A fully automated approach to calculate the melting temperature of elemental crystals

Zhu, Li Fang; Janssen, Jan; Ishibashi, Shoji; Körmann, Fritz; Grabowski, Blazej; Neugebauer, Jörg

DOI
10.1016/j.commatsci.2020.110065

Publication date
2021

Document Version
Final published version

Published in
Computational Materials Science

Citation (APA)
Zhu, L. F., Janssen, J., Ishibashi, S., Körmann, F., Grabowski, B., & Neugebauer, J. (2021). A fully automated approach to calculate the melting temperature of elemental crystals. *Computational Materials Science*, *187*, Article 110065. https://doi.org/10.1016/j.commatsci.2020.110065

Computational Materials Science 187 (2021) 110065

Contents lists available at ScienceDirect

Computational Materials Science

journal homepage: www.elsevier.com/locate/commatsci

A fully automated approach to calculate the melting temperature of elemental crystals

Li-Fang Zhu [a,*], Jan Janssen [a], Shoji Ishibashi [a,b], Fritz Körmann [a,c], Blazej Grabowski [d], Jörg Neugebauer [a]

[a] Max-Planck-Institut für Eisenforschung GmbH, Max-Planck-Str. 1, 40237 Düsseldorf, Germany
[b] National Institute of Advanced Industrial Science and Technology, Tsukuba 305-8568, Japan
[c] Department of Materials Science and Engineering, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands
[d] Institute of Materials Science, University of Stuttgart, Pfaffenwaldring 55, 70569 Stuttgart, Germany

ARTICLE INFO

Keywords:
Interface method
Melting point
Arbitrary potential
pyiron

ABSTRACT

The interface method is a well established approach for predicting melting points of materials using interatomic potentials. However, applying the interface method is tedious and involves significant human intervention. The whole procedure involves several successive tasks: estimate a rough melting point, set up the interface structure, run molecular dynamic calculations and analyze the data. Loop calculations are necessary if the predicted melting point is different from the estimated one by more than a certain convergence criterion, or if full melting/solidification occurs. In this case monitoring the solid–liquid phase transition in the interface structure becomes critical. As different initial random seeds for the molecular dynamic simulations within the interface method induce slightly different melting points, a few ten or hundred interface method calculations with different random seeds are necessary for performing a statistical analysis on these melting points. Considering all these technical details, the work load for manually executing and combining the various involved scripts and programs quickly becomes prohibitive. To simplify and automatize the whole procedure, we have implemented the interface method into pyiron (http://pyiron.org). Our fully automatized procedure allows to efficiently and precisely predict melting points of stable unaries represented by arbitrary potentials with only two user-specified parameters (interatomic potential file and element). For metastable or dynamically unstable unary phases, the crystal structure needs to be provided as an additional parameter. We have applied our automatized approach on fcc Al, Ni, dynamically unstable bcc Ti and hcp Mg and employed a large set of available interatomic potentials. Melting points for classical interatomic potentials of these metals have been obtained with a numerical precision well below 1 K.

1. Introduction

The melting point of a material is a highly relevant quantity. For example, it is one of the key quantities in searching for high-temperature stable refractory materials [1,2]. In contrast to material quantities, such as elastic constants or lattice constants, the melting point is not only numerically but also computationally much harder to compute. The method to determine the melting point at constant pressure is by finding the temperature at which the Gibbs energy of the solid and liquid phase are equal. There are two popular approaches for calculating the melting point. One is the free energy approach [3], which needs to explicitly calculate the Gibbs energies of the solid and liquid phase, e.g., by using thermodynamic integration. The melting point is then determined by the crossing point of solid and liquid Gibbs energies. An alternative approach simulates solid and liquid coexistence and is known as the interface method [4] (also called "coexistence approach"). These two approaches were originally introduced and carried out with classical interatomic potentials. The corresponding potentials were either fitted to the experimental data or data from *ab initio* calculations. Calculations based on empirical potentials are generally very efficient even for large systems and long simulation times. However, the accuracy of the computed melting temperature strongly relies on the quality of the interatomic potentials. To simulate both the liquid and the solid phase with a single potential a high transferability of the potential is critical.

