbers involved—the force acting on any atom i depends not only on the parameters of i, but also on the parameters of all its neighbors j within r_{cut} (typically 120–500 atoms) and, to a much lesser degree, on the parameters of the neighbors k of the neighbors (typically 800–3500 additional atoms).

Let us now turn our attention to the concrete application of an LOTF-like scheme that we had in mind. A snapshot of a typical nanoindentation-nanoscratching simulation is shown in Fig. 1. The material is copper and the indenter atoms are artificially fixed to make the indenter infinitely hard. The potential cutoff is $r_{\mathrm{cut}} = 10$ Å. A quantum-based region, in the shape of a cylinder, is positioned below the tip of the indenter. Periodic boundary conditions are applied along the z direction, hence the cylindrical symmetry. The radius of the region is $r_{\mathrm{reg}} = 10$ Å and the region comprises about 560 atoms. Assuming the "neighbor-of-neighbor" contributions are negligible, it is sensible to take a cylinder of radius $r_{\mathrm{fit}} = r_{\mathrm{reg}} + r_{\mathrm{cut}} = 20$ Å as the subset of the system where the force optimization should act to modify the parameters of the atoms (we shall call this the *fitting region*). The fitting region will encompass about $N_{\mathrm{fit}} = 2200$ atoms. Furthermore,

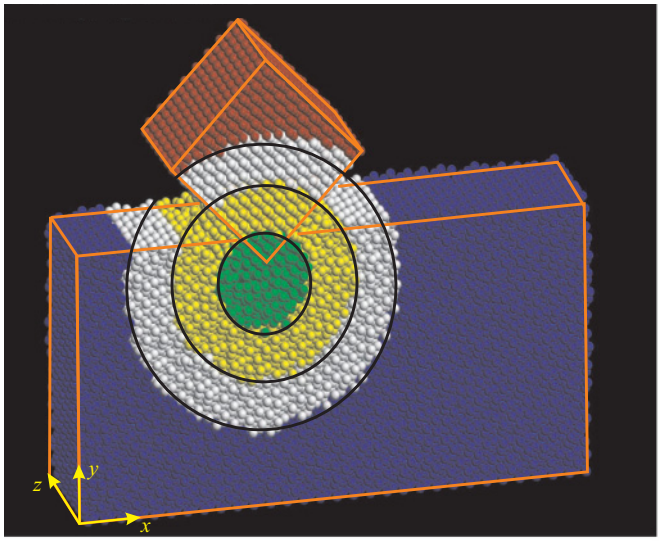

FIG. 1. (Color online) A snapshot of a hybrid nanoindentation-nanoscratching simulation. Atoms in the quantum-based region are drawn in green (innermost circle). Atoms in the fitting region are drawn in yellow and green (two innermost circles). Atoms in the shell region are drawn in white, yellow and green (all circles). The overall shape of the workpiece and indenter are outlined for clarity.

we realize that every change of parameter within the fitting region influences the forces on the atoms outside it, up to a range of another 10 Å, even if we once again ignore the "neighbor-of-neighbor" contributions. In the parameter optimization procedure, we thus need to calculate the square difference between the target and current forces on atoms within a cylindrical region of $r_{\text{shell}} = r_{\text{fit}} + r_{\text{cut}} = 30$ Å, which shall be termed the *shell region*. In this case, it will encompass some $N_{\text{shell}} = 4100$ atoms.

We are now ready to fully appreciate the magnitude of the optimization problem we need to solve when tuning the parameters of the potential. We are in fact trying to optimize a highly nonlinear function of 11 000 variables (2200 atoms, five parameters each), where the function yields a vector of 12 300 values (4100 atoms, three force components each). Not only is this problem much larger than the one posed in Ref. 6 (where 90–200 atoms were treated with the quantum-based method and the shell region encompassed at most 400 atoms), but also the Jacobian matrix involved in the optimization is in our case much more dense because of the longer range of the interaction and the many-body nature of the potential. Furthermore, the force derivatives constituting the Jacobian matrix elements are very involved in the case of the Sutton-Chen potential. Also note that at each step of the optimization stage, the MD forces need to be computed on all atoms i within the shell region, which, as stated earlier, requires the computation of lattice sums for atoms j, some of which lie beyond the shell region.

We have attempted to attack the problem directly, i.e., to apply the original LOTF scheme with the Stillinger-Weber potential replaced by the Sutton-Chen potential, using a high-performance nonlinear Levenberg-Marquadt optimizer, LEVMAR.[27] Posed as such, we have found the problem to be unwieldy because of its sheer size—the $11\,000 \times 12\,300$ Jacobian matrix $\mathcal{J}$ occupies over 1 GB of storage. Apart from the force calculation, each step of the optimization involves computing the following (approximate times on an Intel Xeon machine are given in parentheses): $\mathcal{J}$ (29 000 s), solving a set of linear equations by Cholesky decomposition (500 s) and the computation of the product $\mathcal{J}^T\mathcal{J}$ (900 s). Arguably, the computation of $\mathcal{J}$ could be optimized and parallelized, and the matrix product could be sped up by delegating to BLAS (although LEVMAR already includes a cache-friendly version of this operation, and in this particular case only half of the product needs to be computed). Cholesky decomposition is difficult to parallelize and LEVMAR already uses a LAPACK-based implementation. Note that the timings above refer to only one step in the optimization stage, and we would expect on the order of 100 steps to reach desired convergence. Thus it became obvious that even with aggressive optimization and parallelization, the direct application of the original LOTF scheme is not possible for systems described by the Sutton-Chen potential, unless the fitting region is extremely small.

Since most of the computational effort in the optimization procedure is related to the need to compute and work with the derivatives of the Sutton-Chen force, in the next attempt we have tried to employ a gradient-free optimization method in the form of a genetic algorithm (GA).[28,29] We have augmented our NANOTB[30] computer code with an implementation of a basic GA.[28] Two problems prevented this approach from being successful. The first stemmed from the fact that for every individual ("candidate solution") in the population, the forces on all atoms in the shell region had to be computed. Taking into account that realistic population sizes are on the order of 1000 individuals and the number of generations that need to be evaluated is also about 1000, we arrive at the requirement to calculate some 4 billion MD forces. After aggressive optimization involving loop unrolling, cache-friendly look-up tables for distance components, vectorizing some operations using the Intel Math Kernel Library (MKL), delegating summations to BLAS, trading memory for speed by using cache-friendly look-up tables for parameters, and employing multithreading, we have arrived at a timing of 70 ms for the evaluation of the Sutton-Chen force on all the shell region atoms on a quad-core Intel Xeon processor. This still meant about 19 h of wall time required to perform the optimization. Although GA's are not difficult to parallelize, we note that to make the procedure reasonably fast, about 200–300 quad-core processors would have to be used, assuming near-perfect massive parallelization (which is unlikely). The second problem that plagued our attempt, and one that could not be ameliorated by any amount of computing power, was that the GA did not provide sufficient convergence, i.e., it easily got stuck in local minima, which is not surprising considering the search space is 12 300-dimensional. Typical techniques, such as increasing population size, various forms of fitness rescaling,[28] employing Gray coding to avoid Hamming cliffs, increasing population size, or repeated "shaking" of the system by periodically increasing the mutation probability, improved the situation only marginally (these were all tried on toy models, where the timings were more reasonable). We thus conclude that a direct attack on the problem employing GA's is also infeasible.

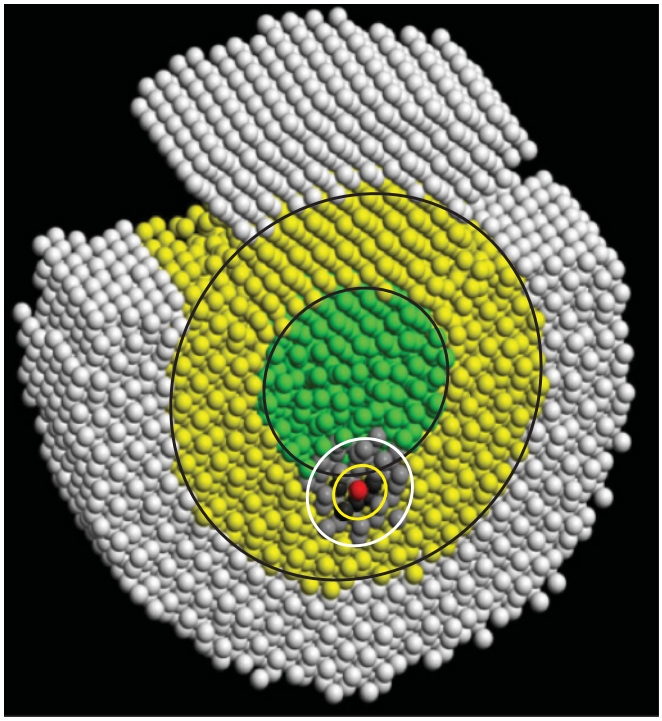

FIG. 2. (Color online) The subset of the system that is of interest during the optimization stage. The quantum region, global fitting, and shell regions are colored as in Fig. 1. The grain center is shown in red (centre of the smallest circle). Atoms belonging to the local fitting region are shown in dark gray and red (smallest circle). Atoms belonging to the local shell region are shown in light gray, dark gray and red (second smallest circle). The optimization grain shown has $n_{\text{fit}} = 6$, $n_{\text{shell}} = 60$ (some of the local shell atoms are hidden from view; they are located in the back of the picture due to periodic boundary conditions).

D. Divide-and-conquer learn-on-the-fly scheme

Having demonstrated that a system treated with the Sutton-Chen potential does not lend itself to a traditional LOTF approach, we will now propose a different scheme, where the optimization stage is modified to proceed in a divide-and-conquer fashion. We begin the optimization by picking an atom from the fitting region and constructing a small cluster of n_{fit} nearest neighbors, with n_{fit} being on the order of 10 (cf. Fig. 2). A slightly larger cluster of n_{shell} nearest neighbors (n_{shell} being on the order of 100) is also constructed around the chosen atom. We shall term these clusters the *local fitting region* and the *local shell region*, respectively, and the chosen atom the *grain center*. Several steps of Levenberg-Marquadt minimization are then performed, with only the atoms in the local fitting region having their parameters optimized and atoms in the local shell region having their forces computed and compared with the target forces. We term this partial optimization an *optimization grain*. After dealing with a single optimization grain, we clearly improve force matching in the local shell region and, hopefully, in the (global) shell region, although from time to time a local improvement can lead to a global worsening, which we accept.

We then repeat the procedure, picking different atoms from the fitting region as grain centers. If care is being taken not to overfit locally (by limiting the number of optimization steps), we generally observe good global convergence after several thousand optimization grains. We find that having picked every atom of the fitting grain as the grain center, dealing with N_{fit} different optimization grains is not sufficient to achieve desired convergence. We assume the forces are sufficiently converged if the rms force difference in the (global) shell region is below 0.01 eV/Å, which is on the order of 1% of the typical force magnitudes in our simulation. Typically it takes $2N_{\text{fit}}$ to $10N_{\text{fit}}$ optimization grains to reach convergence.

After having outlined the basic procedure, we now highlight several questions that needed to be answered before the algorithm could successfully converge to a satisfying error level:

(i) How should grain centers be picked from the fitting region?

(ii) How large should the local regions be, i.e., what are good values for n_{fit} and n_{shell}?

(iii) What should be the starting parameters from which we optimize?

(iv) How many iterations of the optimization procedure should we perform in an optimization grain?

(v) What should be the allowed ranges for the parameters?

We begin with a discussion of question (i). By repeated numerical experiments. we have found that picking the grains at random led to slow convergence and to the optimization getting stuck in a local minimum. Picking the atoms with largest errors first led to some improvement. The following procedure led to a massive improvement of convergence. The optimization stage is divided into a set number of *rounds*. In each round, only atoms with a square difference in forces above a certain threshold are picked as grain centers. The round ends after all grains satisfying this condition have been picked. The threshold is then decreased exponentially and a next round is started. Any atom in the fitting region can be picked at most once in a round, but it may be picked in more than one round. Referring to Fig. 3(a), note that at the start of the optimization the forces are heavily mismatched for atoms in the quantum region and not mismatched at all outside it (in the remainder of the shell region). In the round-based approach, the optimization begins with matching the most mismatched forces first, typically only on several atoms. Then it proceeds to tuning moderately mismatched forces of the remaining atoms in the quantum region. As local optimizations strive only to improve local matching, disregarding longer-range effects of changing the parameters, and as the forces outside the quantum region are initially perfectly matched, any change in parameters can only degrade the matching outside the quantum region [cf. Figs. 3(b) and 3(c)].

This sacrifice is readily made in the name of dealing with the most mismatched forces first. Subsequent rounds of the optimization start to pick atoms outside the quantum region, as the threshold delineating the atoms of interest decreases. The procedure stops either if desired convergence is reached, or a set number of optimization grains is dealt with without reaching convergence (in this case, the algorithm is found to have converged to a local minimum, which is usually close to desired error levels).

Regarding question (ii), our findings indicate that $n_{\text{fit}} = 5$ is the optimum value, with a large margin of tolerance. Much

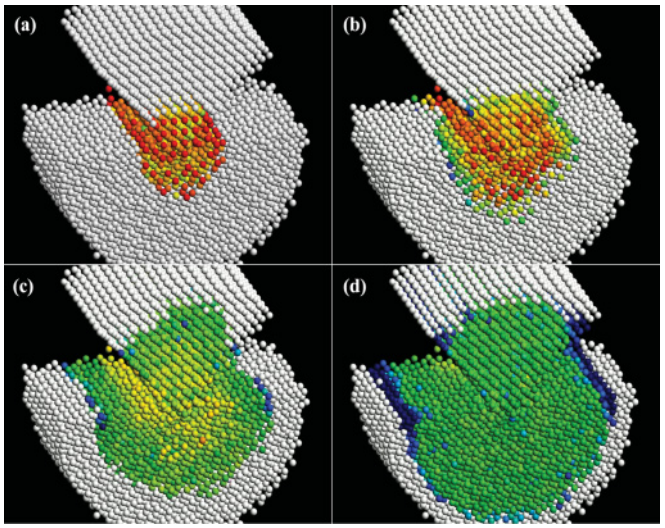

FIG. 3. (Color online) Four snapshots of the optimization stage, taken at (a) the beginning, (b) after having processed 0.5% of grains, (c) after having processed 8% of grains, and (d) having reached the desired error level. The atoms are colored according to the logarithm of the square error between the target and current force. The scale is arbitrary, with green corresponding to an acceptable error level (0.01 eV/Å), yellow to 0.1 eV/Å, red to 1 eV/Å and more, blue to 0.001 eV/Å, dark blue to 10^{-4} eV/Å and below, and white to exactly 0.

larger grains gave faster initial convergence, but inevitably led to premature convergence to one of the local minima, rather far from the desired error level. The likelihood of worsening global force matching while obtaining good local convergence also increased markedly with larger grains. Smaller grains, on the other hand, led to unacceptably slow convergence. For $n_{\text{fit}} = 5$, we have found $n_{\text{shell}} = 65$ (with a large margin of tolerance) to give the best results. Shells much smaller than that meant a disregard for the effects of local optimization on the global picture; shells much larger only served to increase computational time, giving diminishing returns in the quality of convergence. We stress that the values given above should not be regarded as crucial parameters of the method and that observed properties of the system did not depend on the choice of n_{fit} and n_{shell}, provided they were within wide "reasonable limits."

