IOP Publishing Modelling and Simulation in Materials Science and Engineering
Modelling Simul. Mater. Sci. Eng. **29** (2021) 055018 (16pp) https://doi.org/10.1088/1361-651X/ac01b9

Highly efficient parallel grand canonical simulations of interstitial-driven diffusion-deformation processes

S Sina Moeini-Ardakani[1], S Mohadeseh Taheri-Mousavi[2] and Ju Li[3,*]

[1] Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, United States of America
[2] Department of Mechanical Engineering and Department of Materials Science and Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, United States of America
[3] Department of Nuclear Science and Engineering and Department of Materials Science and Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, United States of America

E-mail: liju@mit.edu

Received 13 February 2021, revised 15 April 2021
Accepted for publication 14 May 2021
Published 10 June 2021

Abstract

Absorption of interstitial alloying elements like H, O, C, and N in metals and their continuous relocation and interactions with various microstructural features such as vacancies, dislocations, and grain boundaries have crucial influences on metals' properties. However, besides limitations in experimental tools in capturing these mechanisms, the inefficiency of numerical tools also inhibits modeling efforts. Here, we present an efficient framework to perform hybrid grand canonical Monte Carlo and molecular dynamics simulations that allow for parallel insertion/deletion of Monte Carlo moves. A new methodology for calculation of the energy difference at trial moves that can be applied to many-body potentials as well as pair ones is a primary feature of our implementation. We study H diffusion in Fe (ferrite phase) and Ni polycrystalline samples to demonstrate the efficiency and scalability of the algorithm and its application. The computational cost of using our framework for half a million atoms is a factor of 250 less than the cost of using existing libraries.

Keywords: interstitial alloy, grand canonical Monte Carlo, molecular dynamics, hybrid framework

*Author to whom any correspondence should be addressed.

S Supplementary material for this article is available online

(Some figures may appear in colour only in the online journal)

1. Introduction

H-embrittlement, oxidation, creep, and carbide precipitation are examples of life-limiting chemo-mechanical mechanisms of metallic alloys in service. While continuum theories can capture the deformation mechanism phenomenologically, they have a limited range of validity when the inherent interactions between deformation and interstitial atoms are unspecified. On the other hand, simulations at atomic resolution can reveal the underlying mechanisms and their correlations, but are severely limited in computational efficiency.

Molecular dynamics (MD) simulations have been widely used for equilibrium and non-equilibrium processes, while they have limitations in capturing long time-scale mechanisms such as diffusion. Monte Carlo (MC) simulations, besides their convenience in implementation, cannot capture non-equilibrium deformation mechanisms. The combination of these two techniques seems to be ideal. However, in practice, the probabilistic nature of the MC scheme and also the evolving number of degrees of freedom complicate the efficient implementation, i.e. parallelization of the hybrid MC-MD framework on multi-core architectures.

Diffusion simulations have been conducted by various frameworks at atomistic scale. These frameworks are classified into two main categories depending on whether the kinetics of diffusion is directly handled in the simulation or not. If the actual movements of atoms are not handled, the outcome of diffusion, which is mainly captured by thermodynamics and semi-classical statistical physics, is of concern. For the former case, the kinetic MC technique is the most viable approach [1–4]. For the latter case, depending on the type of alloying elements and the target thermodynamic properties, different ensembles are considered. For example, in the case of a fixed percentage of substitutional alloying elements [5], or glass transition [6, 7], MC simulations for swapping of pair atom types are performed. For a constant chemical potential of substitutional alloying elements, semi-grand canonical MC (SGCMC) simulations are used [8, 9]. For interstitial alloying elements [10, 11], mainly grand canonical MC (GCMC) simulations, which allow for insertion or deletion of these elements for a given chemical potential/partial pressure, are conducted. It is noteworthy that in all these simulations, the indistinguishability of particles is accounted for, while other quantum effects are ignored ($\hbar \to 0$). Combinations of all these various MC techniques with MD allow modeling diffusion-deformation problems at atomistic scale, while the computational efficiency of these hybrid frameworks remained the most critical challenge to be addressed.

During the last decade, there have been multiple attempts at parallelization of MC. Sadigh *et al* [12] designed and proposed a novel parallelization scheme capable of performing simultaneous MC moves based on domain decomposition. Yamakov [13] implemented parallel MD and SGCMC simulations using this algorithm to efficiently model the behavior of substitutional alloys. The objective of this work is to develop and implement a new GCMC algorithm that is as scalable as MD, while being capable of addressing non-pair and many-body potentials.

Here, we first present the theoretical framework and reformulate the grand canonical ensemble. Then, we present a newly developed software package that, among many features, has a new hybrid GCMC based on the new formulation to model interstitial induced deformation processes. In addition, we introduce a more generalized concept of the linked list algorithm [14] that can be utilized to greatly improve the performance of the GCMC algorithm. Performing isothermal H-charging and discharging in a Ni model material will show that our library has two orders of magnitude less computational cost than Large-scale Atomic/Molecular

Massively Parallel Simulator (LAMMPS) [15]. Moreover, being able to conduct simulations at polycrystalline scale reveals the unknown parameters of analytic formulas to extract the concentration-pressure relationship of different microstructural defects in a model material system.

2. Methodology

2.1. Theory

2.1.1. Proof of detailed balance. Consider a simulation supercell with a fixed total volume V, temperature T, and a chemical potential μ of an isotope of mass m. Even though the treatment here is for a monatomic system, it can be trivially generalized to multiple chemical species (isotopes), with constant $\{\mu_c, m_c\}$ for $c = 1 \ldots C$ nuclide species. In the semi-classical treatment, the grand partition function $\mathcal{Q}$ can be written for N identical particles of position vector $\mathbf{x}^{3N} \equiv \mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_N$ as follows [16]:

$$\mathcal{Q}(T, \mu, V) \equiv \sum_{N=0}^{\infty} \int \frac{\mathrm{d}\mathbf{x}^{3N}}{N!\lambda_{\mathrm{th}}^{3N}} \mathrm{e}^{\beta\left(N\mu - U\left(\mathbf{x}^{3N}\right)\right)}, \tag{1}$$

where $\beta \equiv (k_{\mathrm{B}}T)^{-1}$, $\lambda_{\mathrm{th}} \equiv h/\sqrt{2\pi m k_{\mathrm{B}} T}$ is the thermal de Broglie wavelength, and $U\left(\mathbf{x}^{3N}\right)$ denotes the potential energy of the system. The differential probability of finding the system at a phase-space volume $\left(N, \mathbf{x}^{3N}\right)$ is therefore:

$$p(N, \mathbf{x})\,\mathrm{d}\mathbf{x}^{3N} = \frac{\mathrm{d}\mathbf{x}^{3N}}{N!\lambda_{\mathrm{th}}^{3N}} \mathrm{e}^{\beta\left(\mathcal{G} + \mu N - U\left(\mathbf{x}^{3N}\right)\right)}, \tag{2}$$

where $\mathcal{G} \equiv -k_{\mathrm{B}}T \log \mathcal{Q}$ is the grand potential. Note that if we consider the particle-index permutation symmetry in $\mathbf{x}^{3N} \equiv \mathbf{x}_1\mathbf{x}_2 \ldots \mathbf{x}_N$, there are $N!$ copies of this phase-space volume with exactly the same U and therefore the same probability density. (2) Represents just one of these copies in a particular differential volume $\mathrm{d}\mathbf{x}^{3N} = \mathrm{d}\mathbf{x}_1\,\mathrm{d}\mathbf{x}_2 \ldots \mathrm{d}\mathbf{x}_N$, namely nuclide 1 in $(\mathbf{x}_1, \mathbf{x}_1 + \mathrm{d}\mathbf{x}_1)$, nuclide 2 in $(\mathbf{x}_2, \mathbf{x}_2 + \mathrm{d}\mathbf{x}_2)$, …, nuclide N in $(\mathbf{x}_N, \mathbf{x}_N + \mathrm{d}\mathbf{x}_N)$.