[*] Corresponding author.
E-mail address: l.zhu@mpie.de (L.-F. Zhu).

https://doi.org/10.1016/j.commatsci.2020.110065
Received 4 June 2020; Received in revised form 19 August 2020; Accepted 11 September 2020
Available online 1 October 2020

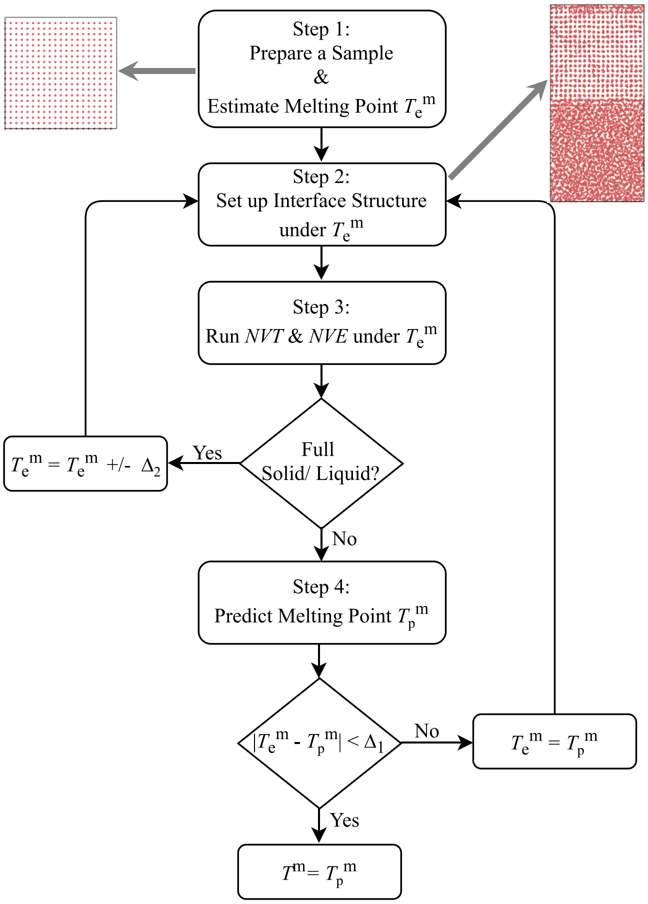
Step 1:
Prepare a Sample
&
Estimate Melting Point T_e^m
Step 2:
Set up Interface Structure
under T_e^m
Step 3:
Run *NVT* & *NVE* under T_e^m
Full
Solid/ Liquid?
Yes
$T_e^m = T_e^m$ +/- Δ_2
No
Step 4:
Predict Melting Point T_p^m
$|T_e^m - T_p^m| < \Delta_1$
No
$T_e^m = T_p^m$
Yes
$T^m = T_p^m$

Fig. 1. Schematic representation of the various steps that have to be performed in the interface method. If the difference between the predicted and the estimated melting point, $|T_e^m - T_p^m|$, is larger than the convergence criterion Δ_1, a further loop over Step 2 to 4 using the newly estimated melting point, $T_e^m = T_p^m$, is necessary. If the entire cell becomes either completely solid or liquid in Step 3, the estimated melting point needs to be increased/decreased by Δ_2 and a new loop using this newly estimated melting point is performed. For the calculations in the present study we use $\Delta_1 = 1$ K and $\Delta_2 = 0.05 \times T_e^m$.