As far as question (iii) is concerned, we have tried to start the optimization from either the original Sutton-Chen parameters, distorted randomly by a small percentage, or from the previously obtained parameters. The second approach gave extremely poor convergence and was quickly abandoned. With the first approach we have found that large distortions (over 10%) caused the initial force mismatch to be so great that the optimization could not reach convergence, getting stuck in a local minimum far from the desired error level. Conversely, not distorting the parameters enough (or not at all) gave better initial matching, but led to slow convergence, which also was caught in a local minimum. We have settled for a distortion of 1%.

Regarding question (iv), we have found five optimization steps per each grain to be a good value. Fewer steps meant slow convergence, whereas too many steps led to local overfitting at the expense of worsening global convergence. When considering the rate of convergence with respect to wall time, rather than the number of local optimizations, the effect of the number steps in the local optimization hardly played any role.[30]

Finally, we turn our attention to question (v), which is difficult to answer. Obviously, constraining the allowed parameter values too much meant the optimization had little room to maneuver, possibly being unable to reach the desired error levels. On the other hand, the newly determined parameters are used for several time steps, with the atoms having moved to new positions. Too much freedom in the choice of the parameters (recall that the parameters m and n are exponents) could lead to unacceptably large forces in future time steps. Also, giving the optimization too much room to maneuver contributed to local overfitting, because it was too easy to fit the forces in an optimization grain. We have found marked improvement after the following procedure was applied. For the sake of clarity, let us rename the Sutton-Chen parameters as x_i, with $i = 1, \ldots, 5$, and their default values (given by Sutton and Chen) as x_i^0. With each parameter x_i we associate a lower and upper bound, $x_i^{\min}$ and $x_i^{\max}$, respectively, constraining the optimization so that every parameter value is contained within $[x_i^{\min}, x_i^{\max}]$. Typically we would choose $x_i^{\min} = 1/2\, x_i^0$ and $x_i^{\max} = 2\, x_i^0$. This gave the optimization rather limited room to maneuver and by itself would typically not result in convergence to satisfying error levels. To facilitate convergence, we would then slowly decrease $x_i^{\min}$ and increase $x_i^{\max}$ with the round number. In that way, the optimizer would first try to get as good results as possible with a restricted range of parameters. In later rounds, "difficult" configurations would be dealt with by allowing the parameters to vary in a larger interval. At all times we have assumed lower and upper "hard limits," typically $\tilde{x}_i^{\min} = 1/3 x_i^0$ and $\tilde{x}_i^{\max} = 3 x_i^0$, beyond which the varied bounds (and so the values of the parameters) could not move. Typical distributions of parameter values are given later in the paper. No parameter conditioning apart from limiting the values was used. The achieved quality of convergence was not sensitive to the details of the strategy employed to widen the ranges for the parameters.

Two points should also be clarified regarding the "legal" values of parameters. First, Sutton and Chen have limited themselves to integral values of exponents m and n, but this was done for performance reasons. We have allowed m and n to take nonintegral values, since this was already the case in alloy systems, where parameter mixing takes place. The second point regards the restriction of $m < n$ implied in the Sutton-Chen potential. The default values for Cu are $m = 6$ and $n = 9$, thus, with the assumed parameter ranges, it would be possible to violate this restriction locally. We have found that this violation did indeed take place in $< 0.1\%$ of atoms, without any visible impact on the results. Reference simulations in which we have explicitly disallowed the violation by taking $\tilde{m}^{\max} = 7.5$ and $\tilde{n}^{\min} = 7.5$ led to macroscopically indistinguishable results.

E. Parallel divide-and-conquer LOTF

The inherent $O(N^3)$ bottleneck of nonorthogonal tight-binding computations for metallic systems cannot, unfortunately, be alleviated by using massively parallel computers.

The reason for this is that solving the generalized eigenvalue problem parallelizes poorly,[31] mostly due to the large communication overhead associated with the reorthogonalization of the resultant eigenvectors. Parallel diagonalization techniques, such as that in Ref. 32, usually rely on the sparseness of the Hamiltonian matrix, which, again, is a property of systems with short ranges of interaction. Our tests have shown that a mediocre speed-up of 2.5 was achieved with eight or more processors using SCALAPACK (pdsyevx routine) and MPI[33] on a parallel machine with a very fast InfiniBand interconnect. An alternative that is parallel, but not massively, is to use a multithreaded BLAS and traditional (nonparallel) LAPACK routines (dsygv or dsygvd). We have found this to yield a speed-up of only 4.0 on an eight-core shared-memory Intel Xeon machine, which is slightly better. Naturally, this solution does not scale beyond a single machine and so it cannot be made massively parallel.

The divide-and-conquer LOTF scheme, on the other hand, offers promising opportunities for parallelization, and we will now turn our attention to the possibilities of exploiting these. Two sections of the algorithm account for over 98% of the computational workload of the method—these are the quantum-based computation and the force optimization. Other parts of the code, such as region selection or the molecular-dynamics engine, although algorithmically complicated, do not significantly contribute to simulation time.

The splitting of the quantum-based force calculation into independent parts (clusters) as realized in the LOTF scheme renders the parallelization of the first section a trivial manner. Delegating subsequent clusters to computational nodes, which then concurrently perform the quantum-based computation, easily allows for massive parallelism. In the NANOTB[30] code we have used a master-slave task farm scheme, where a master processor deals out clusters to a pool of slave processors. Each of the slave processors works on one cluster at a time and immediately asks for another after having finished. In this way, the workload is perfectly balanced across all slave processors until the cluster pool is exhausted. Measurements show close-to-perfect scaling with the number of slave processors.

The parallelization of the second section, the divide-and-conquer optimization, presents a much greater challenge, since gradient-based function optimization is an inherently sequential operation. For this reason, we have begun with parallelizing the force evaluation routine, because it is called repeatedly during the optimization stage (typically 1–2 million times). The LEVMAR optimizer allows for the numerical estimation of the Jacobian matrix by finite differences, at the expense of additional calls to the force routine—we have found this to be more efficient than the analytical computation of the Jacobian because of the extensive optimizations of the force routine we had untertaken and the complexity of the analytical formulas in the case of the Sutton-Chen potential.

Splitting the lattice sums across processors was out of the question, because these represent too fine computational grains—computing the force acting on any atom takes 20–30 μs on one core of the Intel Xeon, depending on the cutoff radius, thus the communication overhead would dwarf the actual computation time in any attempt to multithread this sum. What we have attempted instead was to divide across processors the several tens ($= n_{\text{shell}}$) of force computations that need to be undertaken for every optimization grain. The typical time to deal with an optimization grain on our machine was 1500 μs and the NPTL POSIX thread creation time was on the order of 100 μs. Starting several threads to concurrently work on parts of an optimization grain was thus an option, and we have implemented a multithreaded force computation routine. The obtained timings were close to the expected values, e.g., for four threads the time dropped to approximately $(1500/4 + 4 \times 100) = 775$ μs (a speed-up of 1.9) without hope of improvement with increased numbers of threads, the obvious bottleneck being the thread creation time.

Our second approach relied on a thread pool, where threads were created only once and were awoken using pthread_cond_signal() whenever an optimization grain was ready to be processed. The overhead of signaling and waking threads is substantially smaller than that of thread creation, and we have achieved a speed-up of 3.1 on an eight-core machine. Three factors that contributed to this rather disappointing efficiency, as determined by careful profiling, were as follows: (a) the remaining thread signaling and waking overhead, (b) imperfect load-balancing across threads, (c) penalties for out-of-cache data access and insufficient processor-memory bus bandwidth for concurrent accesses by eight cores. Better efficiency can be achieved for larger shell regions and larger cutoff radii, as the computation time will increase, reducing the impact of factors (a) and (b).

With only limited success in multithreading the force computation routine, we have attempted to parallelize the divide-and-conquer LOTF (DCLOTF) optimization scheme itself using MPI.[33] Given p processors, we attempt to concurrently deal with p optimization grains, delegating each grain to a processor. The obvious difficulty here lies in the fact that, in general, the grains can overlap, i.e., the parameters of some atoms can be optimized (differently) by more than one processor. We shall call such grains *conflicting*. The existence of conflicting grains implies a need for some kind of protocol for maintaining consistency of parameters and forces, forcing the processors to proceed in lock-step with the optimization. Also implied is the impossibility of obtaining massive parallelism—the number of conflicting atoms will quickly grow with an increasing number of processors.

The procedure we have adopted proceeds as follows. The loop that iterates over optimization grains in each round enters an execution barrier (a synchronization primitive that causes all p processors to wait for each other) every p iterations. On every processor, all subsequent iterations but one (representing grains) are skipped, with the iteration that is not skipped being different on each processor (meaning each processor deals with a different grain). As the processors "meet up" at the next barrier, they engage in a conflict-resolving procedure. During this procedure, the conflicting grains are discovered, and the final parameters of the atoms they are centered on are determined as averages over the values obtained on all (usually only two, but in principle up to p) processors that participate in the conflict. The parameters a and ε use geometrical averaging; the remaining parameters are averaged arithmetically, as suggested by Rafii-Tabar and Sutton.[26] The results are communicated to all processors using message passing. Finally, the forces on all atoms in the (global) shell region are recalculated (because parameter changes influence

forces up to $2\,r_{\text{cut}}$ away, it is not sufficient to recalculate forces in the p local shell regions) and the loop continues.

With this scheme we have achieved moderately good speed-ups (4.1 for 8 processors, 6.8 for 16 processors, and 10.3 for 32 processors). Further scaling is seriously limited by the increasing number of conflicts and increasing load imbalance, since the processors need to engage in a synchronous conflict-resolving procedure every time the early finishers have to wait for the last processor to compute its grain. Measurements have shown that for more than 16 processors, the algorithm spends most of its time waiting on the barrier. To avoid this (and to decrease the incidence of conflicts), we combine the two schemes of parallelization—e.g., instead of distributing the computation onto 64 cores via MPI, it is more efficient to use 16 processors with 4 threads, each utilizing one core. With this combined scheme, we are able to achieve a speed-up of 17.8 on 64 cores.

III. EXAMPLE APPLICATION

To illustrate how the proposed method works, we will present results for two sets of simulations of nanoindentation and nanoscratching of Cu with an infinitely hard indenter. In the first set, the proposed hybrid DCLOTF technique was employed; the second set utilized pure MD and served as a reference.

A. Simulation details

We adopt the left-handed coordinate system of Fig. 1. Three systems of differing workpiece orientations were studied. In each, the workpiece was carved out in the shape of a cuboid from a perfect fcc crystal, rotated beforehand so that the crystalline plane of interest would become the xz plane, orthogonal to the direction in which the tool (indenter) moves during nanoindentation. We will distinguish the three systems of interest by specifying the Miller indices of the crystalline plane; these were (010), (110), and (111). The lattice constant was assumed to be $a = 3.62$ Å. The bottom layer of the workpiece was artificially fixed to prevent it from translating upon contact with the tool. Periodic boundary conditions were imposed along the z axis. For the (010) and (110) systems, the system thickness (along the z axis) was taken to be $4a = 14.48$ Å. The requirement to honor the $abcabc\ldots$ stacking of the (111)-oriented workpiece along the direction of the periodic boundary conditions led to a thickness of 18.84 Å (corresponding to $abcabcabc$) for the (111) system. Since the indenter remained in the same orientation as for the other systems, it was extended by two additional layers of atoms (to 18.10 Å) and then stretched by a factor of $18.84/18.10 \approx 1.04$ to comply with the periodic boundary conditions. The workpiece comprised 7564, 7312, and 9480 atoms for the (010), (110), and (111) orientation, respectively.

During nanoindentation, the indenter was moved by an application of a constant velocity of 50 m/s along the $-y$ direction to all indenter atoms. Similar simulations[21] performed with pure molecular dynamics indicate that although considerably larger than the experimental nanoindentation velocity, 50 m/s is not nearly large enough to cause serious artifacts. The indenter was cuboid in shape and was carved out from a perfect fcc crystal, rotated by 45° about the z axis. In our model, the indenter was assumed to be infinitely hard, and forces acting on any of its atoms were ignored and their degrees of freedom were removed from the simulation. The indenter comprised 2916 atoms for the (010)- and (110)-oriented workpiece and 3645 atoms for the (111)-oriented workpiece.

The geometry of the three systems under study closely resembled that of three, from a total of eight, systems studied by Komanduri *et al.*[18] The work material and indenter dimensions along the x and y axes were identical when expressed in multiples of the lattice constant, a. Absolute dimensions differed, because the aforementioned paper[18] dealt with Al, whereas in this work Cu was studied. The thickness of the system along the z direction was also different—Komanduri *et al.*[18] used $3a$ (although Table II therein mistakenly states $6a$, the text gives the correct notion of 6 *atomic layers*), whereas in this work $4a$ was used. This was necessitated by the fact that we use a many-body potential, in contrast to the Morse potential. The thickness of $4a$ is still admittedly rather small, as it required using a maximum MD cutoff radius of $r_{\text{cut}} = 2a$ (and this was used here)—this choice was forced upon the authors due to their desire to limit the required computational time. The nanoindentation and nanoscratching speed was ten times smaller in this work than in that of Komanduri *et al.*[18]

Simulation time step was taken to be $\Delta t = 0.8$ fs, because values much larger than that negatively impacted energy conservation.[30] To allow the system to equilibrate, all simulations began with 100 000 steps of pure MD. During this time, velocity scaling was employed—velocities were adjusted every 10 steps during the first 1000 steps, then every 100 steps until step 10 000, then every 1000 steps until step 100 000. The obtained configurations served as starting points for the simulations discussed later in the text. After 100 000 steps, the distance between the tip of the indenter and the top surface of the workpiece was about 31 Å (the value is approximate, because the workpiece is not static and the top surface continually oscillates with an amplitude of about 0.5 Å), which is several times larger than r_{cut}, and thus there was no interaction whatsoever between the indenter and the workpiece at this time. Subsequent simulation was performed with the indenter continuing to move downward with the same velocity, until the work material was penetrated $2a = 7.24$ Å deep. During this stage, velocity scaling was turned off and a Nosé-Hoover thermostat[34] was used to keep the temperature close to the desired value of 300 K.