Metropolis MC [17] relies on transition rates that respects detailed balance. If we perform particle insertion with some transition $\left(N, \mathbf{x}^{3N} \to N+1, \mathbf{x}^{3N+3}\right)\mathrm{d}\mathbf{x}^{3N+3}$ rate where the proportionality to destination volume $\mathrm{d}\mathbf{x}^{3N+3}$ is made explicit, leaving the transition $\left(N, \mathbf{x}^{3N} \to N+1, \mathbf{x}^{3N+3}\right)$ function itself an intensive quantity with finite value, then the corresponding reciprocal deletion rate must satisfy

$$\frac{\text{transition}\left(N, \mathbf{x}^{3N} \to N+1, \mathbf{x}^{3N+3}\right)\mathrm{d}\mathbf{x}^{3N+3}}{\text{transition}\left(N+1, \mathbf{x}^{3N+3} \to N, \mathbf{x}^{3N}\right)\mathrm{d}\mathbf{x}^{3N}} = \frac{p\left(N+1, \mathbf{x}^{3N+3}\right)\mathrm{d}\mathbf{x}^{3N+3}}{p\left(N, \mathbf{x}^{3N}\right)\mathrm{d}\mathbf{x}^{3N}}, \tag{3}$$

where the right-hand side (rhs) is the ratio of the resident probabilities that we desire to approach, and the left-hand side is the ratio of the conditional transition probabilities. It is then clear that differential phase-space volumes $\mathrm{d}\mathbf{x}^{3N+3}$, $\mathrm{d}\mathbf{x}^{3N}$ be canceled out from both sides, making the exact values of these infinitesimal quantities immaterial, as they should be.

In principle, there needs to be no relation between $\mathbf{x}^{3N}$ and $\mathbf{x}^{3N+3}$; in other words, the positions of all atoms can be changed, even by a lot, in one move. But in the simplest incarnation that preserves detailed balance, we choose to preserve almost all of the atomic positions except for the atom (or a position) in question:

$$\mathbf{x}^{3N+3} = \mathbf{x}^{3N}\mathbf{x}_{\text{question}} \tag{4}$$

in which case

$$\mathrm{d}\mathbf{x}^{3N+3} = \mathrm{d}\mathbf{x}^{3N}\,\mathrm{d}\mathbf{x}_{\text{question}}, \tag{5}$$

and (3) is simplified to

$$\frac{\text{transition}\left(N, \mathbf{x}^{3N} \to N+1, \mathbf{x}^{3N+3}\right)\mathrm{d}\mathbf{x}_{\text{question}}}{\text{transition}\left(N+1, \mathbf{x}^{3N+3} \to N, \mathbf{x}^{3N}\right)} = \frac{\mathrm{e}^{\beta(\mu-\Delta U)}\,\mathrm{d}\mathbf{x}_{\text{question}}}{(N+1)\lambda_{\text{th}}^{3}} \tag{6}$$

where

$$\Delta U \;\equiv\; U\left(\mathbf{x}^{3N+3}\right) - U\left(\mathbf{x}^{3N}\right) \tag{7}$$

is always the energy difference between the high-particle-number configuration and the low-particle-number configuration.

There is an index permutation issue, though, about exactly what (6) means. We know there are $N!$ copies of a particular differential volume hypercube $\mathrm{d}\mathbf{x}^{3N} = \mathrm{d}\mathbf{x}_1\,\mathrm{d}\mathbf{x}_2 \ldots \mathrm{d}\mathbf{x}_N$ that are energetically degenerate, permuting only the position of indistinguishable atoms. Similarly, we know that $(N+1)!$ copies of the differential volume hypercube $\mathrm{d}\mathbf{x}^{3N+3} = \mathrm{d}\mathbf{x}_1\,\mathrm{d}\mathbf{x}_2 \ldots \mathrm{d}\mathbf{x}_{N+1}$. Are we allowing MC transitions between any pair of them, i.e. a total of $N!(N+1)!$ transitions in labelled-atom space, or are we only adding/deleting the last atom in the labelled-atom space without random permutation afterwards, i.e. in total only $(N+1)N! = (N+1)!$ MC transition bridges? Since with the same computational cost, building more MC bridges facilitate approaching equilibrium, the first interpretation ($N!(N+1)!$ transitions) is preferable. Thus, if we take $\{N, \mathbf{x}^{3N}\}$ to mean the 'index-free' collection of all $N!$ degenerate copies, where every copy in the $\{N, \mathbf{x}^{3N}\}$ set automatically share the weight in the collective, then (6) is simplified to

$$\frac{\text{transition}\left(\{N, \mathbf{x}^{3N}\} \to \{N+1, \mathbf{x}^{3N+3}\}\right)\mathrm{d}\mathbf{x}_{\text{question}}}{\text{transition}\left(\{N+1, \mathbf{x}^{3N+3}\} \to \{N, \mathbf{x}^{3N}\}\right)} = \frac{\mathrm{e}^{\beta(\mu-\Delta U)}\,\mathrm{d}\mathbf{x}_{\text{question}}}{\lambda_{\text{th}}^{3}}, \tag{8}$$

where the $\{N, \mathbf{x}^{3N}\}$ notation is philosophically closer to the quantum mechanical interpretation for identical particles.

Requirement (8) is not that different from the standard MC for the canonical ensemble. Given one is at $N, \mathbf{x}^{3N}$, one can attempt to insert (probability a^+)/accept insertion, or attempt to delete ($a^- = 1 - a^+$)/accept deletion ([18]). Both types of attempts would involve computational cost, and rejection of either type of attempts would mean wasted computations, and therefore a^+/a^- may be chosen to optimize performance, i.e. speed of approaching chemical equilibrium, and efficiency in computing the thermodynamic averages of measurables.

Within the a^+ attempt branch, there is a question of where to insert. Again in the spirit of the simplest incarnation, we can choose 'anywhere in the supercell, equally', and therefore the attempt probability to $\mathrm{d}\mathbf{x}_{\text{question}}$ is $(a^+/V)\mathrm{d}\mathbf{x}_{\text{question}}$, representing a spatially uniform prior. It does not have to be this way. If we have advanced screening information, we could use umbrella sampling to tune this attempt probability (indeed, with the domain decomposition scheme to come later, such issue could arise). But right now let us choose the simplest insertion prior.