Thus, utilizing parameter-free *ab initio* calculations based on these two approaches has become highly attractive to determine melting temperatures of materials. For example, based on the interface method Alfè proposed either to use an on-top *ab initio* free energy correction to obtain the required accuracy [5], or to directly perform the interface method with *ab initio* calculations [6]. Another recent development is the interface pinning (IP) method proposed by Pedersen et al. [7]. The basic idea of this method is to compute the average force required to pin the interface of a two-phase system via a harmonic bias potential. This approach was validated with a Lennard-Jones model and later applied to calculate the melting temperatures of Na, Mg, Al and Si from *ab initio*. A very recent approach based on the free energy method is the TOR-TILD methodology (*Two-Optimized Reference Thermodynamic Integration using Langevin Dynamics*) [8,9]. It has been applied to calculate the melting properties of Al, Cu and Ni. Even though these methodologies provide *ab initio* accuracy, they generally require expensive computing power and expert user knowledge. Therefore, for scientists who want to select an empirical potential for their specific materials and are interested in the melting temperature represented by this potential, or for those who aim to construct interatomic potentials with good performance on high temperature melting properties, the conventional interface method using an interatomic potential is the method of choice and widely used due to its implementational simplicity, numerical efficiency and its straightforward theoretical background [10,11].

The basic principle of the interface method is to find the temperature at which solid and liquid coexist in thermodynamic equilibrium. During the simulation the interface between the solid and liquid acts as the nucleation site for melting or solidification. If the temperature of the system is slightly below the melting point, part of the liquid phase will tend to solidify and generate some heat which will heat up the system to the melting point. Vice versa, if the temperature of the system is above the melting point, part of the solid phase will absorb some heat and tend to melt, which will cool down the system to the melting point. Simultaneously, the pressure of the system will also tend to equilibrate and the whole system will quickly evolve to an equilibrium phase.

However, applying the interface method in practical calculations is not straightforward. The procedure of how to apply the interface method simulation has been documented, e.g. in the user guide of KISSMD (Kinetic Simulation System based on molecular dynamics) [12]. Many computational steps and technical details need to be carefully executed. Fig. 1 shows the workflow of the interface method. It comprises the following steps: 1) prepare a sample and estimate a rough melting point, T_e^m, 2) set up the interface structure under T_e^m, 3) successively perform molecular dynamic (MD) simulations with the canonical ensemble (*NVT*) and the micro-canovical ensemble (*NVE*) at T_e^m on the interface structures under different strains (here normal strain perpendicular to the interface is performed while keeping the interface area unchanged), and 4) predict the melting point by extracting the temperature–pressure dependence from the *NVE* simulations on the strained interface structures, where the temperature at $P = 0$ GPa is the predicted melting point, T_p^m. Besides these successive steps, there are a few obstacles that hamper the manual application of the interface method. First, during the MD simulations full solidification or full melting has to be avoided. This is achieved by monitoring the interface structures. If full solidification/melting occurs, the procedure from step 2) to 3) needs to be repeated with a higher/lower estimated temperature. Second, if the difference between the predicted melting point, T_p^m, and the estimated melting temperature, T_e^m, is larger than a certain convergence criterion, a further loop from step 2) to 4) needs to be performed with a new temperature estimate $T_e^m = T_p^m$. The loop is continued until the criterion is reached. Third, as loop calculations cannot be avoided, providing a good first estimate in step 1) is critical to reduce the number of loops. Fourth, as the predicted melting point is extracted from the temperature–pressure dependence, to obtain a high numerical precision of the computed melting point it is crucial to ensure that the temperatures and pressures during *NVE* simulations are fully converged. Therefore, it is mandatory to permanently monitor the temperature and pressure convergence as function of the number of MD steps. Finally, to guarentee statistically converged results the MD simulations need to be performed with different initial random seeds to provide different initial atomic velocities. These small differences result in small changes in target quanlities such as the solid–liquid volume ratio or melting points.