The indentation depth of two lattice constants was assumed, to coincide with the nanoindentation simulations in Ref. 18 to facilitate comparison of results. The concept of "depth of indentation" needs to be precisely defined here, since the top surface of the work material tends to shift from its initial position mostly due to the nonzero system temperature. Also, after velocity scaling was ceased, the work material started to relax, with its top surface performing small oscillations. Thus, depending on the temperature and, to a lesser degree, on whether a pure MD or a hybrid simulation was performed, the top surface of the work material was displaced by about 1 Å from its initial position. To account for this difference, the following procedure was adopted. Of the atoms lying on the top surface of the work material, the rightmost one-third was selected and their average displacement along the y axis

was recorded and plotted. The atoms in question were the atoms lying directly under the indenter. An average over four complete oscillations of the displacement of the selected atoms was computed; it was assumed that the equilibrium position of the top surface corresponds to this displacement. In all cases, the tool tip was beyond r_{cut} from even the topmost atom of the surface at the time of the averaging, thus it had no impact on the calculation. The tool came within the r_{cut} from the surface roughly after the fifth oscillation, and jump-to-contact ensued, roughly, after the sixth oscillation. To facilitate comparisons between different systems, subtly differing by the position of the relaxed workpiece surface, we shall from now on employ a coordinate system translated along the y direction in such a way that $y = 0$ will always coincide with the relaxed position of the surface. Under this convention, indentation ceases when the indenter tip reaches $y = -2a$. Correspondingly, we shall denote with $t = 0$ the point in time where the indenter tip reaches $y = 0$.

After the penetration of the work material to the depth of two lattice constants, the nanoindentation terminated and the tool proceeded to scratch the material by moving in the $-x$ direction, with the same speed of 50 m/s. Thus we followed a similar procedure to that of Komanduri *et al.*[18] We have, however, twice increased the length of the nanoscratching stage, terminating the scratching after the tool tip reached $x = -12a$.

In all hybrid simulations, the x coordinate of the center of the quantum region corresponded to the x coordinate of the indenter tip. The y coordinate of the region center coincided with the *initial* position of the work-material surface, because the relaxed position of the surface could not be determined *a priori*. For the sake of simplicity, the region was not moved as the indenter approached the work material. However, during the nanoscratching stage, the quantum region followed the movement of the tool. The quantum region was cylindrically shaped, with a diameter of 42 Å. Depending on the system and the stage of the simulation, it comprised 700–1000 atoms. The fitting region was $42\,\text{Å} + 2 \times r_{\text{cut}} = 56.48$ Å in diameter and comprised 1400–1900 atoms. The shell region was $42\,\text{Å} + 4 \times r_{\text{cut}} = 70.96$ Å in diameter and comprised 2200–3100 atoms. All quantum-based computations employed the NRL Total Energy TB[35–41] implemented from scratch into the the NANOTB code, although, in principle, the technique may be used with other formulations of not only tight-binding but other quantum-based techniques, such as DFT.

B. Results: Behavior of the proposed method

The quantum-based computations and the ensuing force-matching were performed every 10 steps. Following Csànyi *et al.*,[6] we have set the force-matching goal to an error of 0.01 eV/Å in the force, computed as an rms average over the global shell region. We have found that in most cases we could only reach somewhat poorer matching, to 0.02–0.03 eV/Å, which is still satisfactory, keeping in mind the fact that the underlying TB computation is not expected to yield forces to an accuracy better than 0.05 eV/Å,[30] and that the average force magnitude in the simulations described here was 0.2 eV/Å. A total of 3000–4000 optimization grains needed to be processed before the above degree of matching was reached.

In all simulations, eight quad-core processors were utilized, with four threads per processor. The average time to perform an MD step for the whole system was 1.8 s, the average time to perform the quantum-based computation was 128 s, and the average time of the optimization stage was 115 s, yielding, on average, a timing of 25.9 s per one step of the simulation, which compares favourably with 341 s for direct-diagonalization tight binding, embedded using our previous technique.[11,12] However, the limited usefulness of such direct timing comparison should be pointed out. First, in the previous technique, a large fraction (up to 90%) of the quantum-based forces had to be discarded, and thus the actual number of atoms that get quantum treatment is much smaller. Second, the present simulation utilized 32 processor cores, and the one using the previous technique used only one core (because of the impossibility of reasonable parallelization mentioned earlier). More meaningful conclusions can be drawn by observing that the new technique scales as $O(N)$ with a large prefactor, and the old one scales as $O(N^3)$ with a smaller prefactor; the new technique can utilize several tens of computing cores, while the old one cannot. Thus, a simulation with a quantum region twice as large would take four times as much wall time with the new technique and 64 times as much wall time with the old technique (as the number of atoms grows with a square of the cylinder size).

Figure 4 presents histograms of the residual error between the target and current forces at three points during the parameter optimization stage—at the start of the optimization, after 1600 grains have been processed, and after 5200 grains have been processed. It can be seen how the error quickly diminishes from unacceptably large values of 0.1–1 eV/Å. The outermost atoms of the shell region, hardly influenced by the parameter changes in the fitting region, contribute to the left slope of the histogram. A small fraction of atoms (about 0.3%) remains with an error in the forces larger than 0.1 eV/Å, and 25% of the atoms remain with the error in the forces larger than the goal of 0.01 eV/Å, however their number decreases quickly for larger error magnitudes. The particular atoms for which the optimization goal was not reached differ between the steps and are almost universally located in the QM region or in its close vicinity. We attribute these occurrences to ill-fated local optimizations, which are then difficult to undo with further local optimizations. We are presently working on techniques to "iron out" such occurrences, e.g., by repeated attempts to optimize locally, starting from different initial parameters or by locally employing a genetic search. Comparing panels (b) and (c) reveals the quickly diminishing returns from investing more effort in optimizing further grains. For this reason, in the simulations described here the optimization was terminated before 5200 grains were processed.

Figures 5–9 show the histograms of the potential's parameter values before and after a typical optimization stage. It can be seen that only a minority of atoms have extreme values of the parameters, which we interpret as the algorithm being well-behaved without the need for parameter conditioning. The only exception is the ε parameter, but this is as expected, since the Sutton-Chen potential is linear with respect to this parameter.

Hybrid methods are often plagued by artifacts resulting from the fact that two very different methodologies are

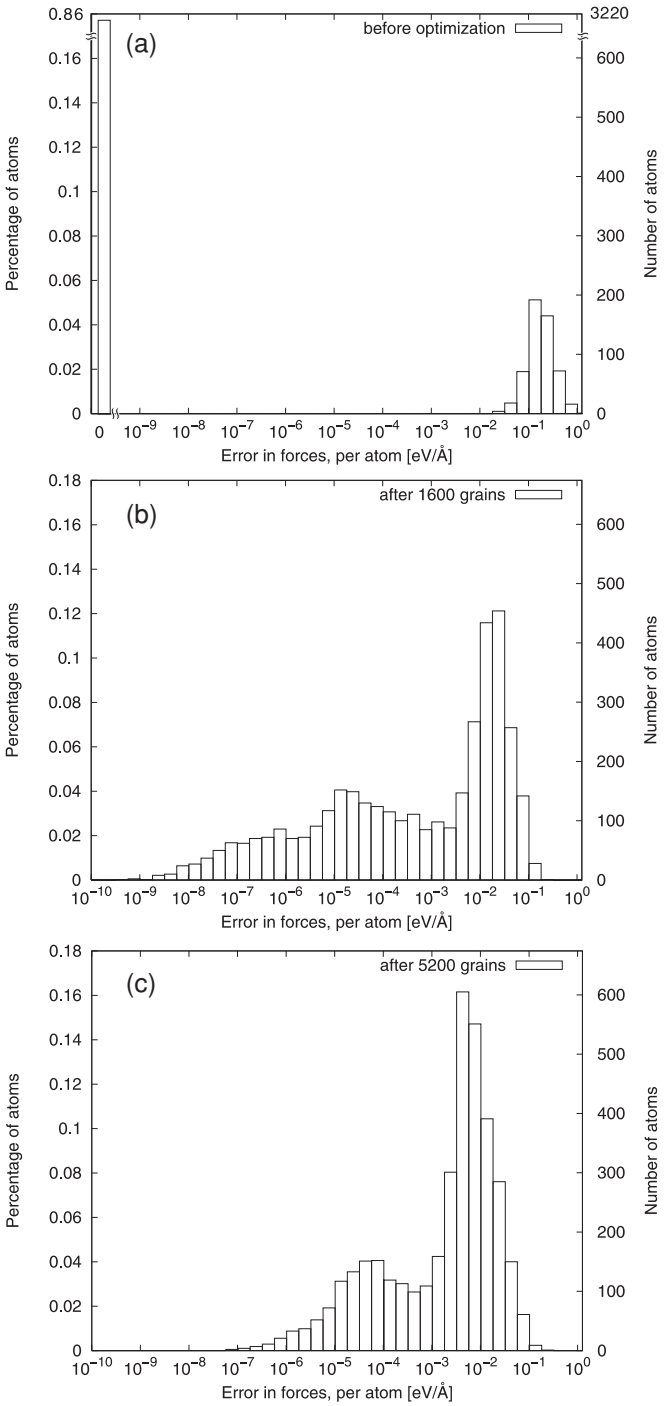

FIG. 4. Distribution of the residual error between the target forces and the current forces, calculated per atom: (a) before the optimization, (b) after 1600 grains have been processed, and (c) after an extended optimization, after 5200 grains. Note how before the optimization, all the forces are either unacceptably distorted (atoms in the quantum region) or exactly matched (atoms outside the quantum region, where the target force is the MD force and the error is zero by definition).

combined within one simulation. Force-mixing methods will usually not conserve either total momentum or total energy.[5] It is also crucial that bulk properties, such as lattice constants and elastic constants, between the two descriptions (QM and MD) are well matched.[5] We have already examined elsewhere[30] the mismatch between the lattice constants of Cu as predicted by NRL-TB and MD with the Sutton-Chen potential and how it affects local pressure. Work on parameter rescaling to eliminate this small discrepancy is in progress. We have also shown[30] that although the LOTF formalism does not promise to conserve total energy, we observe surprisingly good energy conservation in the long run in simulations similar to the ones described here, but performed without thermostatting and the nonequilibrium associated with artificially moving the tool.

In this paper, we add to these statements a simple *post-hoc* observation that no obvious serious artifacts (runaway forces or energies, spectacular failures of force-matching, significant violations of conservation of momentum, etc.) were observed in any of the three hybrid simulations described here. We have, however, observed a moderate artifact in which the vicinity of the quantum region tended to heat up spontaneously. This, however, is expected in a technique where the system Hamiltonian is nonconservative and atoms are allowed to move between QM and MD regions (which is especially true when the QM region moves; for a further discussion, see Ref. 5, Sec. 3]) and it can be counteracted by employing a thermostat. We used the simplest thermostatting approach by applying a global Nosé-Hoover thermostat.[34] As in these simulations the heating is local and mostly occurs near the quantum region, it is important to control this region carefully. This can be done in a more efficient way by using a massive thermostat, i.e., by thermostatting each particle with a separate thermostat. Implementing more advanced thermostats, such as a massive Nosé-Hoover thermostat, a Branka-Wojciechowski[42] thermostat, and chain thermostats, is planned in future works. In principle, one could even use Langevin-like thermostatting.[43] The global nature of the employed thermostat meant that a modest time step of only $\Delta t = 0.8$ fs had to be used, whereas with pure MD we have successfully performed similar simulations with time steps of $\Delta t = 2.5$ fs.

C. Results: Nanoindentation and nanoscratching of single-crystal Cu

We begin the analysis by visually inspecting the atomic configurations of the systems under study. Since the simulated systems are somewhat thicker than in similar simulations reported by Komanduri *et al.*,[18] direct observation of projections of atomic positions onto the xy plane is only moderately revealing. For this reason, we have chosen to visually indicate plastic deformation and slips by employing slip-vector analysis. The atomic slip vector serves as a convenient measure of local plastic deformation and, following Zimmermann *et al.*,[44] it can be easily calculated as

$$\vec{S}_i(t) = \begin{cases} 0 & \text{if} \quad N_i^{\mathrm{sl}} = 0, \\ \frac{1}{N_i^{\mathrm{sl}}} \sum\limits_{\substack{j \neq i, \\ |\vec{r}_{ij} - \vec{r}_{ij}^{\,0}| > \delta}}^{N_i^n} \left(\vec{r}_{ij} - \vec{r}_{ij}^{\,0}\right) & \text{if} \quad N_i^{\mathrm{sl}} > 0, \end{cases} \qquad (2)$$

where $\vec{r}_{ij}$ and $\vec{r}_{ij}^{\,0}$ are vectors joining atom i with any j of its N_i^n neighbors at current time and in the undeformed configuration, respectively. N_i^{sl} is the number of neighbors that have been displaced by more than a certain threshold displacement δ from atom i (that is, the number of terms under the sum). Note

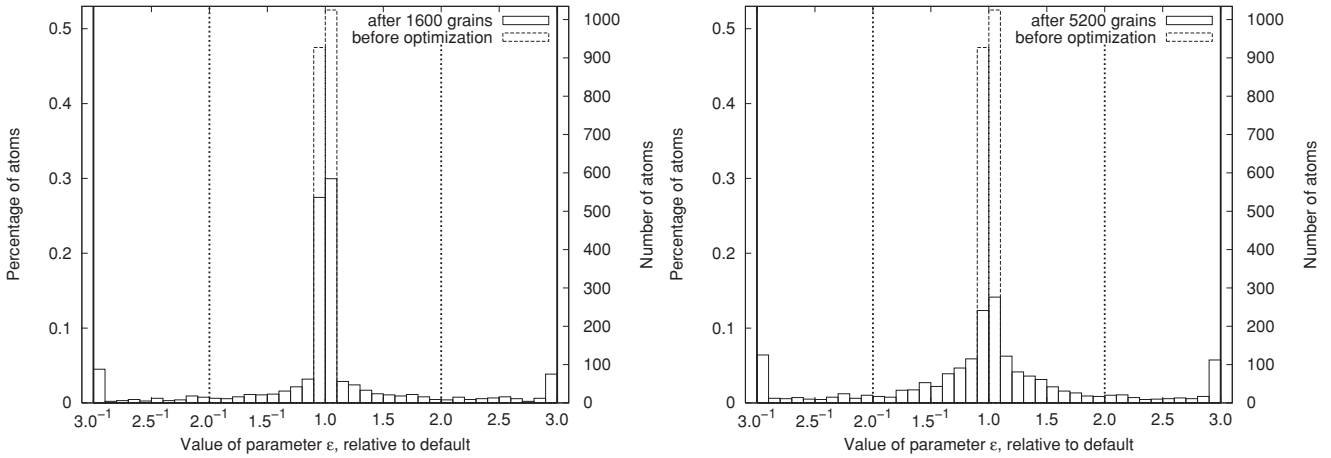

FIG. 5. Histogram of the values of parameter ε before optimization (dashed line), after 1600 optimization grains (solid line, left panel), and 5200 grains (solid line, right panel). Values are shown on a geometric scale. Dashed vertical lines represent $x_i^{\min}$ and $x_i^{\max}$, solid vertical lines represent $\tilde{x}_i^{\min}$ and $\tilde{x}_i^{\max}$.

that in contradistinction to the typical meaning used in the text, where "neighbors" denoted the atoms within r_{cut} from the atom in question, here only atoms within the first coordination shell around atom i are considered to be "neighbors." These are found by employing a cutoff marginally larger than the nearest-neighbor distance $a\sqrt{2}/2$. The threshold δ was taken as 25% of the lattice constant to indicate not only regions (usually planes) that have already undergone irreversible slip, but also those where slipping only begins to take place and would be prevented if the load on the material were to be removed.