Within the a^- attempt branch, we can also choose 'any of the atoms in the supercell, equally'. It does not have to be this way. If we have advanced screening information, we could may umbrella sampling to tune this attempt probability also. But the a^-/N prior (or $a^-/(N+1)$ for the high-particle-number configuration) does lead to the simplest proof of detailed balance.

Therefore, with these simplest insertion/deletion priors, the requirement (8) is converted to

$$\frac{(a^+/V)\text{acceptance}\left(\{N,\mathbf{x}^{3N}\}\to\{N+1,\mathbf{x}^{3N+3}\}\right)}{(a^-/(N+1))\text{acceptance}\left(\{N+1,\mathbf{x}^{3N+3}\}\to\{N,\mathbf{x}^{3N}\}\right)}=\frac{\mathrm{e}^{\beta(\mu-\Delta U)}}{\lambda_{\text{th}}^3}, \tag{9}$$

where $\mathrm{d}\mathbf{x}_{\text{question}}$ is canceled out, or

$$\frac{\text{acceptance}\left(\{N,\mathbf{x}^{3N}\}\to\{N+1,\mathbf{x}^{3N+3}\}\right)}{\text{acceptance}\left(\{N+1,\mathbf{x}^{3N+3}\}\to\{N,\mathbf{x}^{3N}\}\right)}=\frac{a^-V\,\mathrm{e}^{\beta(\mu-\Delta U)}}{a^+(N+1)\lambda_{\text{th}}^3}. \tag{10}$$

The metropolis MC dichotomy [17] is then used to achieve (10) literally, by making one of the acceptance $\left(\{N,\mathbf{x}^{3N}\}\to\{N+1,\mathbf{x}^{3N+3}\}\right)$, acceptance $\left(\{N+1,\mathbf{x}^{3N+3}\}\to\{N,\mathbf{x}^{3N}\}\right)$ unity, and the other $\leqslant 1$, depending on the sign of the rhs. So the standard GCMC algorithm is just

$$\text{acceptance}\left(\{N,\mathbf{x}^{3N}\}\to\{N+1,\mathbf{x}^{3N+3}\}\right)=\begin{cases}1, & \dfrac{a^-V\,\mathrm{e}^{\beta(\mu-\Delta U)}}{a^+(N+1)\lambda_{\text{th}}^3}\geqslant 1,\\ \dfrac{a^-V\,\mathrm{e}^{\beta(\mu-\Delta U)}}{a^+(N+1)\lambda_{\text{th}}^3}, & \dfrac{a^-V\,\mathrm{e}^{\beta(\mu-\Delta U)}}{a^+(N+1)\lambda_{\text{th}}^3}<1,\end{cases} \tag{11}$$

and

$$\text{acceptance}\left(\{N+1,\mathbf{x}^{3N+3}\}\to\{N,\mathbf{x}^{3N}\}\right)=\begin{cases}1, & \dfrac{a^+(N+1)\lambda_{\text{th}}^3}{a^-V\,\mathrm{e}^{\beta(\mu-\Delta U)}}\geqslant 1,\\ \dfrac{a^+(N+1)\lambda_{\text{th}}^3}{a^-V\,\mathrm{e}^{\beta(\mu-\Delta U)}}, & \dfrac{a^+(N+1)\lambda_{\text{th}}^3}{a^-V\,\mathrm{e}^{\beta(\mu-\Delta U)}}<1,\end{cases} \tag{12}$$

or equivalently,

$$\text{acceptance}\left(\{N,\mathbf{x}^{3N}\}\to\{N-1,\mathbf{x}^{3N-3}\}\right)=\begin{cases}1, & \dfrac{a^+N\lambda_{\text{th}}^3}{a^-V\,\mathrm{e}^{\beta(\mu-\Delta U)}}\geqslant 1,\\ \dfrac{a^+N\lambda_{\text{th}}^3}{a^-V\,\mathrm{e}^{\beta(\mu-\Delta U)}}, & \dfrac{a^+N\lambda_{\text{th}}^3}{a^-V\,\mathrm{e}^{\beta(\mu-\Delta U)}}<1,\end{cases} \tag{13}$$

where ΔU is always the energy difference between the high-particle-number configuration and the low-particle-number configuration. It is seen from above that a^+/a^-, $1/V$ and $1/N$ are just *choices*, reflecting the simplest prior about how to make moves. There is nothing set in stone about them. As long as we use these choices *consistently*, detailed balance can be established.

It is clear from the derivations above that the $0<a^+<1$ insertion attempt rate and $a^-=1-a^+$ deletion attempt rate can be arbitrarily tuned, and also the $(a^+/V)\mathrm{d}\mathbf{x}_{\text{question}}$ and a^-/N priors can be substantially changed. Indeed, our domain decomposition scheme below changes these prefactors of MC sampling for each domain, as $N_{\text{domain}}/V_{\text{domain}}$ can vary from domain to domain, and does not have to be equal to the average N/V.

2.1.2. Domain decomposition. Imagine that our supercell is spatially partitioned into $d=1\ldots D$ separate domains. Analytically, one can rearrange (1) into the following form

$$\mathcal{Q}(T,\mu,V)=\mathcal{I}_1\ldots\mathcal{I}_D\,\exp\left[-\beta U\left(\mathbf{x}^{3N_1},\ldots,\mathbf{x}^{3N_d}\right)\right], \tag{14}$$

where

$$\mathcal{I}_d\equiv\sum_{N_d=0}^{\infty}\int_{V_d}\frac{\mathrm{e}^{\beta\mu N_d}}{N_d!\lambda_{\text{th}}^{3N_d}}\,\mathrm{d}\mathbf{x}^{3N_d}, \tag{15}$$

is an integration and summation operator. The $1/N_d!$ prefactor comes from the combinatoric $N!/N_1!N_2!\ldots N_D!$ copies of assigning labelled but identical nuclides to different domains, that yields the same integral for (1). Namely, in the way (1) was written, all particles can traverse and live in all domains, but now in the new form (14), within each $\mathcal{I}_d$ only 'citizens' of that domain can live and contribute to the integral. Therefore the probability density of a microstate while ignoring the labelling of particles inside each domain is:

$$\bar{p}\left(\{N_1, \mathbf{x}^{3N_1}\}, \ldots, \{N_D, \mathbf{x}^{3N_D}\}\right) = \mathrm{e}^{\beta\mathcal{G}} \left(\prod_{d=1}^{D} \frac{\mathrm{d}\mathbf{x}^{3N_d}}{\lambda_{\mathrm{th}}^{3N_d}} \, \mathrm{e}^{\beta\mu N_d} \right) \exp\left[-\beta U\left(\mathbf{x}_1, \ldots, \mathbf{x}_D\right)\right]. \quad (16)$$

Now let us modify the standard GCMC algorithm by changing the priors. Every time that an insertion or deletion attempt trial is to be performed, one of the domains, say d is randomly chosen with a probability P_d:

$$\sum_{N_d=0}^{\infty} P_d = 1, \quad (17)$$

P_d can be chosen to be proportional to its volume V_d, for example. Other priors may be chosen, but one should be careful in not letting P_d depending on N_d, which can dynamically change, that unless proven, may break the detailed-balance requirement. A time-constant $\{P_d\}$ distribution should always be fine, if computational efficiency is of no concern.