Considering all these technical details, having to manually set up the interface structure, perform the calculations and do the analysis, results in a work load that is not feasible for routine or high-throughput calculations. To speed up the application of the interface method, the whole procedure needs to be optimized and automatized. pyiron [13] is an integrated development environment (IDE) for computational material science. It is specially designed to automatize such routine tasks. pyiron allows to interactively implement and test simulation protocols and it provides all necessary tools to enable rapid prototyping of complex simulation protocols. We have therefore implemented the interface method in pyiron and applied the automatized procedure on fcc Al, Ni, bcc Ti and hcp Mg. To test the performance and robustness of the approach we consider three conceptually very different potentials. One is a “standard” embedded atom model (EAM) from literatures [14,15] (referred to as “litEAM” in the present work). The litEAM has high transferability and is well fitted to most of the important properties such

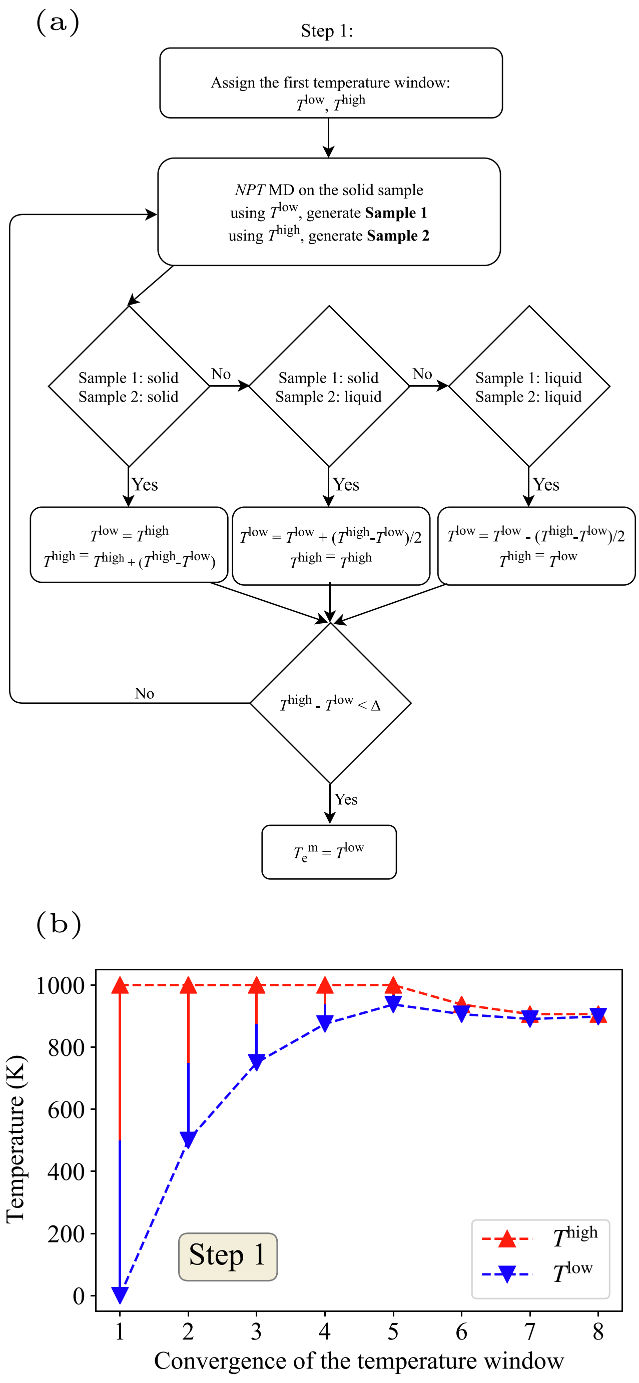
(a)
Step 1:
Assign the first temperature window:
T^{low}, T^{high}
NPT MD on the solid sample
using T^{low}, generate **Sample 1**
using T^{high}, generate **Sample 2**
Sample 1: solid
Sample 2: solid
No
Sample 1: solid
Sample 2: liquid
No
Sample 1: liquid
Sample 2: liquid
Yes
Yes
Yes
$T^{low} = T^{high}$
$T^{high} = T^{high} + (T^{high}-T^{low})$
$T^{low} = T^{low} + (T^{high}-T^{low})/2$
$T^{high} = T^{high}$
$T^{low} = T^{low} - (T^{high}-T^{low})/2$
$T^{high} = T^{low}$
No
$T^{high} - T^{low} < \Delta$
Yes
$T_e^m = T^{low}$
(b)
Temperature (K)
0
200
400
600
800
1000
1
2
3
4
5
6
7
8
Convergence of the temperature window
Step 1
T^{high}
T^{low}