We have employed in-house visualization software to produce animations showing the behavior of the systems under study with a resolution of 0.4 ps (snapshots taken every 500 steps). In each frame of the animation, every atom is assigned a color corresponding to the magnitude of its associated slip vector. Atoms with a zero slip vector are shown in gray. Figures 10–12 show snapshots of the configurations at certain "milestone" points in the simulation, while videos depicting the complete process can be downloaded from here.[45] Careful examination of the animations and associated figures reveals certain subtle differences between the hybrid and classical simulations.

The tendency of the empirical potential to overstructure is revealed by comparing the top corners of the work material—either between the hybrid and reference simulations, or between the left (classically treated) and right (the one given quantum-based treatment) corners in the hybrid simulation. For the (010)-oriented work material, the prediction of pure MD is that at a temperature of 300 K there is no surface reconstruction and no associated rounding of the corners, and no migration of individual atoms along the surface, whereas in the hybrid simulation the part of the system in the vicinity of the corner relaxes, leading to a rounding of the corner. To exclude the possibility that this is merely an artifact caused by a local temperature increase, we have performed a similar simulation, where instead of a hybrid technique we have utilized tight-binding-driven molecular dynamics (TBMD), i.e., one where all the atoms were directly driven by forces obtained from a tight-binding calculation for the whole system,

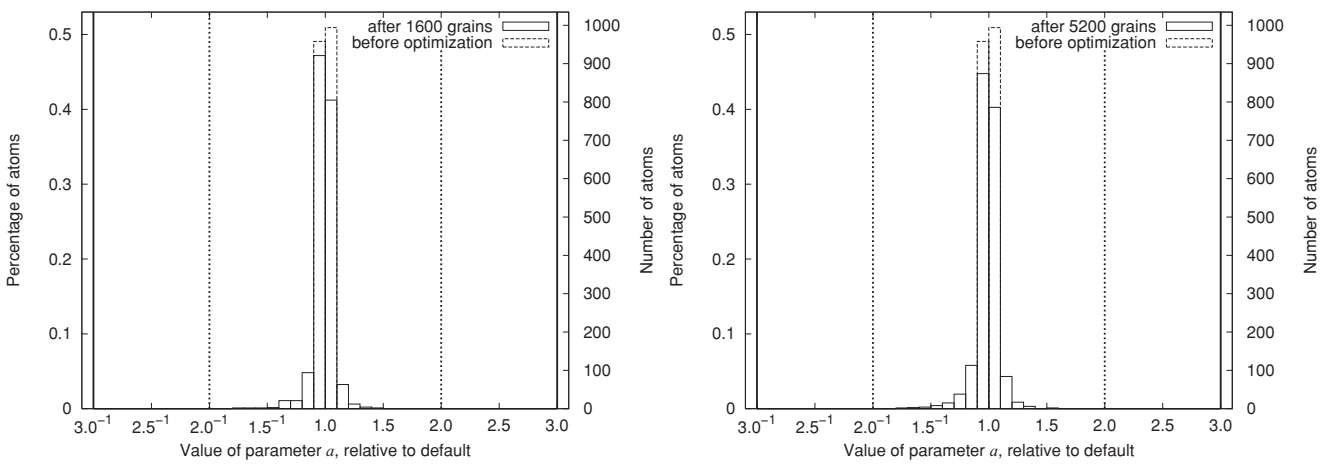

FIG. 6. Histogram of the values of parameter a before optimization (dashed line), after 1600 optimization grains (solid line, left panel), and after 5200 grains (solid line, right panel). Values are shown on a geometric scale. Dashed vertical lines represent $x_i^{\min}$ and $x_i^{\max}$, solid vertical lines represent $\tilde{x}_i^{\min}$ and $\tilde{x}_i^{\max}$.

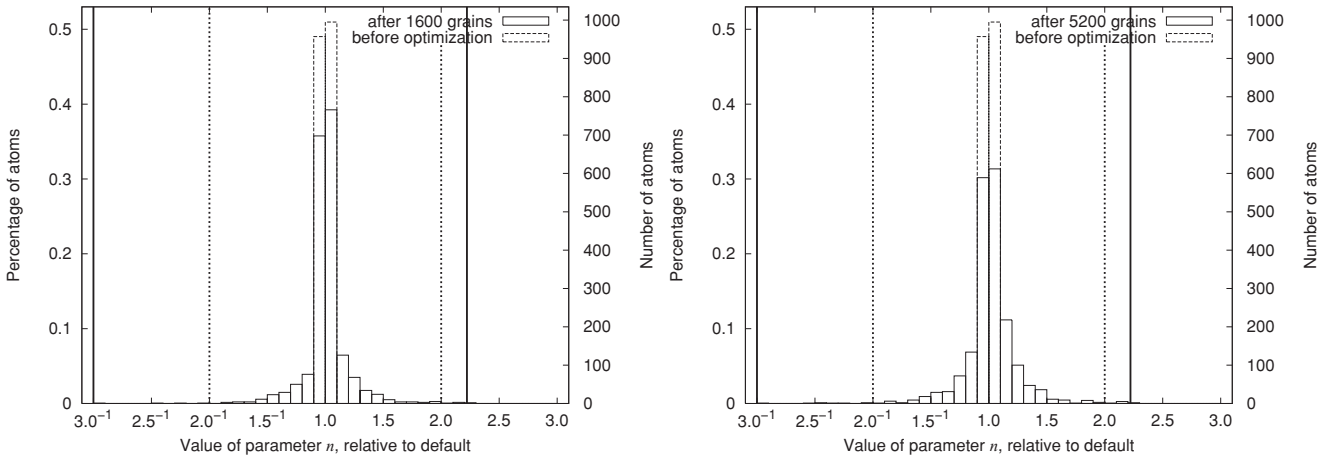

FIG. 7. Histogram of the values of parameter n before optimization (dashed line), after 1600 optimization grains (solid line, left panel), and after 5200 grains (solid line, right panel). Values are shown on a geometric scale. Dashed vertical lines represent $x_i^{\min}$ and $x_i^{\max}$, solid vertical lines represent $\tilde{x}_i^{\min}$ and $\tilde{x}_i^{\max}$.

thus ensuring conservation of momentum and total energy and eliminating any possibility of spurious heating. Because of the high computational effort associated with a full TBMD calculation, we were forced to use a "toy model" system of 728 atoms representing a smaller version of the workpiece, without the tool. As expected, we have observed similar relaxation of all the free corners in this system after a brief run of 7 ps while this relaxation was absent in a pure MD simulation of the same system. We thus conclude that the observed behavior is a direct consequence of the action of the NRL-TB Hamiltonian, and not an artifact of the proposed method. The relaxation of the corners of the (010)-oriented workpiece in the hybrid simulation was still observed at 100 K (but no longer at 50 K), whereas in the pure MD simulations it only occurred at temperatures higher than 300 K. Similar comments can be made about the (110)- and (111)-oriented workpieces, although for these the migration of individual atoms along the surface and a small degree of relaxation can be observed with pure MD even at 300 K, cf. Figs. 11 and 12, yet the effect is more pronounced in the hybrid simulations.

As the indenter approaches the work material, an initial attractive normal force (cf. Fig. 13) is observed in both the reference and the hybrid simulations. This well-known effect is explained by an opportunity for a lowering of energy for the interfacial atoms allowed by their formation of new bonds with the atoms of the indenter tip, while not completely severing the existing bonds with the surrounding atoms of the work material. In metallic systems this translates to a tendency of these atoms to minimize their density-dependent embedding energy, while still maintaining electronic bonds with the rest of the work material.[46] The behavior and magnitude of the observed attractive force is in agreement with results of MD simulations of Rafii-Tabar[46] and of first-principles simulations by Ciraci *et al.*,[47] however it only moderately resembles the results of Komanduri *et al.*,[18] cf. Fig. 17 therein, where the force is reported to oscillate quasiperiodically between attraction and repulsion and is more dependent on the work-material orientation.

The initial attraction between the indenter and the workpiece leads to a jump-to-contact (JC) phenomenon, where

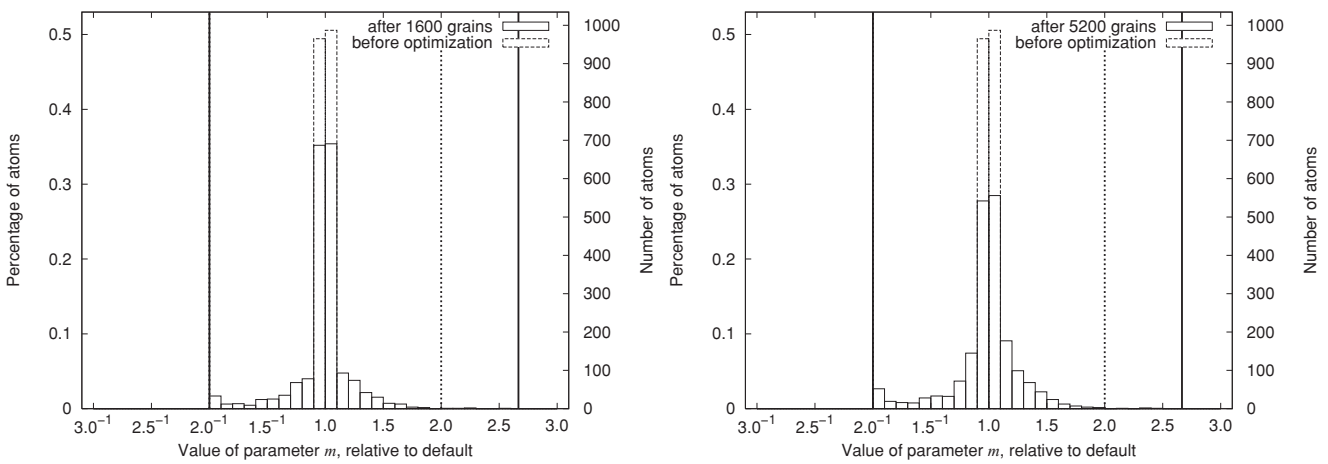

FIG. 8. Histogram of the values of parameter m before optimization (dashed line), after 1600 optimization grains (solid line, left panel), and after 5200 grains (solid line, right panel). Values are shown on a geometric scale. Dashed vertical lines represent $x_i^{\min}$ and $x_i^{\max}$, solid vertical lines represent $\tilde{x}_i^{\min}$ and $\tilde{x}_i^{\max}$.

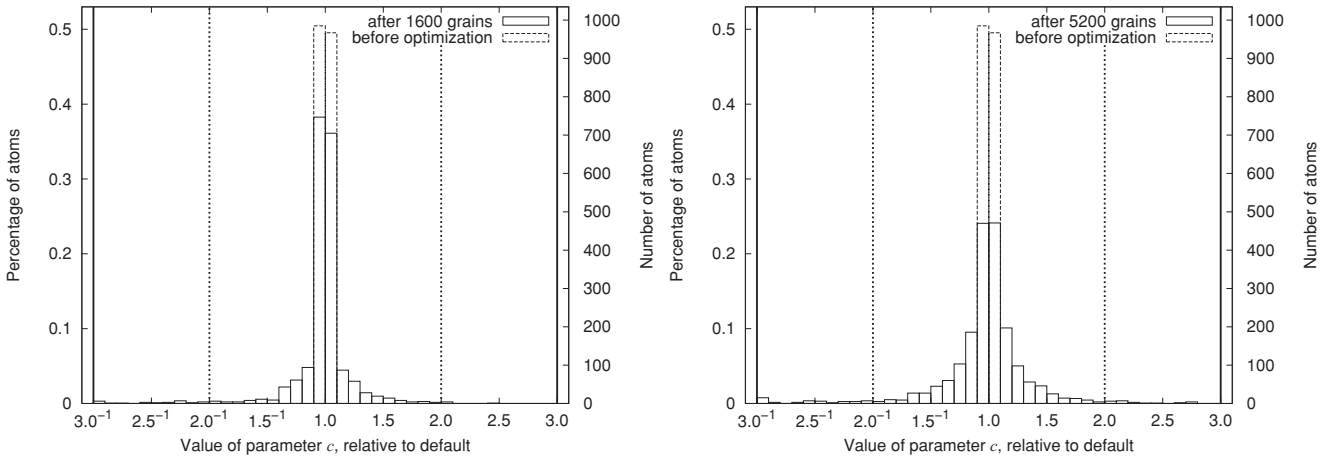

FIG. 9. Histogram of the values of parameter c before optimization (dashed line), after 1600 optimization grains (solid line, left panel), and after 5200 grains (solid line, right panel). Values are shown on a geometric scale. Dashed vertical lines represent $x_i^{\min}$ and $x_i^{\max}$, solid vertical lines represent $\tilde{x}_i^{\min}$ and $\tilde{x}_i^{\max}$.

either the atoms of the tip of the indenter suddenly jump toward the workpiece, or else the topmost atoms of the work-material surface suddenly jump toward the indenter tip. Since in our simulations the indenter is moved artificially, we observe the latter. For a more detailed study of JC and the similar phenomenon of avalanche, the reader is encouraged to consult Rafii-Tabar[46] or Smith *et al.*[48] and references therein. After jump-to-contact ensues and the indenter begins to penetrate the work material, the normal force becomes repulsive. We observe this to happen earlier in the hybrid simulations than