The domain-d insertion/deletion attempt rate is therefore $P_d a_d^+ / P_d a_d^-$, and we can straightforwardly show that

$$\text{acceptance}\left(\{N_d, \mathbf{x}^{3N_d}\} \to \{N_d + 1, \mathbf{x}^{3N_d+3}\}\right) = \begin{cases} 1, & \dfrac{a_d^- V_d \, \mathrm{e}^{\beta(\mu-\Delta U)}}{a_d^+ (N_d+1)\lambda_{\mathrm{th}}^3} \geqslant 1, \\ \dfrac{a_d^- V_d \, \mathrm{e}^{\beta(\mu-\Delta U)}}{a_d^+ (N_d+1)\lambda_{\mathrm{th}}^3}, & \dfrac{a_d^- V_d \, \mathrm{e}^{\beta(\mu-\Delta U)}}{a_d^+ (N_d+1)\lambda_{\mathrm{th}}^3} < 1, \end{cases} \quad (18)$$

and

$$\text{acceptance}\left(\{N_d, \mathbf{x}^{3N_d}\} \to \{N_d - 1, \mathbf{x}^{3N_d-3}\}\right) = \begin{cases} 1, & \dfrac{a_d^+ N_d \lambda_{\mathrm{th}}^3}{a_d^- V_d \, \mathrm{e}^{\beta(\mu-\Delta U)}} \geqslant 1, \\ \dfrac{a_d^+ N \lambda_{\mathrm{th}}^3}{a_d^- V_d \, \mathrm{e}^{\beta(\mu-\Delta U)}}, & \dfrac{a_d^+ N_d \lambda_{\mathrm{th}}^3}{a_d^- V_d \, \mathrm{e}^{\beta(\mu-\Delta U)}} < 1, \end{cases} \quad (19)$$

where ΔU is always the energy difference between the high-particle-number configuration and the low-particle-number configuration, and may depend on nearby domains, would preserve detailed balance. While (18) and (19) is factually a different algorithm from (11) and (13), their derivation follows exactly the same logic flow of the previous section, just with a *different set of screening priors* for making the next move. That is, in (18) and (19) all 'citizens' and all volume elements of the same 'country' (domain) are treated the same before the 'testing' (evaluation) of the potential ('presumed innocent'), but they are treated differently from country to country, whereas in (11) and (13) all 'citizens' and volume elements of the 'world' (supercell) are treated equally before the testing of the potential. Even though (18) and (19) is factually a different algorithm from (11) and (13), as long as being run consistently, both can provide the

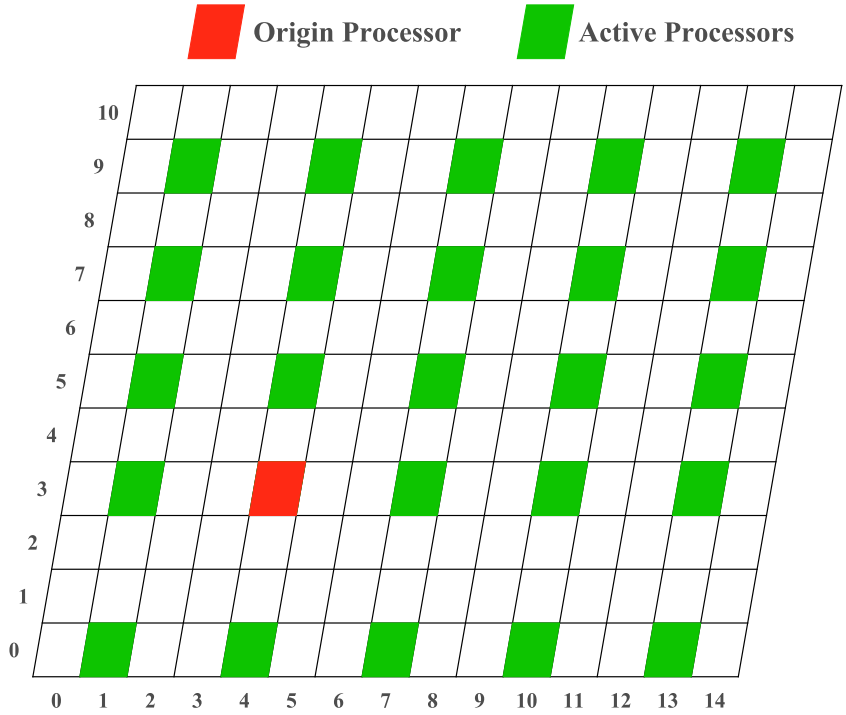

Figure 1. Schematic representation of subdomain selection pattern in two dimensions for $\vec{n} = \{15, 11\}$ and $\vec{m} = \{2, 1\}$. Red and green subdomains denote origin and active processors, respectively.

equilibrium ensemble distribution (2). So in this sense, the domain partition of the supercell and the $\{P_d, a_d^+/a_d^-\}$ are just 'gauge' choices. These gauge choices, however, can be used to enhance the computational efficiency, because it is also clear from the logic flow of the proof that the 'global citizen' approach of (11) and (13) has no reason to possess optimal efficiency. Thus $\{V_d, P_d, a_d^+/a_d^-\}$ can be optimized, and when V_d is taken to be quite small, it is clear that the P_d spatial distribution would amount to a screening prior in umbrella sampling. While we do not explore this degree of freedom fully here in this paper, taking the simplest uniform V_d, P_d approach, the connection between domain decomposition and umbrella sampling is noted.

In addition to statistical sampling efficiency, there is the critical issue of efficient interatomic potential evaluation and load balancing, well-known in parallel computing for discrete agent-based simulations with short-range interactions. To this end, we break down a large supercell to smaller individual domains, whose size is chosen according to the radial cutoff distance in the interatomic potential and possible range of dynamic strain in the supercell [19, 20]. If the atoms in two separate domains cannot possibly interact with one another, energy difference would only depend on the affected domain and nothing prohibits us from performing D simultaneous moves. Although the aforementioned assumption is almost never the case, it is possible to choose a pattern of domains such that the changes of potential energy due to any perturbation in the domains are independent of one another.