Fig. 2. (a) Schematic diagram of the automatized procedure for generating the first estimate of the melting temperature, T_e^m. (b) Convergence of the temperature windows between T^{low} and T^{high} when applying this approach. As an example, fcc Al with the TOR-TILD potential [9] has been used. The threshold temperature window between the lowest and highest temperature limit is $\Delta = 10$ K. The estimated melting point is ~898 K.

as lattice parameters, cohesive energy, elastic moduli, melting temperature and laten heat. A second interatomic potential is a moment tensor potential (MTP) [16] which is a class of machine-learning potentials and recently shown to perform best among different machine-leaning models [17]. A third interatomic potential is a set of EAM potentials only fitted to the free energy of solid phase and has low transferability (referred to as "TOR-TILD potential" as it is specially designed for TOR-TILD method [8,9]). The predicted melting points for these interatomic potentials are provided in Section 3. Note that our fully automatized approach allows to predict the melting point of any given interatomic potential with a very high numerical precision but not necessarily with high accuracy. Accuracy is about the closeness of the predicted melting temperature to the experimental value, which is solely a consequence of the quality of the potential, not of our approach. In the following we will introduce the implementation of the interface method in pyiron.

2. The interface method in pyiron

In order to achieve a high computational efficiency we follow the workflow outlined in the user guide of KISSMD [12] except if otherwise noted. A critical part of our procedure is that all tasks can be automatically performed and linked without the need to manually address the technical details. The general strategy for performing the interface method in pyiron is mapped into 4 steps as shown in Fig. 1. Within each step the challenges with respect to manually performing the calculations and how we address them are given in detail.

Step 1: Prepare a solid sample $N \times N \times N$ as shown in Fig. 1 and roughly estimate a melting point, T_e^m, based on this solid sample. Here N is the supercell size. In the case of a fcc structure, if $N = 10$, the number of atoms in this supercell is 4000.

As mentioned in the introduction, getting a good approximate value of T_e^m is critical to make the approach computationally efficient. If T_e^m is much lower than the real melting point of the potential, full solidification may occur. Opposite, a much higher T_e^m results in full melting. Therefore, a good estimate allows to dramatically reduce the number of loops. A common approach to obtain a first estimate is by heating up the solid and cooling down the liquid, respectively. Due to the absence of nucleation seeds in the solid/liquid, overheating/overcooling occurs. For example, for fcc Al with the TOR-TILD potential [9], by heating up a solid (using the prepared solid sample with 4000 atoms) from 450 to 1100 K and cooling down a liquid (using the last snapshot from the heating procedure) from 1100 to 450 K in temperature steps of 10 K with a simulation time of 20 ps (a timestep of 2 fs is used), the overheating transition point is ~1040 K and the undercooling one ~500 K. The actual melting point lies between these boundaries. A rough initial estimate is the arithmetic average of ~770 K. Even this rough estimate is computationally expensive. Performing the heating/cooling procedure requires 66 MD simulations with the isothermal-isobaric ensemble (*NPT*) and each simulation runs 10 000 MD steps.