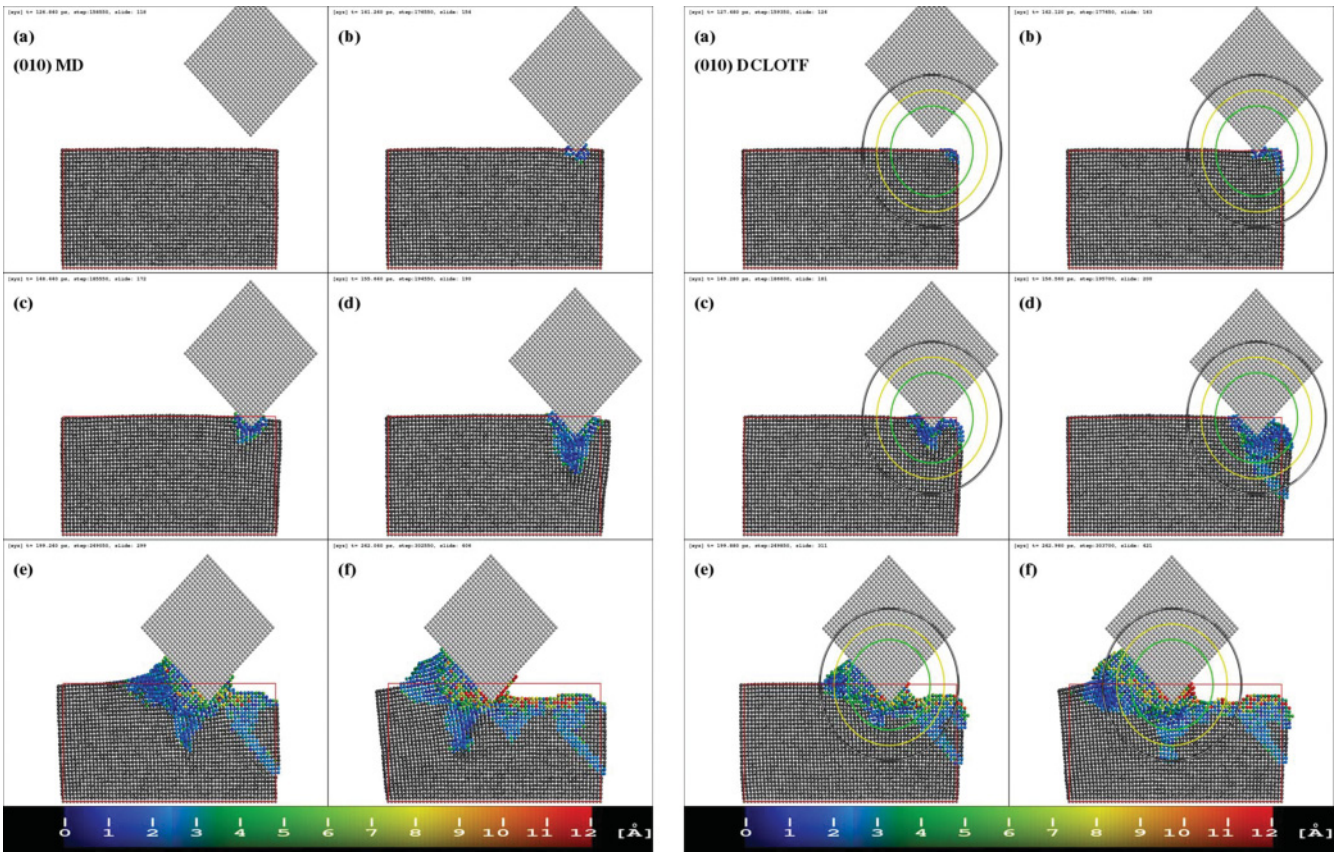

FIG. 10. (Color online) Snapshots of the system configuration for the reference (MD) simulation (left) and the hybrid (DCLOTF) simulation (right) for the (010)-oriented workpiece. For both simulations in panels (a), (b), (c), and (d), the indenter tip has reached $y = -2a$, $y = 0$, $y = a$, and $y = 2a$, respectively, and in panels (e) and (f) the indenter tip has reached $x = -6a$ and $-12a$, respectively. Atoms with a nonzero magnitude of the slip vector are shown in color, corresponding to the magnitude. The outline of the undeformed workpiece is shown for clarity. The quantum, fitting, and shell regions are outlined; the colors of the outline are consistent with those in Fig. 1.

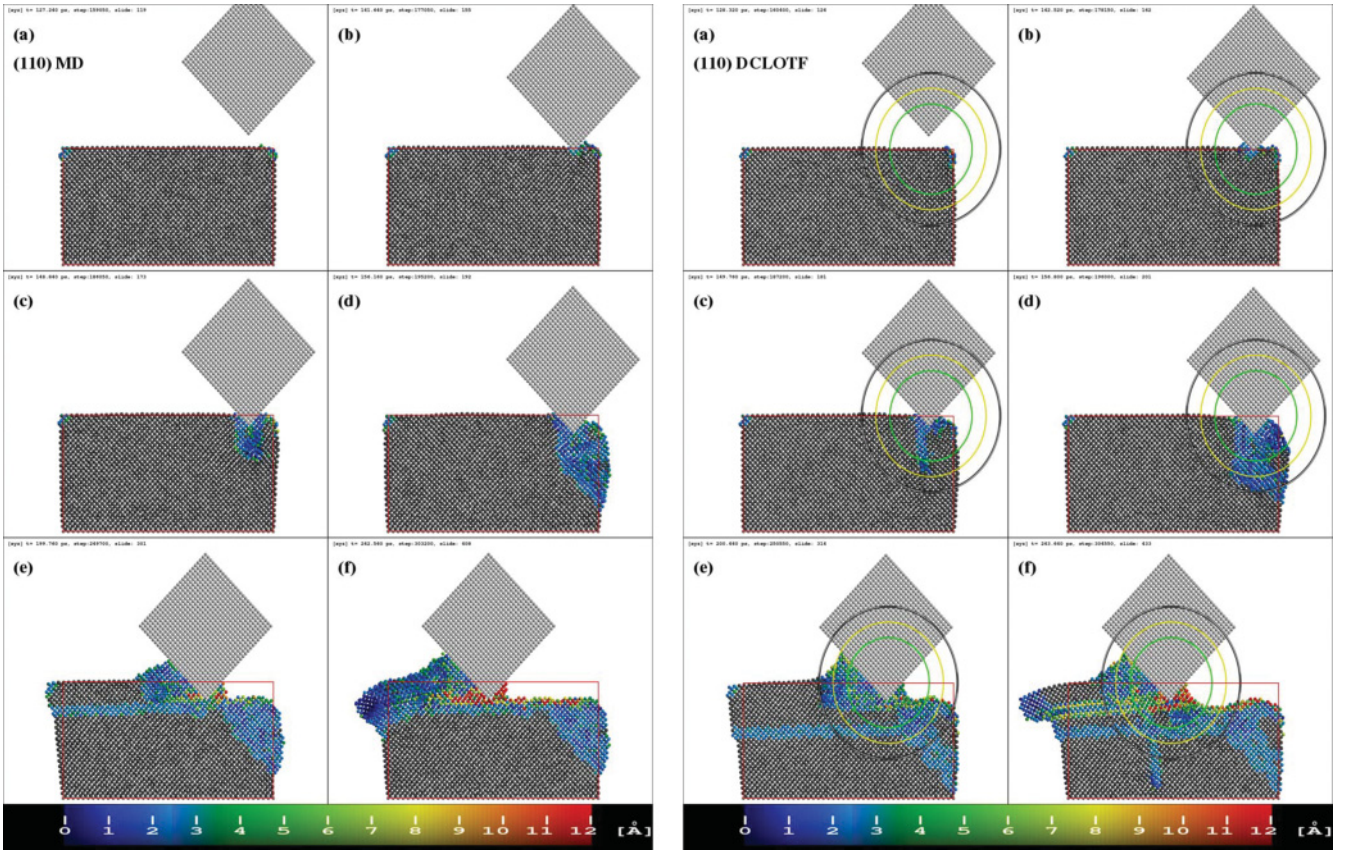

FIG. 11. (Color online) Same as Fig. 10, but for the (110)-oriented workpiece.

in the reference pure MD simulation, and careful examination reveals that the prediction regarding the nature of JC differs between the two approaches. Figure 14 compares the system configurations at the point where JC takes place, revealing that the empirical potential predicts elastic bending of the top of the workpiece toward the indenter, while the prediction of the hybrid approach is that of very limited elastic deformation, accompanied by a displacement of several individual atoms in the immediate vicinity of the tip to form a contact. This is especially seen in the (010)-oriented work material, cf. Fig. 14, panels (a). The displaced atoms then return to their crystalline positions after the tool is advanced. The effect is less pronounced for the (111)-oriented work material and hardly present for (110)-oriented work material, consistent with the observations of Komanduri *et al.*,[18] cf. Fig. 7(a) therein.

To exclude the possibility that the observed effect is merely an artifact of the DCLOTF method, we performed complementary single-point energy calculations with the NRL-TB approach on as large a subset of the system as was possible. We compared the potential energies of the suitably chosen subsets at two instants—immediately after equilibration and immediately after jump-to-contact; these are denoted by (a) and (b), respectively, in Fig. 15. The obtained energies and their differences are shown in Fig. 16. It is seen that the plastic jump-to-contact predicted by the DCLOTF approach is indeed more energetically favorable, at least under the NRL-TB Hamiltonian. We stress the fact that although the configurations were taken from MD and DCLOTF simulations, the energy discussed here was obtained from a fully QM calculation, making a direct comparison of absolute energies meaningful.

Certain differences between the hybrid and classical simulations can be easily observed during nanoindentation. The classical approach predicts that the (010)-oriented workpiece deforms plastically only directly under the tool tip, with the rest of the material deforming elastically by bending the workpiece by as much as one lattice constant [cf. Fig. 10, left panel (d)]. In the hybrid simulation, however, the nanoindentation process ends in brittle fracture of the workpiece, by a combination of slips along the $(1\bar{1}1)$ and $(1\bar{1}1)$ planes [cf. Fig. 10, right panel (d)]. We have reported this phenomenon before[30] and we have consistently observed it in similar hybrid simulations, across a range of temperatures and for varying diameters of the quantum region [36 Å, 42 Å (reported here), 50 Å, and 60 Å], but not when smaller quantum regions (30, 24, and 22 Å in diameter) were employed nor when a purely classical simulation was performed. Thus we believe that this behavior is not coincidental, but rather represents real behavior that is revealed when a sufficiently large portion of the system is treated quantum mechanically.

For the (110)-oriented workpiece, the predictions of both approaches are similar. Directly under the indenter tip, a thin vertical stripe of material begins to slip downward, but this does not happen readily as it corresponds to the $(\bar{1}10)$ plane of the original crystal, which is not a favorable slip plane. This is evidenced by the very high magnitude of the normal force experienced by the tool for this orientation (cf. Fig. 13, middle panel). The stripe of slipped material, in its last stage, can be

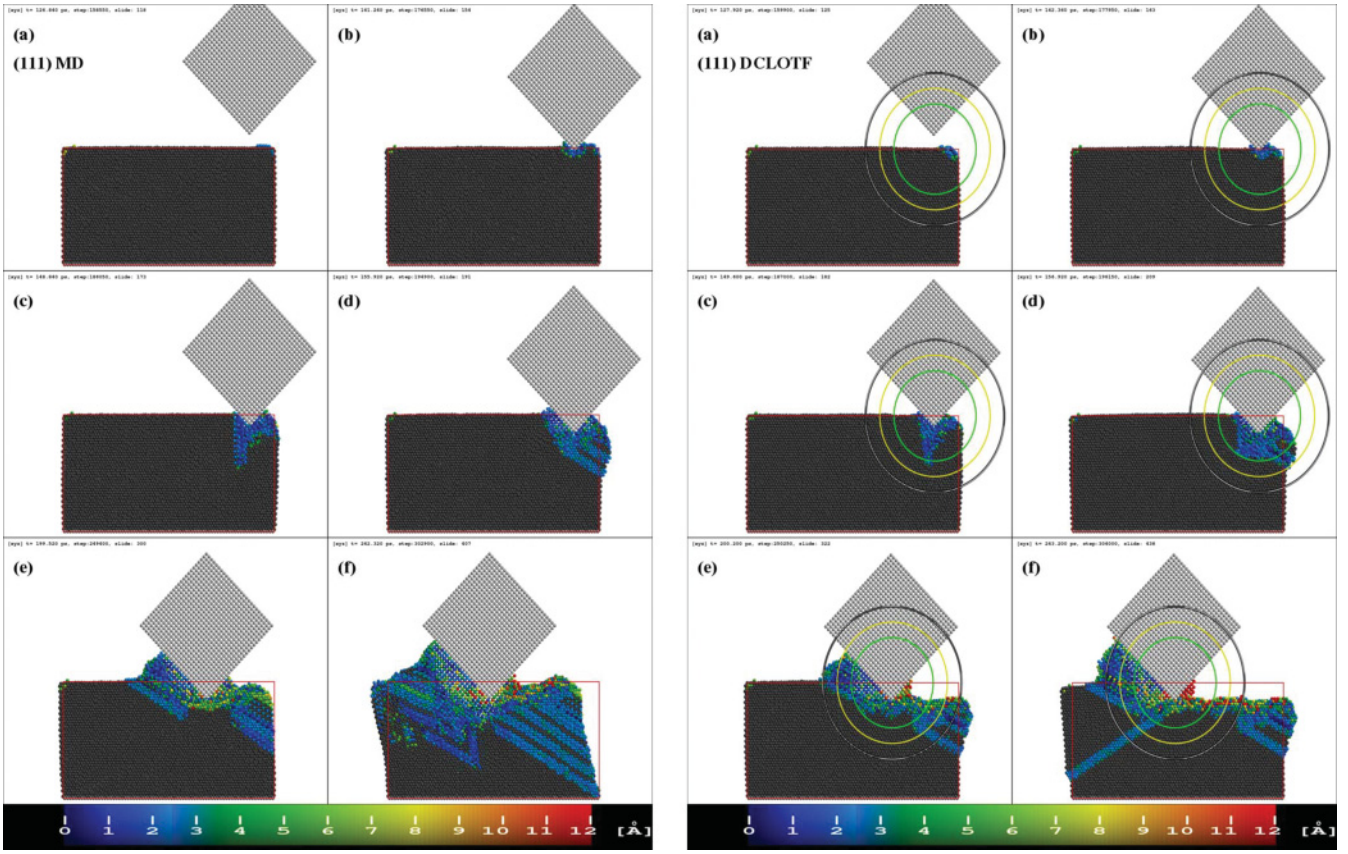

FIG. 12. (Color online) Same as Fig. 10, but for the (111)-oriented workpiece.

easily seen in Fig. 11, right panel (c), whereas the left panel (c) of Fig. 11, owing to the minimally faster unfolding of events in the MD simulation, shows how it disappears, as the material relaxes by a complex combination of slips along (100), $(11\bar{1})$, $(1\bar{1}1)$, (111), and $(1\bar{1}\bar{1})$, to the configuration seen in Fig. 11, panels (d).