In the limit of truly isolated, non-interacting domains, it is clear that (14) will become a product of domain-specific grand partition functions. Thus, (18) and (19) can be interpreted as running chemical equilibration between the constant-μ reservoir with each domain inde-

pendently, using the classic GCMC algorithm. For load balancing purposes, the frequency of attempting to equilibrate these different domains may not need to be the same. In problems involving multi-phase chemical equilibrium, for instance, one may have a vapor phase with very different N_d/V_d with a solid phase that it should be in equilibrium with. The freedom to pick $P_d, a_d^+/a_d^-$ is in effect an umbrella-sampling scheme based on spatial location for load-balancing and sampling efficiency. For detailed balance, $P_d, a_d^+/a_d^-$ should not depend on N_d turn-by-turn, as N_d sustains microscopic fluctuations. But by our derivations above, one should be able to re-adjust $P_d, a_d^+/a_d^-$, say after every 1000 acceptances in that domain. One may also re-partition the supercell and redefine the domains for computational efficiency and expediency (say in a parallel computer, the number of allocated computing nodes may be forced to change from time to time), as long as this is not done too frequently.

2.2. Pattern selection algorithm

As mentioned in the previous section, it is possible to construct a pattern that involves non-interacting domains. Such a pattern can be established by considering that these domains have to be at least $n \times r_C$ apart in any direction, where r_C is the cutoff radius of the interatomic potential and n is an integer determined by the type of the forcefield (see [12]). For example, for pair potentials, $n = 1$; for embedded atom method (EAM), $n = 2$; and for modified embedded atom method (MEAM), $n = 3$. It is noteworthy that as domains are chosen based on this parameter, increasing the number of processors never changes the accuracy of simulations. Here, we describe how to construct such a pattern given that the domains are predetermined by the MD part of the GCMC + MD method.

Suppose that our supercell is spanned by $n_0 \times n_1 \times n_2$ processors. Each processor controls a domain determined by a domain decomposition method. Figure 1 depicts a simplified two dimensional schematic of such a supercell. Our first task is to determine how far apart our active domains should be. In other words, we must determine how many domains in each direction will at least cover $n \times r_C$, namely integer vector $\vec{m}$. The best case scenario would be $\vec{m} = \{1, 1, 1\}$. In figure 1, $\vec{m} = \{2, 1\}$. Let us define another integer vector

$$s_i = \lfloor n_i/(m_i + 1) \rfloor, \quad i = 0, 1, 2, \tag{20}$$

where $\lfloor . \rfloor$ is the floor function. $\vec{s}$ defines the number of simultaneous trial moves in each direction at any given GCMC step. Therefore, the total number of simultaneous trial moves is $s_0 \times s_1 \times s_2$.

At every step of GCMC, a processor is chosen at random. This processor will serve as the 'origin,' and its position in the domain grid is denoted by an integer vector $\vec{o}$ (red domain in figure 1). Now, all the processors that perform trial moves, namely 'active,' have to be determined (green domains in figure 1).

Consider a processor whose position in the domain grid is denoted by the integer vector $\vec{p}$. This processor is active if and only if

$$\sum_{i=0}^{2} [(p_i + n_i - o_i) \bmod n_i] \bmod (m_i + 1) = 0 \tag{21}$$

where 'mod' denotes a modulo operation. The first modulo operation takes into account the periodic boundary condition. The second one ensures that active domains are non-interacting. Once the active processors are determined, they will perform trial moves simultaneously.

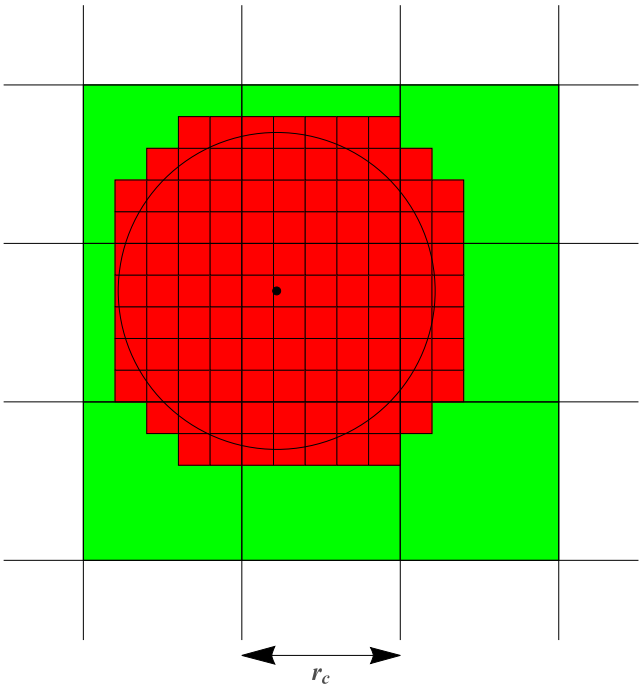

Figure 2. Classical linked list algorithm (green) versus proposed extension (red).

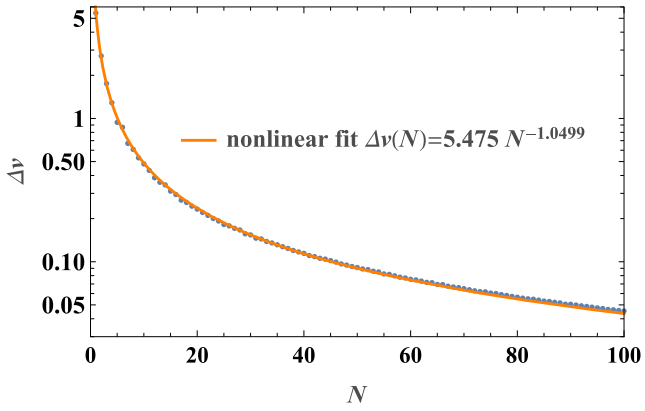

Figure 3. Relative excess search volume (Δv) with respect to number of discretization of cutoff radius (N).

2.3. Potential energy difference calculation

As was pointed out earlier, the most expensive part of GCMC or any kind of atomistic simulation is the potential energy evaluation. This is due to the fact that potential energy and forces depend on the interaction of atoms. However, in the case of short-range force fields, the interactions of an atom with its surrounding can be restricted to a volume within a cutoff radius. In MD, all the pairs that are within a specified cutoff radius are included in a sparse matrix, usually

referred to as a neighborlist. The forcefield employs the neighborlist to calculate energy/forces of the system. Since the generation of the neighborlist is itself time-consuming, researchers have come up with two major algorithms to speed things up: the cell/linked list [14] and the Verlet algorithm [21].

The cell/linked list algorithm is employed to limit the search volume for finding all the pairs interacting with an atom. The Verlet algorithm makes it possible to generate neighborlists less often.

In the case of GCMC, due to insertion or deletion moves, the maintenance of a neighborlist has extra complication. Nonetheless, a properly designed linked list will be relatively simple to update. In our implementation of parallel GCMC, every processor has its own linked list. Whenever a trial move is to be performed, a neighborlist for the affected atom will be instantly generated using the linked list and passed to the forcefield to calculate the energy difference. Next, the energy difference will determine whether the move is accepted or rejected. If the move is accepted, the linked list will be updated, accordingly.