To reduce computer time as well as human effort, we implement an automatic approach to determine a rough melting point of the corresponding interatomic potential. The procedure is outlined in Fig. 2 (a). The underlying concept is straightforward and based on the idea of quicksort. Given a start temperature window T^{low} and T^{high}, perform *NPT* MD simulations on the prepared solid sample ($N \times N \times N$) using T^{low} and T^{high}, separately. The relaxed structure from T^{low} is called "Sample 1" and that from T^{high} called "Sample 2". To identify the solid–liquid transition in the samples the common neighbour analysis (CNA) [18] detector in OVITO [19] is used. If both samples remain in the solid phase, the real melting point is higher than T^{high}, the temperature window is updated by T^{low}=T^{high}, and T^{high}=T^{high}+(T^{high}-T^{low}); if "Sample 1" remains in the solid phase and "Sample 2" changes into the liquid phase, then the real melting point is inside the current temperature window and the temperature window will be narrowed down by T^{low}=T^{low}+(T^{high}-T^{low})/2, and T^{high}=T^{high}; if both samples change into the liquid phase, the real melting point is lower than T^{low}, the temperature window is updated by T^{low}=T^{low}-(T^{high}-T^{low})/2, and T^{high} = T^{low}. Using the newly assigned lower and upper temperatures T^{low} and T^{high}, a *NPT* simulation will be performed again on the solid sample. The iteration will stop until the temperature window is smaller than a certain criterion Δ, e.g., 10 K. In our automatized procedure, $T^{low} = 0$ K and $T^{high} = 1000$ K are assigned as the initial temperature window. This choice has been tested and proved to work well in all considered cases. The advantage of the assignments is that it is independent of the specific material and converges rapidly with a step width of T^{high}-T^{low} or $(T^{\mathrm{high}}$-$T^{\mathrm{low}})/2$. The reason to choose 0 K as the initial T^{low} is that the structure at this temperature is in a solid phase, i.e., a significant calculation is not needed. For the initial T^{high}, no benefit has been found by giving a more material specific temperature, such as e.g., the experimental melting temperature. In contrast, when using a material specific value the user needs to pay additional effort to compute it or search for it.

To analyze the efficiency of our automatized approach we use fcc Al with the TOR-TILD potential as an example. The first estimate from our automatized approach is ~898 K. To obtain this rough melting point only 8 *NPT* simulations on the solid sample are necessary (Fig. 2(b)). Comparing to the common approach where 66 *NPT* simulations are needed for the heating/cooling procedure, the computational efficiency is significantly improved. This step can be skipped if the user knows the approximate melting point of the interatomic potential.

Step 2: The procedure of the interface method reported in Ref. [12] suggests to separately prepare the solid and liquid samples and to bring the two samples together to set up the interface structure only in a second step. With this approach the solid and liquid structures are formed separately so that atoms in the interface region are far away from their relaxed configurations. Thus, an extended equilibration is needed. To reduce the equilibration time we construct an initial interface structure where all atoms are close to their equilibrium configuration. For this purpose we double the solid sample from Step 1 in z direction ($N \times N \times 2N$) and perform *NPT* MD simulation on it with $T_{\mathrm{e}}^{\mathrm{m}}$ from Step 1. During the simulation an isotropic relaxation is performed and at the end of the simulation the full structure still remains in the solid phase. Afterwards, we keep one half of the structure as solid by applying selective dynamics. The other half is heated up by doing *NPT* simulation at much higher temperature, e.g., $T_{\mathrm{e}}^{\mathrm{m}} + 1000$ K, in order to trigger the liquid phase. Once the liquid phase is formed, a *NPT* simulation is performed at $T_{\mathrm{e}}^{\mathrm{m}}$ with the atoms in the solid part still fixed. During this simulation an anisotropic relaxation is applied, i.e., only the z direction of the supercell (normal to the interface) is allowed to change its size while the lateral cell-dimensions are kept fixed. At the end of Step 2, an interface structure between the solid and the liquid phase has been established. Fig. 1 shows a snapshot of the interface structure for fcc Al. It should be noted that this interface structure still needs further relaxation to reach full equilibration.