Nanoindenation of the (111)-oriented workpiece again proceeds similarly in both the hybrid and reference simulations. In contrast to the (110)-oriented workpiece, plastic deformation immediately ensues not only directly under the indenter tip, but in a larger region, probably because of the vicinity of the edge of the work material. A vertical slip, difficult to express in terms of crystalline slip planes, then begins to form under the indenter [best seen in Fig. 12, left panel (c)], followed by the material quickly relaxing owing to the appearance of two slips along $(1\bar{1}\bar{1})$, best seen in Fig. 12, left panel (d). The two slips are partially overlaid in the case of the hybrid simulation, but the general behavior is similar.

We will now proceed to the discussion of the nanoscratching that followed nanoindentation. For all work-material orientations and with both the classical and the hybrid technique, we observe a pile-up of amorphous material directly in front of the tool, which is expected as the tool has a high negative rake angle. However, as the scratching progresses, up to several layers of piled-up atoms attach to the tool and adopt its perfect crystalline structure, as demonstrated in Fig. 17. We note that a similar effect would probably be absent in an experimental investigation of nanoscratching, because the fact that the tool is undeformable and chemically compatible with the worked material is a characteristic of our simplified model.

As the tool continues to move, atoms directly under the tip are compressed underneath it and reappear behind the tool, where surface reconstruction takes place. To study this reconstructed surface in more detail, we will refer to Fig. 18, which shows a close-up of the top layers of the work material after nanoscratching ceased, and to Figs. 19–21, which show a set of cross sections through these layers. From an examination of Fig. 18 it is clear that the surface is reconstructed roughly at the level corresponding to the indentation depth (denoted layer 0), but that the reconstruction is far from perfect. In the case of (010)-oriented work material, one extra layer of atoms is deposited on the surface close to the point where the tool had indented the material (cf. Fig. 19, layer 1). For this orientation, the reconstruction of the fcc structure is almost perfect—the crystalline ordering of atoms in the layers below the indentation depth (denoted with negative ordinals) is clearly seen in the same figure. The fact that most of the atoms in Fig. 19 have an associated nonzero slip vector (thus being shown in color) indicates that it is not the *original* fcc structure, but rather a reconstructed one. Only in deeper layers can the evidence of the original, undisturbed fcc structure be seen. We note that the hybrid simulation predicts the disorder to reach deeper into the work material, to at least four layers below the tool tip.

For the (110)-oriented work material, we observe that the layer corresponding to the indentation depth is almost completely absent after scratching and the layer immediately

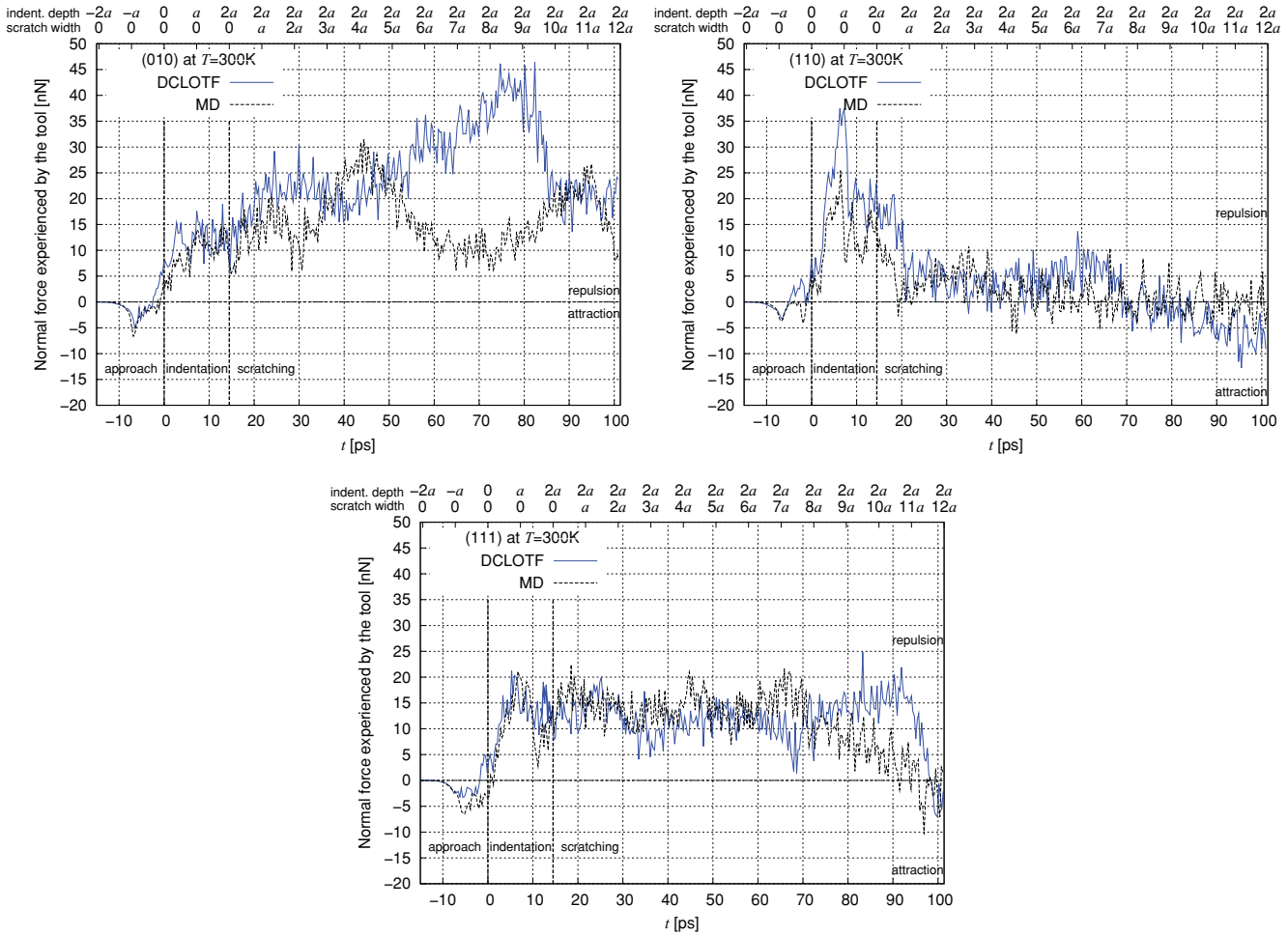

FIG. 13. (Color online) Vertical (normal to the top face of the work material) force experienced by the indenter during the course of the simulation with the (010)-oriented workpiece (top left), (110)-oriented workpiece (top right), and (111)-oriented workpiece (bottom). The solid blue curve refers to the DCLOTF simulation, the dashed black curve to the reference pure MD simulation. The two vertical lines indicate the commencement of indentation and scratching, respectively.

below it (layer 1) is not fully reconstructed. Subsequent layers (layers 2–4) reconstruct to the fcc structure, but this reconstruction is not perfect. The hybrid approach predicts the disorder to reach significantly further than the MD approach (cf. the differences in layers 3 and 4 in Fig. 20). We note that an apparent absence of atoms in the cross sections does not necessarily indicate a presence of voids, but can also be caused by atoms having being displaced in the vertical direction, by a distance larger than the threshold used when generating the cross sections (nevertheless, it is an indicator of disruptions of the perfect fcc structure or of a local deformation of a large magnitude, which may or may not be elastic). These are seen to reach at least four layers below the indentation depth.

In the case of the (111)-oriented work material, it is obvious from Fig. 18, left panel (c) and Fig. 21, left panel that the structure of the work material has undergone a dramatic change in the classical simulation, but not in the DCLOTF simulation. This is also seen in Fig. 12, panel (f) and in the animated video[45] of the simulation. It is apparent that under the stress induced by the tool, the work material has undergone a phase change. This unfortunate effect has to be attributed to a deficiency of our model—especially the modest thickness of the system, associated periodic boundary conditions, and the short potential cutoff it necessitated. We note that a similar simulation with a slightly increased potential cutoff [made possible by the fact that the (111) system was about 4 Å thicker than the remaining systems] did not exhibit this problem, however to maintain consistency, we present here the results of the original calculation. Whether the fact that the hybrid simulation did not suffer from the above-mentioned artifact indicates that the associated Hamiltonian is more robust or was purely accidental is difficult to say. The hybrid approach predicts a reconstruction of the surface at one layer below the indentation depth and moderate disorder up to four layers below.

We will now address the question of the nature of the changes to the original structure of the work material that take place when plastic deformation ensues, i.e., whether the material deforms by brittle fracture or by amorphization or by a combination of the two and how this depends on the orientation of the work material. For the (010)-oriented work material, we find that no brittle fracture occurs with the classical approach, and instead in the immediate vicinity of the tool (up to six lattice constants below, directly under or under and in front of

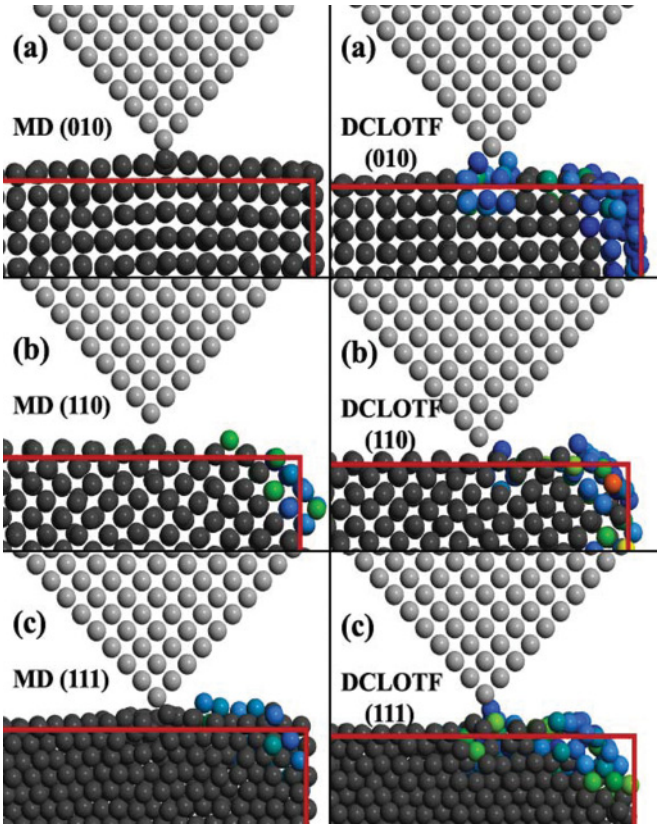

FIG. 14. (Color online) Snapshots of the system configuration for the reference (MD) simulation (left) and the hybrid (DCLOTF) simulation (right) illustrating the difference in the predictions regarding the jump-to-contact phenomenon. Panels (a), (b), and (c) correspond to (010)-, (110)-, and (111)-oriented workpieces, respectively. Coloring follows that of Fig. 10. The quantum region of the DCLOTF simulation is not shown, because all the atoms seen here are within it. The outline of the undeformed workpiece is shown for clarity.

the tool), moderate disorder ensues, which then resolves into the crystalline structure once the stress is relieved by the tool having moved on. The predictions of the hybrid approach are similar, with the exception that the recovery of the crystalline structure is not perfect and plastically deformed regions remain up to several lattice constants below the indentation depth even after the stress induced by the tool is no longer present. In the case of the (110)-oriented work material, both approaches predict an appearance of a crack parallel to the surface of the work material, several layers below the indentation depth. This crack is easily seen in Fig. 11, panels (e). Since this is not an expected slip system for the fcc structure, we examined this crack in more detail to find that it is composed of two intersecting slip planes, viz., $(1\bar{1}1)$ and $(\bar{1}\bar{1}1)$, which in a thin system give an impression of a horizontal slip plane. Subsequently, in the classical approach amorphization ensues above the crack as the tool continues scratching, whereas the DCLOTF approach predicts an appearance of further cracks [seen in Fig. 11, right panel (f)] followed by a thin layer of the work material continuing to slide neatly along the crack, creating the protrusion seen to the left of the same figure. The amorphization in this case is negligible.

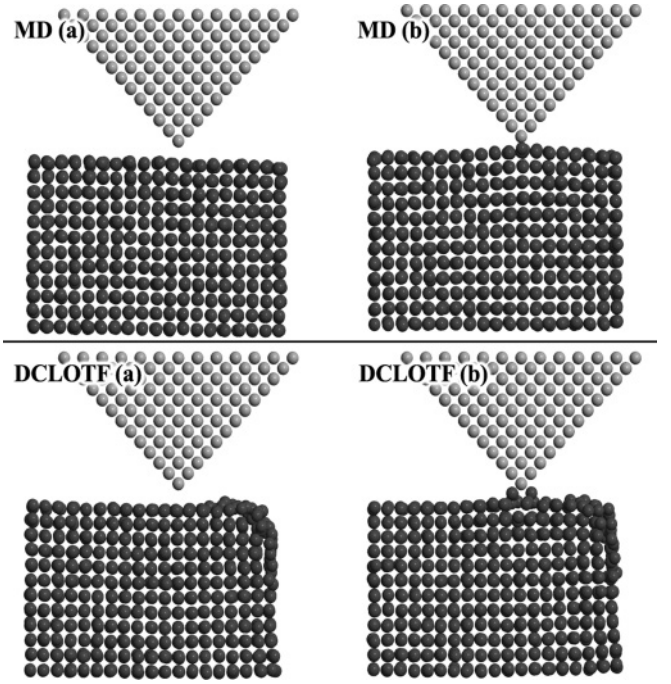

FIG. 15. The subsets of the system for which the single-point energy calculation by NRL-TB was performed, extracted from the MD (top panels) and DCLOTF (bottom panels) configurations immediately after equilibration [panels (a)] and immediately after jump-to-contact [panels (b)]. Each subset comprised approximately 1300 atoms.

Thus, brittle cracking is favored by the hybrid approach for this work-material orientation, while the classical approach predicts a mixture or brittle cracking and amorphization. For the (111)-oriented work material, the hybrid approach predicts initial amorphization, followed by an appearance of a crack [seen in Fig. 12, right panel (f)] only after a significant pile-up of amorphous material. As mentioned earlier, the reference classical simulation for the (111)-oriented work material needs to be discarded.