In order to speed up even further, we modified the linked list algorithm. In the traditional linked list algorithm, the supercell is gridded by cubes (cells) with an edge of length r_C. A linked list would be generated to link all the atoms within a cell. The basic premise of the algorithm is that when the neighborlist of an atom inside one of these cells is to be generated, only the atoms in the surrounding cells need to be searched, and these are accessible via the said linked list. In three dimensions, the search volume will be decreased from the volume of the whole supercell V to $27r_C^3$.

It is possible to reduce the search volume even further. In our implementation the cell's edge length is r_C/m, where m is a positive integer to be set by the user, see figure 2. For two cells to interact, the minimum possible distance between them must be smaller than or equal to r_C. Suppose cell Ω_i is labeled by a 3D vector $\vec{s}_i$, which determines its position in the grid and the volume it covers. In other words, $\vec{x} \in \Omega_i$ if

$$s_{i_\alpha} \frac{r_C}{m} \leqslant x_\alpha < \left(s_{i_\alpha} + 1\right) \frac{r_C}{m}, \quad \alpha = 0, 1, 2. \tag{22}$$

One can easily show that for cells Ω_i and Ω_j to be interacting, the following must be true:

$$\sum_{\alpha=0}^{2} \left[\min\left(|s_{i_\alpha} - s_{j_\alpha}| - 1, 0\right)\right]^2 \leqslant m^2. \tag{23}$$

Using this relationship, a 'relative' neighborlist for the cells is created and will be utilized to build the neighborlist of a specific atom in one of the cells. Figure 3 shows relative excess volume Δv vs the number of cutoff discretizations. Here, relative excess volume is defined by the ratio of search volume to the volume of the sphere with a radius r_C minus one. As the number of discretizations tends to infinity, the excess volume will tend to zero. One might naively conclude that the higher number of discretizations must inevitably lead to better performance. However, this improving trend is true only up to a point. The downside of increasing the number of discretizations is an increase in the memory storage of the head array of the linked list. If the storage becomes too large, it can lead to cache pollution and indeed poor performance. Much like the Verlet algorithm, in which the size of the shell must be chosen according to the problem under study, the number of discretizations can be chosen empirically by trial and error.

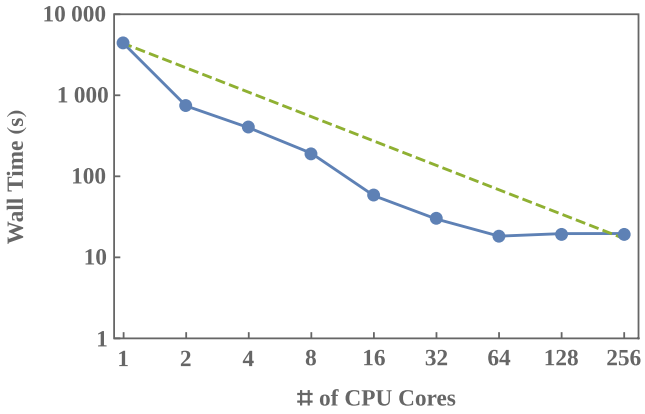

Figure 4. Total wall time as function of number of processors for 10^6 MC trial moves of H in ferrite.

3. Results

3.1. Scalability tests

To demonstrate the performance and the scalability of our implementation, several benchmark simulations were performed. All of these tests were conducted on a Beowulf Linux cluster, where each of the computational nodes contained two Intel Xeon Gold 6248 processors with 20 cores. The code was compiled using a C++ GNU compiler and $-$O3 optimization flag.

These benchmarks simulated H absorption in a ferrite-phase Fe single crystal. A sample consisting of $64 \times 64 \times 64$ Fe body-centered-cubic unit cells was generated. To facilitate the insertion of H, 0.1% of atoms were randomly removed. The total number of Fe (ferrite phase) atoms is approximately 524 000. The EAM interatomic potential developed by Ramasubramaniam *et al* [22] was used to model atomic interactions.

The sample was in equilibrium with a reservoir that had a -2.4242 eV H chemical potential. The temperature was 300 K; in total, 10^6 MC insertion/deletion trials were conducted.

The tests were performed on 1, 2, 4, 8, 16, 32, 64, 128, and 256 CPU cores. For each given number of processors, 4 tests were performed, and the average wall time was recorded as the result. Figure 4 shows the total CPU wall-time with respect to the number of processors, with the green dashed line being the ideal scalability line. Overall, the trend looks like a typical domain decomposition parallel application.

The biggest decrease in the computational time is the transition from the serial execution (1 core) to the parallel execution on 2 cores. However, this decrease is not due to parallel MC moves. In fact, the parallel MC moves will only take place starting from 16 processors. The main reason for the reduction in the computational time below the 16 cores is the reduction of cache pollution. In other words, due to the reduction of the number of atoms per processor, fewer cache misses would occur in the energy calculation.

The second largest drop takes place on transitioning from 8 to 16 processors, i.e. the onset of parallel MC moves. The improvement in computational time continues in a consistent manner up to 64 processors, reaching the start of a plateau. Increasing the number of processors does not enhance the performance any further. In fact, it is evident from the plot that the performance even slightly suffers. This slight decrease in performance can be explained by the overhead in inter-processor communication, which is the bottleneck of performance beyond this point.

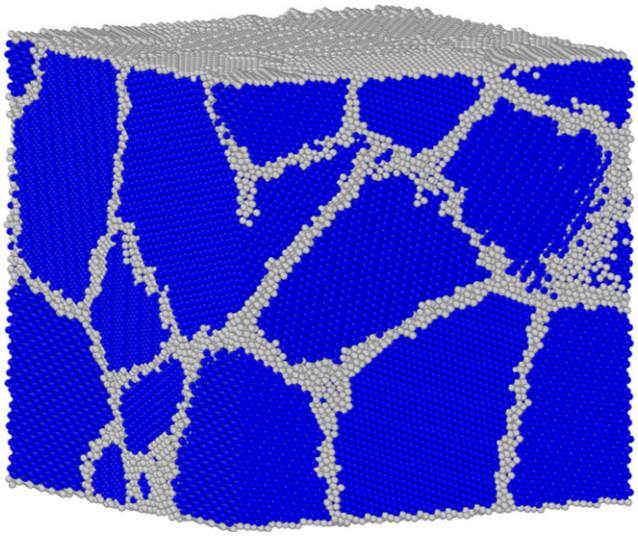

Figure 5. Polycrsytalline Ni sample with 8 randomly oriented grains for hybrid GCMC/MD simulations.

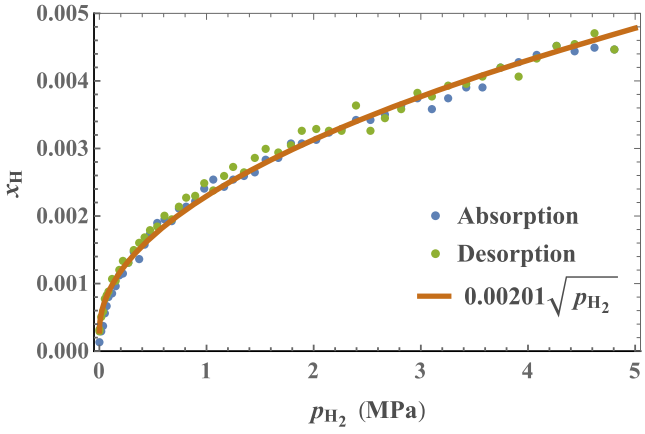

Figure 6. Isothermal curve for H absorption/desorption in the bulk of Ni polycrystalline sample. H concentration has a square root relationship with H pressure and follows Sieverts' law. H mainly locates in grain boundaries.