Step 3: Apply different strains (generally 21 strains between −1.0% and +1.0% are applied, however, a large strain range ±5.0% is used for the first few loops, details see Step 3.1) along the direction perpendicular to the interface and keep the lateral supercell dimensions fixed. Perform *NVT* simulations on various strained interface structures with $T_{\mathrm{e}}^{\mathrm{m}}$ in order to relieve the local stress induced during setting up the interface structure and the additional tension introduced by the applied strain. After relieving the stress *NVE* MD is successively performed. The temperature and pressure of each strained interface structure are obtained by time averages over the equilibrated (typically last few thousand) MD steps.

Step 3 is the central part of the interface method. This step involves most of the obstacles which make manually performing these calculations challenging. We have identified and implemented four algorithmic features in this step. They solve the issues and speed up the calculations without sacrificing final numerical precision. They are decomposed into Step 3.1 to Step 3.4.

Step 3.1: Even though Step 1 automatically provides a good estimate of the melting point, it still can be off by a few 100 K from the final (converged) melting temperature. Thus, an iterative approach running over several loops is applied. To balance computational efficiency and precision, different sets of parameters are applied for the MD simulations.

For the first few loops (first stage), a coarse parameter set (2 fs, 25 000, ±5%) is used. The three parameters in the parentheses respectively refer to the timestep, the number of MD steps and the applied strain range on the interface structure. The reason for initially applying a rather large strain range is that we are interested in predicting the melting temperature at $P = 0$ GPa. However, as shown in Fig. 3 using fcc Al with the litEAM potential [14] as an example, the strain at $P = 0$ GPa is −3.0%. For the next loop our algorithm centers the strain interval around this value. In the subsequent loops the center point is updated. In almost all cases it becomes constant after only one or two loops. Note that in the following strain-related figures the strain is shifted so that the strain at $P = 0$ GPa becomes zero. The rescaled pressure-strain dependence of Fig. 3 is shown in Fig. 6(b). In Fig. 3 the pressures along x, y, z directions are plotted as *pxx*, *pyy* and *pzz*, respectively. As the strain is applied along z direction, *pzz* is more sensitive to the strain change than *pxx* and *pyy*. However, all of them turn to zero at the same strain value −3.0%. Therefore, the mean value of *pxx*, *pyy* and *pzz*, i.e., p^{mean}, is used for discussing the temperature–pressure dependence (see Fig. 3). Another point that needs to be addressed is that the interface structure under a large strain normally requires a long simulation time to reach full convergence. For example, the interface structure (8000 atoms) at a strain of 4.5% in Fig. 6(b) needs at least 40 ps to reach convergence. That means, using a timestep of 1 fs a minimum of 40 000 MD steps would be required to reach equilibrium. To improve the computational efficiency without sacrificing numerical precision at this stage, we apply a large timestep of 2 fs and reduce the number of MD steps (e.g., 25 000).

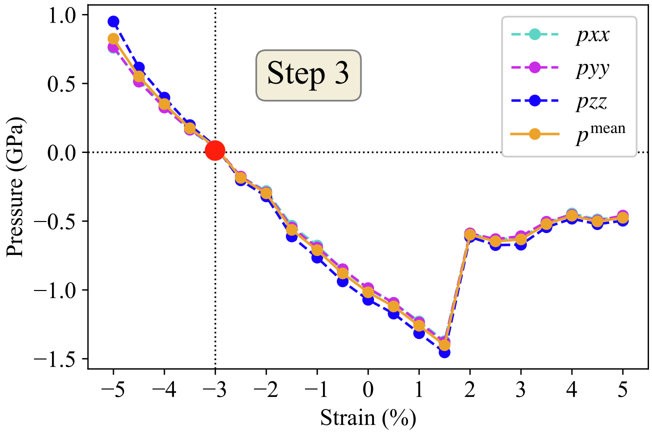

Fig. 3. Pressure-strain dependence obtained in the first loop at temperature $T_{\mathrm{e}}^{\mathrm{m}} = 898$ K. Fcc Al with the litEAM potential [14] has been used. *pxx*, *pyy*, *pzz* are the pressure along x, y and z direction, respectively. p^{mean} is the mean value of *pxx*, *pyy*, *pzz*. The strain of the interface structure at $P = 0$ GPa, marked as the red dot, is −3.0%, i.e., markedly different from zero. To predict the melting point at $P = 0$ GPa with the interface method, the range of strain values should be localized around this point. Therefore, in the next loop calculations the strain range will be updated by applying −3.0% as center point. The abnormal behavior of the last 7 data points is discussed in Step 3.3.