Closing the discussion of nanoscratching, we will comment briefly on the observed tangential forces experienced by

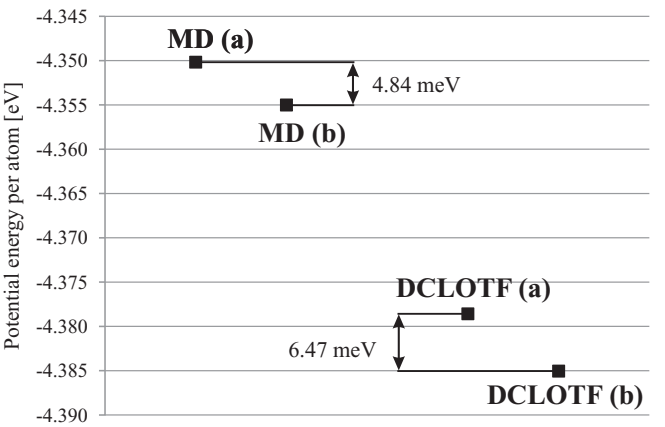

FIG. 16. Absolute potential energies per atom of the configurations shown in Fig. 15, showing that the jump-to-contact variant predicted by DCLOTF offers a more favorable energy decrease.

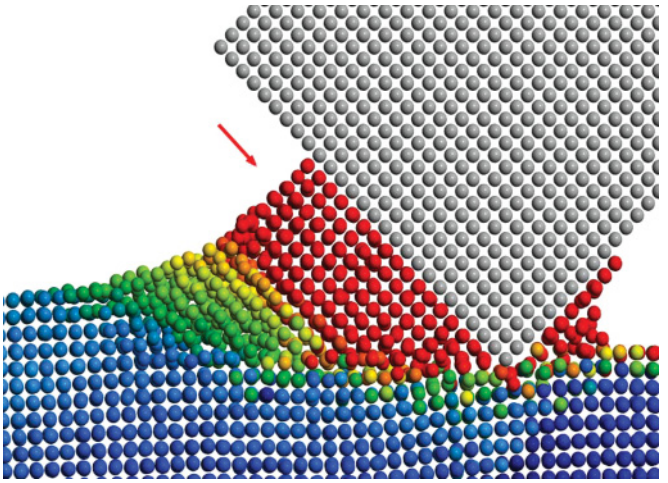

FIG. 17. (Color online) Pile-up of the material in front of the tool, on the example of the reference MD simulation for the (010)-oriented work material. The arrow indicates the piled-up atoms that have adopted the crystalline structure of the tool. In contradistinction to other figures in this work, here the atoms are colored according to their displacement from initial positions, and not according to the magnitude of their slip vectors, making clear the distinction between the newly ordered atoms that have been displaced from far away (red) and atoms just beginning to pile up (green, yellow).

the tool, shown in Fig. 22, with the averages over the duration of the scratching collected in Table I. The qualitative predictions of both approaches are the same. First, we observe that both the classical and the hybrid approach predict the (111)-oriented work material to be most easily scratched,

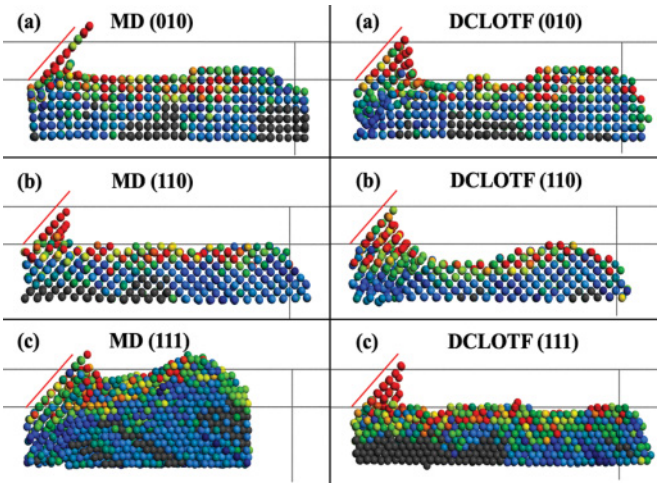

FIG. 18. (Color online) Close-up of the top of the work material after completion of nanoscratching for the reference (MD) simulation (left) and the hybrid (DCLOTF) simulation (right). Panels (a), (b), and (c) correspond to (010)-, (110)-, and (111)-oriented workpieces, respectively. Coloring follows that of Fig. 10. The outline of the tool is shown for clarity. The vertical line in each panel indicates the position of the right-hand surface of the undeformed work material. The upper horizontal line in each panel indicates the position of the relaxed top surface of the undeformed work material, the lower one is positioned at the depth of indentation, $2a$ below, and corresponds to sections labeled "0" in Figs. 19–21.

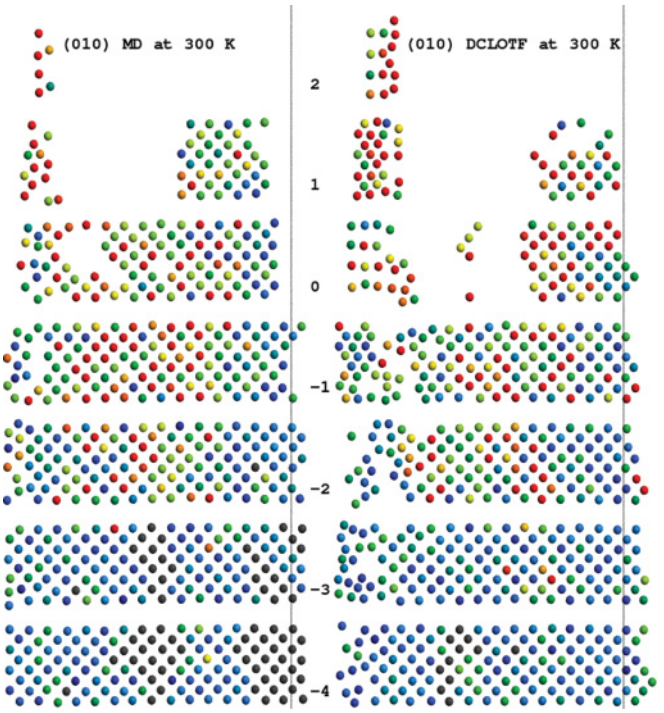

FIG. 19. (Color online) Sections through the top layers of the (010)-oriented work material for the reference (MD) simulation (left) and the hybrid (DCLOTF) simulation (right), corresponding to the situation in Fig. 18, panels (a). The layer corresponding to the indentation depth is labeled "0" and is the layer denoted by the lower horizontal line in Fig. 18, panels (a). Subsequent layers, denoted with negative ordinals, lie below the indentation depth, whereas the layers above the indentation depth are labeled "1" and "2."

which is expected, since it coincides with the favorable slip system for the fcc structure. The (110)-oriented work material is the most difficult to scratch. On the one hand, this agrees with the prediction of Garfinkle *et al.*[49] On the other hand, we explained earlier why this may be overestimated in our model. We note that despite the fact that the two approaches predict a somewhat different course for the nanoscratching, as explained earlier, these differences are not apparent in the graphs of the forces.

Finally, in Table II, we present values for indentation and scratch hardness, along with the friction coefficient, as calculated by the two approaches. First, we note that the predictions of both the classical MD and the DCLOTF models are qualitatively similar. The hybrid technique yields slightly larger values for the indentation hardness, however the predicted trends in all the three quantities are the same—with

TABLE I. Average tangential force experienced by the tool during nanoscratching.

Work-material orientation	Average tangential force (nN)	
	MD	DCLOTF
(010)	13.4	15.0
(110)	17.8	16.8
(111)	12.7	10.1

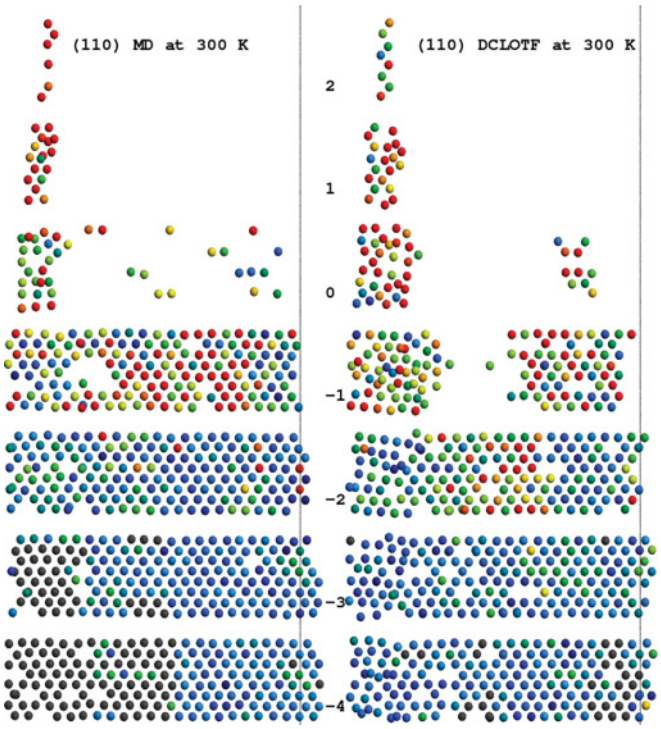

FIG. 20. (Color online) Sections through the top layers of the (110)-oriented work material for the reference (MD) simulation (left) and the hybrid (DCLOTF) simulation (right), corresponding to the situation in Fig. 18, panels (b). The layer corresponding to the indentation depth is labeled "0" and is the layer denoted by the lower horizontal line in Fig. 18, panels (b). Subsequent layers, denoted with negative ordinals, lie below the indentation depth, whereas the layers above the indentation depth are labeled "1" and "2."

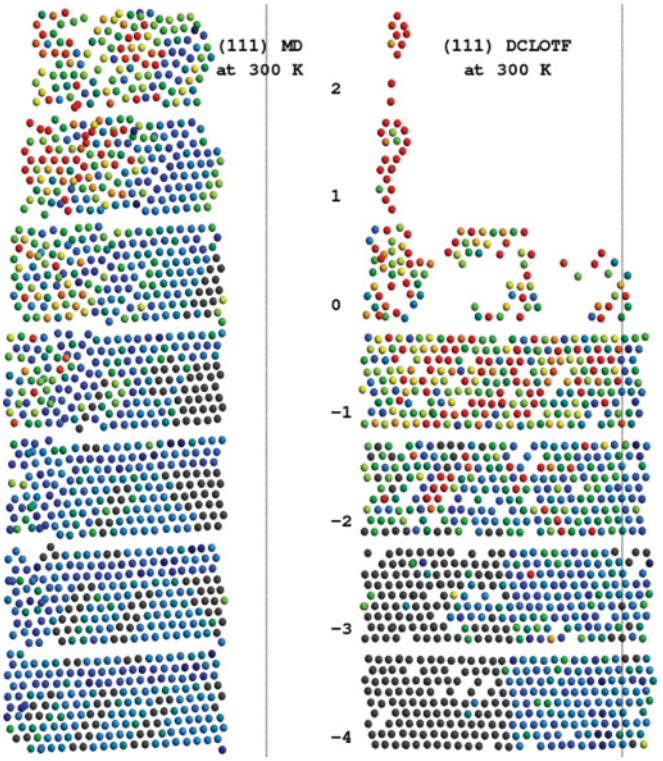

FIG. 21. (Color online) Sections through the top layers of the (111)-oriented work material for the reference (MD) simulation (left) and the hybrid (DCLOTF) simulation (right), corresponding to the situation in Fig. 18, panels (c). The layer corresponding to the indentation depth is labeled "0" and is the layer denoted by the lower horizontal line in Fig. 18, panels (c). Subsequent layers, denoted with negative ordinals, lie below the indentation depth, whereas the layers above the indentation depth are labeled "1" and "2."

(110)-oriented work material being the hardest and (111)-oriented work material being the softest. While Komanduri *et al.*[18] also report that (010)-oriented work material is harder than the work material in the (111) orientation, they predict (110) to be the softest, which we do not observe. On the contrary, as described earlier, we observe the work material in this orientation not yielding readily, as evidenced by the high magnitude of the normal force during indentation. Direct comparison of the obtained hardness values with the results of Komanduri *et al.*[18] is not possible, because the worked material is different. Compared to aluminum, bulk copper is two to four times as hard,[50] depending on the method used to determine hardness. Assuming a similar relation holds for a nanoscale monocrystal or film (which is not unreasonable, since Cu and Al share the fcc structure at ambient conditions), values between 8 and 20 GPa would indicate qualitative agreement with Komanduri *et al.*, which is indeed the case. Experimental data for nanoindentation hardness cannot be compared directly to our results, as hardness is a function of the indentation depth,[51,52] and indentation experiments operate in the regime of tens to hundreds of nm. A rough idea of the expected nanoscale hardness can be obtained by extrapolating the hardness versus indentation depth curves reported by Huo *et al.*[51] and Beegan *et al.*[52] to very small depths. In so doing, one obtains a value of about 6 GPa, which is in moderately good agreement with our observations. Performing a similar extrapolation of the results of Beegan *et al.* (cf. Fig. 7 therein) for the scratching hardness, one obtains a very rough estimate of 10 GPa, which is in agreement with our results. Curiously, Komanduri *et al.* report values of 15–22 GPa for the scratching hardness of Al, which would translate to a scratching hardness in tens of GPa for Cu. As evidenced by Table II, we observe much smaller values. What is more, the values we report for the scratching hardness should be treated as upper bounds, because they do not include the piled-up material in the calculation of the contact surface area, while this pile-up is included in the calculation of the force experienced by the tool, likely increasing it. Finally, we comment on the obtained values of the nanoscratching friction coefficient. The predictions of both approaches are similar and our values are in agreement with those reported by Komanduri *et al.*, except for the unusually high friction coefficient that we observe for the (110)-oriented work material. This is due to the fact that for this orientation, the average normal force experienced by the tool during nanoscratching is extremely small (an order of magnitude smaller than for other orientations, cf. the middle panel of Fig. 13). This, in turn, is caused by the compatibility between the orientation of the indenter and that of the work material. Because of this compatibility and the fact that the indentation depth is a multiple of the lattice constant, we observe that the indenter aligns itself almost perfectly into

TABLE II. Indentation, scratch hardness, and friction coefficient as a function of work-material orientation as predicted by pure MD and the DCLOTF approach.

Work-material orientation	Indentation hardness (GPa)	Scratching hardness (GPa)	Friction coefficient
	MD		
(010)	8.92	12.7	0.83
(110)	11.9	17.0	10.1
(111)	7.83	9.32	1.13
	DCLOTF		
(010)	10.9	14.3	0.59
(110)	18.5	16.0	6.51
(111)	9.19	7.37	0.85

the crystalline structure of the work material, which means it does not experience almost any normal force. We also note that this effect can be responsible for the increased value of the scratching hardness for this orientation, as it is difficult for the tool to plough through the work material after they had been cold-welded together.