Finally, we compared the performance of our code against LAMMPS [15] on 64 cores. The same configuration, chemical potential, and interatomic potential were used in both tests. Like the other benchmarks, this test was repeated 4 times, and the results were averaged. On average, our code is faster by a factor of about 250. The reasons for such staggering differences are twofold. Firstly, LAMMPS cannot perform MC moves simultaneously. Secondly, LAMMPS is ill-equipped when it comes to calculating the energy differences resulting from MC moves of atoms with non-pair potentials. Currently, at each MC move, the whole neighbor-list is rebuilt, and the potential energy of the whole system is calculated and subtracted from the previous value.

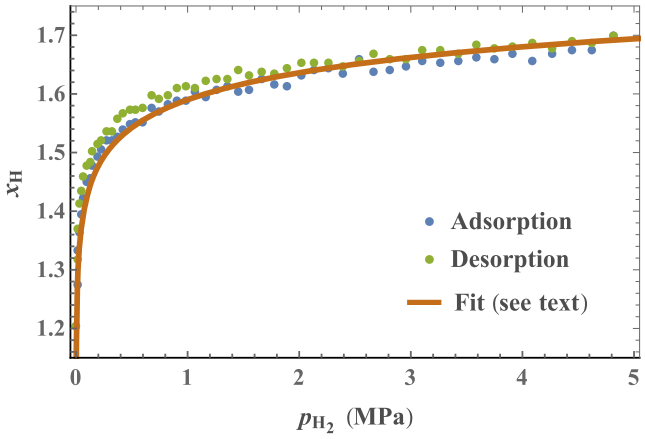

Figure 7. Isothermal curve for H absorption/desorption on the surface of Ni polycrystalline sample. H atom interactions influence the H concentration on the surface.

3.2. Isothermal H absorption/desorption in a Ni polycrystalline sample with free surfaces

As an example of application of the code, we chose isothermal H absorption/desorption in a polycrystalline Ni sample with free surfaces. This example was used in a previous work to study the mobility of dislocations due to H charging [23]. The polycrystalline sample was constructed using Voronoi tessellation to produce eight grains. A random crystal orientation was assigned to each Voronoi cell. The single crystal Ni sample was rotated accordingly and put in each grain volume. Periodic boundary conditions were considered for atoms located on sample edges. Our sample has dimensions of $15 \times 15 \times 15$ nm^3 and is periodic in the x and y directions. A 2.5 nm vacuum layer was considered in the z direction. Thus, the sample contains about 300 000 atoms in eight randomly oriented grains, which are approximately 7.5 nm in diameter (see figure 5). For Ni and Ni–H interatomic interactions, the EAM potential [24] was adapted. Prior to absorption, a Nosé–Hoover thermostat was used to maintain the isothermal–isostress ensemble at room temperature $T = 300$ K and zero stress for 5 ns. After stress relaxation, the sample was relaxed in a μVT ensemble using hybrid GCMC + MD for 2 fs with a time step of 0.5 fs. The chemical potential of H was set to $\mu_0 = -2.547$ eV. After every 1000 steps of MD, 10 000 GCMC trials were conducted. Thereafter, H chemical potential was varied from μ_0 to $\mu_1 = -2.447$ eV in 50 increments of equal pressure change $(p_1 - p_0)/50$ from $p_0 \propto \exp(2\mu_0/(k_B T))$ to $p_1 \propto \exp(2\mu_1/(k_B T))$. At each increment, hybrid GCMC + MD was conducted for 200 ps, using the scheme described above. After the chemical potential reached μ_1, the procedure was reversed to return to the initial chemical potential. Figures 6 and 7 show the evolution of the H-concentration in the bulk and the free surface of the polycrystalline Ni. No hysteresis was observed in either case. In the case of bulk atoms, H mainly locates in grain boundaries. As demonstrated, H concentration inside the bulk has a square root relation with pressure and obeys Sieverts' law (see supplementary discussion 1 for the derivation of the equation (https://stacks.iop.org/MSMS/29/055018/mmedia)).

However, for the surface, the story is different. Due to the high concentration of H on the surfaces, the ideal solution model with no H–H interactions, i.e. the basis of Sieverts' law, is no longer valid. However, employing the regular solution model for 2D, i.e. the Fowler–Guggenheim adsorption isotherm model with lateral interaction between H* and H*, where* means surface site, can capture the behavior. We can define species A and B of regular

solution as a surface site being occupied by H and a vacancy, respectively. The excess Gibbs free energy per site due to mixing is:

$$\Delta g\left(x\right) = n_{\mathrm{s}} k_{\mathrm{B}} T \left[\frac{x}{n_{\mathrm{s}}}\ \log\left(\frac{x}{n_{\mathrm{s}}}\right) + \left(1 - \frac{x}{n_{\mathrm{s}}}\right) \log\left(1 - \frac{x}{n_{\mathrm{s}}}\right)\right] + n_{\mathrm{s}} w \frac{x}{n_{\mathrm{s}}}\left(1 - \frac{x}{n_{\mathrm{s}}}\right). \tag{24}$$

Here x and n_{s} denote the concentration of H and number of H sites per Ni atom, respectively. w is the effective interaction energy of the regular model

$$w = w_{\mathrm{AB}} - \frac{w_{\mathrm{AA}} + w_{\mathrm{BB}}}{2}. \tag{25}$$

Therefore,

$$\mu_{\mathrm{H}} = \frac{\partial \Delta g}{\partial x}|_{p,T} = k_{\mathrm{B}} T\ \log\left(\frac{x/n_{\mathrm{s}}}{1 - x/n_{\mathrm{s}}}\right) + w\left(1 - 2\frac{x}{n_{\mathrm{s}}}\right). \tag{26}$$

As shown in details in supplementary discussion 1, since H is a diatomic gas, we can assume that its pressure is proportional to $\exp\left(2\beta\mu_{\mathrm{H}}\right)$, leading to

$$p_{\mathrm{H}_2}\left(x\right) = C\left(\frac{x/n_{\mathrm{s}}}{1 - x/n_{\mathrm{s}}}\right)^2 \mathrm{e}^{-4\beta w x/n_{\mathrm{s}}}. \tag{27}$$

Based on the concentration curves, we conclude that $n_{\mathrm{s}} = 2$. Figure 7 shows that the fit is in excellent agreement with the values obtained from the simulations. Based on our fitting, the effective interaction was calculated to be almost zero for H in grain boundaries (Langmuir–McLean isotherm), and $w = -0.22$ eV for H on free surfaces.