IV. CONCLUSIONS AND SUMMARY

We have devised a generalization of the hybrid quantum-classical learn-on-the-fly scheme that is applicable to metallic systems involving many-body potentials. By dividing the workload of the force optimization stage into local optimizations, we were able to perform the required force-fitting for 1400–1900 atoms in reasonable time, allowing for a dynamical simulation of over 100 ps in length. The performance of the method was then significantly improved by carefully designed parallelization and multithreading, which allowed for the utilization of several tens of processor cores.

As a proof of concept, we have presented the results of a set of simulations of nanoindentation and nanoscratching of single-crystal Cu, successfully employing the method to embed the results of a tight-binding calculation within a molecular-dynamics simulation. Our confidence in the proposed technique was furthered by its successful application to the study of liquid Au, where it closely reproduced the results of the underlying TB model, as described elsewhere.[53] In both applications, we found the technique to be well-behaved and free of serious artifacts. Since up until now the cross-scaling

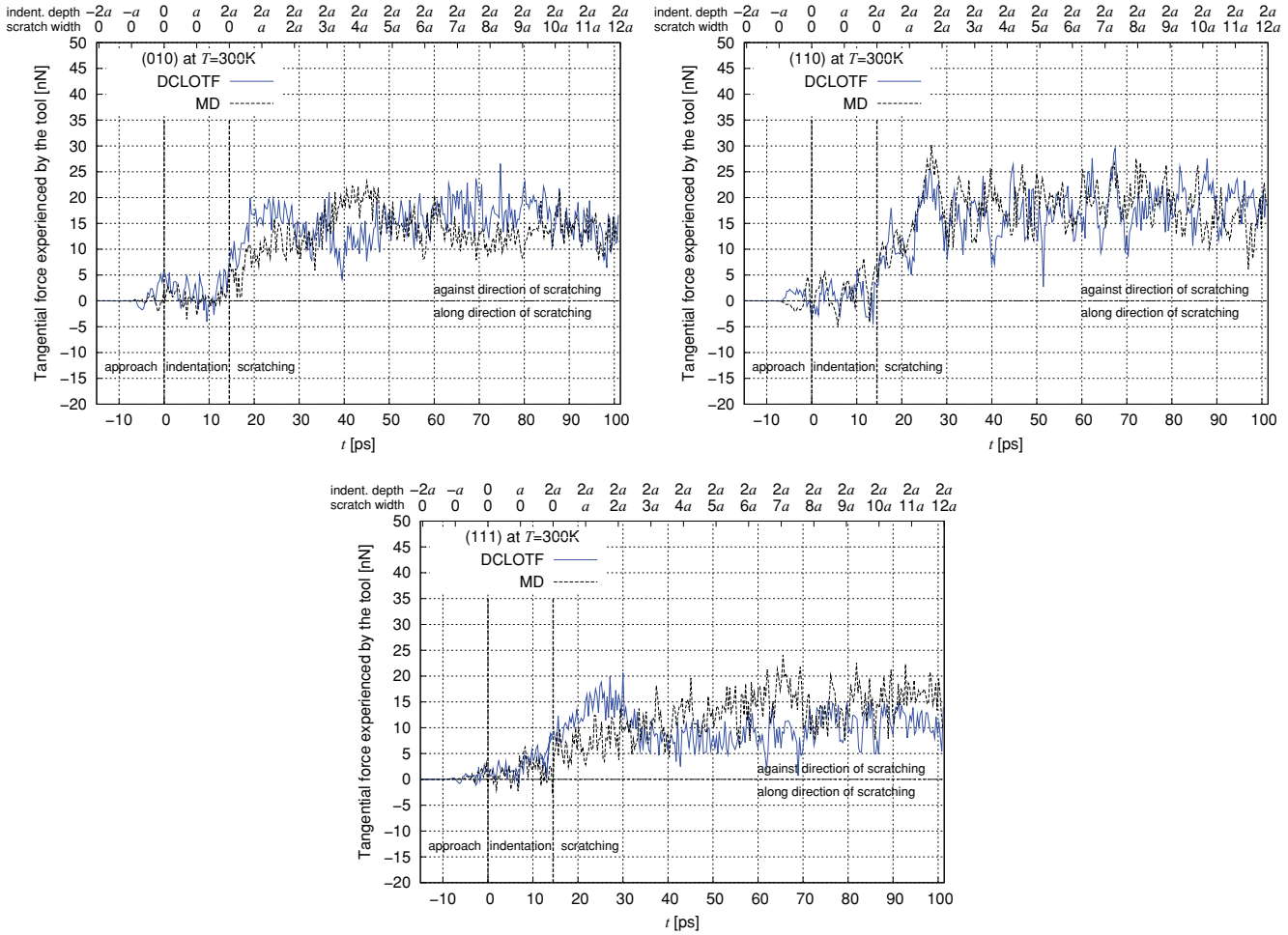

FIG. 22. (Color online) Horizontal (antiparallel to the scratching direction) force experienced by the indenter during the course of the simulation with the (010)-oriented workpiece (top left), (110)-oriented workpiece (top right), and (111)-oriented workpiece (bottom). The solid blue curve refers to the DCLOTF simulation, the dashed black curve to the reference pure MD simulation. The two vertical lines indicate the commencement of indentation and scratching, respectively.

approaches aiming to embed quantum-based calculations within MD simulations have consistently ignored the difficult field of metallic systems, we feel that an important gap in the methodology is now, at least partially, filled.

We have employed a simple model, where the tool is not deformable, and the potential cutoff was modest, as we did not attempt to present a detailed study of nanoindentation or nanoscratching, concentrating instead on the advantages and well-behavedness of the proposed computational technique. Nevertheless, we have shown how the proposed hybrid approach predicts certain effects that are not captured by the fully classical description. The tendency of the empirical potential to overstructure prevented the energetically favorable rounding of the corners of the work material, which, in contradistinction, did take place with both the hybrid and the fully quantum-based approaches. We observed the nature of the jump-to-contact phenomenon to be different between the hybrid and classical approaches, and we confirmed that our prediction does indeed correspond to a lower energy state under the full NRL-TB Hamiltonian. While the classical approach favored amorphous plastic deformation, we have shown how, under certain conditions, brittle fracture can also take place in nanoindented and nanoscratched Cu, when the description of the system is augmented by the use of the proposed DCLOTF approach. Finally, we note that the differing mechanisms of nanoscratching, especially seen in the (110)-oriented system, led to different magnitude of the disorder at the work-material surface behind the tool, which might be of technological importance. However, macroscopic quantities, such as indentation or scratch hardness, were not qualitatively different. Future experimental investigation of nanoindentation and nanoscratching of extremely thin copper films would yield invaluable insight into the detailed microscopic mechanisms involved in plastic deformation at the nanoscale.

ACKNOWLEDGMENTS

The simulations have been performed at the TASK Computer Centre (Gdańsk, Poland). The work has been sponsored by the Polish Ministry of Science and Information Technology under Grants No. N N519 577838 and No. 3 T11F 026 29. The authors would like to thank Professor Krzysztof W. Wojciechowski (Polish Academy of Sciences, Poznan) for fruitful discussions.

*School of Chemistry, University of Southampton, Highfield Campus, SO17 1BJ Southampton, United Kingdom; jaca@kdm.task.gda.pl

[1] D. R. Bowler, M. Aoki, C. M. Goringe, A. P. Horsfield, and D. G. Pettifor, Modell. Simul. Mater. Sci. Eng. **5**, 199 (1997).
[2] C.-K. Skylaris, P. D. Haynes, A. A. Mostofi, and M. C. Payne, J. Chem. Phys. **122**, 084119 (2005).
[3] E. Koch and S. Goedecker, Solid State Commun. **119**, 105 (2001).
[4] S. N. Taraskin, P. A. Fry, X. Zhang, D. A. Drabold, and S. R. Elliott, Phys. Rev. B **66**, 233101 (2002).
[5] N. Bernstein, J. R. Kermode, and G. Csànyi, Rep. Prog. Phys. **72**, 026501 (2009).
[6] G. Csànyi, T. Albaret, M. C. Payne, and A. DeVita, Phys. Rev. Lett. **93**, 175503 (2004).
[7] A. P. Sutton and J. Chen, Philos. Mag. Lett. **61**, 139 (1990).
[8] J. Q. Broughton, F. F. Abraham, N. Bernstein, and E. Kaxiras, Phys. Rev. B **60**, 2391 (1999).
[9] X. P. Long, J. B. Nicholas, M. F. Guest, and R. L. Ornstein, J. Mol. Struct. **412**, 121 (1997).
[10] M. Eichinger, P. Tavan, J. Hutter, and M. Parrinello, J. Chem. Phys. **110**, 10452 (1999).
[11] M. Bobrowski, J. Dziedzic, and J. Rybicki, Phys. Status Solidi B **244**, 842 (2007).
[12] J. Dziedzic, M. Białoskórski, and J. Rybicki, Rev. Adv. Mater. Sci. **14**, 174 (2007).
[13] J. R. Kermode, T. Albaret, D. Sherman, N. Bernstein, P. Gumbsch, M. C. Payne, G. Csànyi, and A. De Vita, Nature (London) **455**, 1224 (2008).
[14] J. R. Kermode, Ph.D. dissertation, Pembroke College, University of Cambridge (2007).
[15] J. R. Kermode, S. Cereda, P. Tangney, and A. De Vita, J. Chem. Phys. **133**, 094102 (2010).
[16] S. Winfield, I. Solt, G. Csànyi, M. Fuxreiter, and M. C. Payne, Conference on Molecular Simulations in Biosystems and Material Science (2008), Abstract No. S8-P14.
[17] S. Winfield (private communication).
[18] R. Komanduri, N. Chandrasekaran, and L. M. Raff, Wear **240**, 113 (2000).
[19] H. Yu, J. B. Adams, and L. G. Hector Jr., Modell. Simul. Mater. Sci. Eng. **10**, 319 (2002).
[20] K. Maekawa and A. Itoh, Wear **188**, 115 (1995).
[21] M. Rychcik-Leyk, Ph.D. dissertation, Gdansk University of Technology, Faculty of Technical Physics and Applied Mathematics (2008), (in Polish); selected fragments (in English) published in TASK Quarterly, the scientific bulletin of the Academic Computer Centre in Gdansk.
[22] R. Komanduri, N. Chandrasekaran, and L. M. Raff, Wear **242**, 60 (2000).
[23] J. Dziedzic, M. Rychcik-Leyk, and J. Rybicki, J. Non-Cryst. Solids **254**, 4309 (2008).
[24] Y. Liu, S. Varghese, J. Ma, M. Yoshino, H. Lu, and R. Komanduri, Int. J. Plasticity **24**, 1990 (2008).
[25] Y. Qi, T. Cagin, Y. Kimura, and W. A. Goddard, Phys. Rev. B **59**, 3527 (1999).
[26] H. Rafii-Tabar and A. P. Sutton, Philos. Mag. Lett. **63**, 217 (1991).
[27] M. Lourakis, *LEVMAR: Implementation of the Levenberg-Marquadt Non-linear Least Squares Algorithm*, [http://lib.stat.cmu.edu/general/levmar.txt].
[28] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, 1st ed. (Addison-Wesley, Reading, USA, 1989).
[29] M. Mitchell, *An Introduction to Genetic Algorithms* (MIT Press, Cambridge, MA, 1998).
[30] J. Dziedzic, Ph.D. dissertation, Gdansk University of Technology, Faculty of Technical Physics and Applied Mathematics (2009); selected fragments published in TASK Quarterly, the scientific bulletin of the Academic Computer Centre in Gdansk.
[31] X. Cao, X. Chi, and M. Gu, *Algorithms and Architectures for Parallel Processing, International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP'02)* (IEEE Computer Society, Los Alamitos, CA, 2002), Vol. 0, p. 0434.

[32]L. Colombo and W. Sawyer, Mater. Sci. Eng. **B37**, 228 (1996).
[33]M. Snir, S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI-The Complete Reference, Volume 1: The MPI Core* (MIT Press, Cambridge, MA, 1998).
[34]W. G. Hoover, Phys. Rev. A **31**, 1695 (1985).
[35]M. J. Mehl and D. A. Papaconstantopoulos, *Topics in Computational Materials Science* (World Scientific, Singapore, 1998), Chap. V, pp. 169–213.
[36]D. A. Papaconstantopoulos, M. J. Mehl, and B. Akdim, in *Proceedings of the International Symposium on Novel Materials*, edited by B. K. Rao (1998), pp. 393–403.
[37]R. E. Cohen, M. J. Mehl, and D. A. Papaconstantopoulos, Phys. Rev. B **50**, 14694 (1994).
[38]M. J. Mehl, D. A. Papaconstantopoulos, N. Kioussis, and M. Herbranson, Phys. Rev. B **61**, 4894 (2000).
[39]F. Kirchhoff, M. J. Mehl, N. I. Papanicolaou, D. A. Papaconstantopoulos, and F. S. Khan, Phys. Rev. B **63**, 195101 (2001).
[40]Y. Mishin, M. J. Mehl, D. A. Papaconstantopoulos, A. F. Voter, and J. D. Kress, Phys. Rev. B **63**, 224106 (2001).
[41]D. A. Papaconstantopoulos, M. Lach-hab, and M. J. Mehl, Physica B **296**, 129 (2001).
[42]A. C. Brańka and K. W. Wojciechowski, Phys. Rev. E **62**, 3281 (2000).
[43]R. L. Davidchack, R. Handel, and M. V. Tretyakov, J. Chem. Phys. **130**, 234101 (2009).
[44]J. A. Zimmerman, C. L. Kelchner, P. A. Klein, J. C. Hamilton, and S. M. Foiles, Phys. Rev. Lett. **87**, 165507 (2001).
[45][http://simgroup.task.gda.pl/nanoindentation.html].
[46]H. Rafii-Tabar, Phys. Rep. **325**, 239 (2000).
[47]S. Ciraci, A. Baratoff, and I. P. Batra, Phys. Rev. B **42**, 7618 (1990).
[48]J. R. Smith, G. Bozzolo, A. Banerjea, and J. Ferrante, Phys. Rev. Lett. **63**, 1269 (1989).
[49]M. Garfinkle and R. Garlick, Trans. Metall. Soc. AIME **242**, 809 (1968).
[50]G. V. Samsonov, *Handbook of the Physicochemical Properties of the Elements* (IFI-Plenum, New York, 1968).
[51]Y. L. D. Huo and K. Cheng, Proc. Inst. Mech. Eng.: Pt. C (J. Mech. Eng. Sci.) **221**, 259 (2007).
[52]S. C. D. Beegan and M. T. Laugier, Surf. Coat. Technol. **201**, 5804 (2007).
[53]J. Dziedzic and J. Rybicki (unpublished, to appear in JNCS).