4. Discussion

In conclusion, we present a hybrid GCMC/MD framework that can efficiently simulate interstitial solid solution behavior in large polycrystalline samples. We show that the parallelization of the code is necessary for samples with large numbers of atoms. We provide two applied case studies for H-absorption/desorption in a polycrystalline Ni sample. Although currently only Lennard-Jones and EAM potentials are implemented in the package, the implementation is general; it can easily be extended to any other pair or non-pair potentials such as bond order potentials and the MEAM potentials. As noted in the *pattern selection algorithm* section, n parameter, which defines the distance between active processors by $n \times r_{\mathrm{C}}$, is to be set with care by the developer in the source code. Otherwise the accuracy will suffer, leading to unreliable results. Our analytical analysis was an excellent match with the obtained numerical results. The hidden parameters in the theory (H-interaction energy) can now be extracted from our efficient library. The framework has broad applications for simulation of interstitial alloying elements such as C, H, and O in different alloying systems and provides a new pathway to study the diffusion-deformation mechanisms in these samples.

All data required to reproduce the findings during this study are included in this manuscript and supplementary information. The code is available for downloading at the code repository https://github.com/sinamoeini/mapp4py. In the *examples* directory of the code repository, multiple examples and guides, including two tutorials, are provided for interested users to get started. In addition, the scripts used for testing scalability and hydrogen absorption/desorption in Ni polycrystalline samples are provided in the noted directory.

Author contributions

SSMA and JL designed the project. SSMA, JL and SMTM performed the analytical derivations. SSMA wrote the code. SSMA and SMTM tested the code and ran the examples. SSMA, SMTM and JL wrote the paper.

Competing interests

The authors declare no competing interests.

Acknowledgments

The authors acknowledge the financial support from Timken (to SSMA) and the Swiss National Science Foundation through Grant P300P2_171423 (to SMTM) and the U.S. Department of Energy (DOE) Fuel Cell Technologies Office under award No. DE-EE0008830 (to JL). Discussions with Doug Smith and R Scott Hyde are gratefully appreciated. The simulations reported were performed on a local high-performance cluster. The authors also acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing (HPC and consultation) resources that have contributed to some of the numerical results reported within this paper.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

ORCID iDs

S Mohadeseh Taheri-Mousavi https://orcid.org/0000-0002-4124-3709
Ju Li https://orcid.org/0000-0002-7841-8058

References

[1] Mao Z, Sudbrack C K, Yoon K E, Martin G and Seidman D N 2007 The mechanism of morphogenesis in a phase-separating concentrated multicomponent alloy *Nat. Mater.* **6** 210–6
[2] Daniels C and Bellon P 2020 Hybrid kinetic Monte Carlo algorithm for strongly trapping alloy systems *Comput. Mater. Sci.* **173** 109386
[3] Vattré A, Jourdan T, Ding H, Marinica M-C and Demkowicz M J 2016 Non-random walk diffusion enhances the sink strength of semicoherent interfaces *Nat. Commun.* **7** 10424
[4] De Backer A *et al* 2015 Modeling of helium bubble nucleation and growth in austenitic stainless steels using an object kinetic Monte Carlo method *Nucl. Instrum. Methods Phys. Res.* B **352** 107–14
[5] Widom M, Huhn W P, Maiti S and Steurer W 2014 Hybrid Monte Carlo/molecular dynamics simulation of a refractory metal high entropy alloy *Metall. Mater. Trans.* A **45** 196–200
[6] Cavagna A, Grigera T S and Verrocchio P 2012 Dynamic relaxation of a liquid cavity under amorphous boundary conditions *J. Chem. Phys.* **136** 204502
[7] Berthier L, Charbonneau P, Coslovich D, Ninarello A, Ozawa M and Yaida S 2017 Configurational entropy measurements in extremely supercooled liquids that break the glass ceiling *Proc. Natl Acad. Sci. USA* **114** 11356–61

[8] Mishin Y 2004 Atomistic modeling of the γ and γ'-phases of the Ni–Al system *Acta Mater.* **52** 1451–67
[9] Ueno K and Shibuta Y 2019 Semi-grand canonical Monte Carlo simulation for derivation of thermodynamic properties of binary alloy *IOP Conf. Ser.: Mater. Sci. Eng.* **529** 012037
[10] Diarra M, Amara H, Ducastelle F and Bichara C 2012 Carbon solubility in nickel nanoparticles: a grand canonical Monte Carlo study *Phys. Status Solidi b* **249** 2629–34
[11] Gai L, Shin Y K, Raju M, van Duin A C T and Raman S 2016 Atomistic adsorption of oxygen and hydrogen on platinum catalysts by hybrid grand canonical Monte Carlo/reactive molecular dynamics *J. Phys. Chem.* C **120** 9780–93
[12] Sadigh B *et al* 2012 Scalable parallel Monte Carlo algorithm for atomistic simulations of precipitation in alloys *Phys. Rev.* B **85** 184203
[13] Yamakov V I 2016 Parallel grand canonical Monte Carlo (paragrandMC) simulation code *Technical Report* NASA/CR-2016-219202, NF1676L-24373 National Institute of Aerospace, Hampton, Virginia
[14] Allen M P and Tildesley D J 2017 *Computer Simulation of Liquids* 2nd edn (Oxford: Oxford University Press)
[15] Plimpton S 1995 Fast parallel algorithms for short-range molecular dynamics *J. Comput. Phys.* **117** 1–19
[16] Kardar M 2007 *Statistical Physics of Particles* 1st edn (Cambridge: Cambridge University Press)
[17] Metropolis N, Rosenbluth A W, Rosenbluth M N, Teller A H and Teller E 1953 Equation of state calculations by fast computing machines *J. Chem. Phys.* **21** 1087–92
[18] O'Keeffe C, Ren R and Orkoulas G 2007 Spatial updating grand canonical Monte Carlo algorithms for fluid simulation: generalization to continuous potentials and parallel implementation *J. Chem. Phys.* **127** 194103
[19] Li J 2005 Basic molecular dynamics *Handbook of Materials Modeling* ed S Yip (Berlin: Springer) pp 565–88
[20] Li J 2005 Atomistic calculation of mechanical behavior *Handbook of Materials Modeling* ed S Yip (Berlin: Springer) pp 773–92
[21] Verlet L 1967 Computer 'experiments' on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules *Phys. Rev.* **159** 98–103
[22] Ramasubramaniam A, Itakura M and Carter E A 2009 Interatomic potentials for hydrogen in α-iron based on density functional theory *Phys. Rev.* B **79** 174101
[23] Koyama M, Taheri-Mousavi S M, Yan H, Kim J, Cameron B C, Moeini-Ardakani S S, Li J and Tasan C C 2020 Origin of micrometer-scale dislocation motion during hydrogen desorption *Sci. Adv.* **6** eaaz1187
[24] Angelo J E, Moody N R and Baskes M I 1995 Trapping of hydrogen to lattice defects in nickel *Modelling Simul. Mater. Sci. Eng.* **3** 289