During the second stage of the loop calculations, the parameter set (2 fs, 20 000, ±1%) is used. For a small strain range between −1.0% and +1.0%, the number of MD steps can be reduced (e.g., 20 000) compared to the one needed for a larger strain. Also for equilibrating the structure a timestep of 2 fs is found to be sufficient.

For the final stage, a high convergence parameter set (1 fs, 50 000, ±1%) is used to ensure a high precision of the predicted melting temperature.

The switching between the different stages is controlled by carefully chosen rules. For example, if the last predicted melting point falls in the temperature window determined from the last loop, the parameter set will be updated to the next stage. Otherwise, if the predicted melting point is not in the new temperature window, the calculation parameters remain unchanged.

Step 3.2: As mentioned in the introduction, it is critical to ensure that the temperature and pressure averages attained from the *NVE* simulations are fully converged. Within each loop calculation, 21 *NVE*

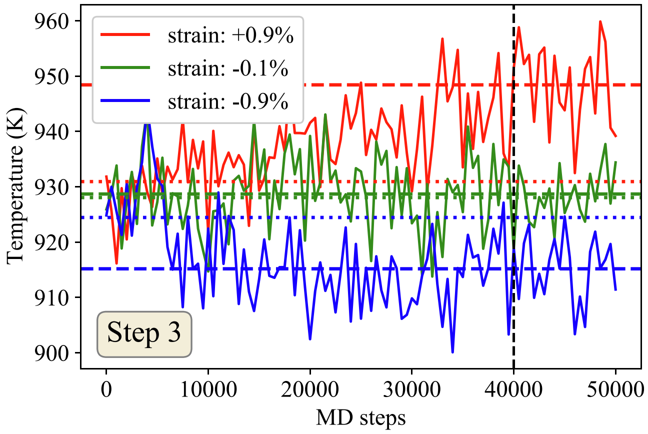

Fig. 4. Monitoring the temperature convergence as function of the MD steps during an *NVE* simulation. The example shows fcc Al with the litEAM potential [14] at strains +0.9%, −0.1% and −0.9% and temperature $T = 926$ K. The results are taken from the final loop. The black vertical dashed line indicates the MD step at which the time average over temperatures starts. The horizontal dashed lines give the mean temperatures over the last 10 000 MD steps. As a comparison, the (non-equilibrated) temperature averages over the first 10 000 MD steps are given (dotted lines).

simulations are generally performed. For example, if 5 loops are necessary to provide a final prediction of the melting point, 5 × 21 calculations need to be analyzed in order to discard unconverged data points. In our automatized procedure the time evolving of temperature/pressure for all loop calculations is automatically monitored. The convergence of each strained interface structure is thus easy to obtain, as shown in Fig. 4.

Step 3.3: To avoid full solidification/melting the algorithm needs to monitor the strained interface structures during *NVE* simulations. If it happens, the corresponding data points are filtered out for the subsequent data analysis. Kernel density estimation (KDE) is a powerful non-parametric technique to estimate densities. Common neighbour analysis (CNA) [18] is an algorithm designed to characterize the local structure environment. Here we combine both techniques. Specially, we use the KDE module from scikit-learn [20] and CNA within OVITO [19]. This combined approach allows us to accurately resolve the solid–liquid boundary at the atomic scale. During the simulation each strained interface structure can be plotted with atoms labeled by different colors (liquid: blue, BCC: orange, FCC: green, and HCP: red), as shown in Fig. 5(a). The corresponding kernel density score along the z direction of the interface structure is provided in Fig. 5(b). The solid–liquid volume ratio in this strained interface structure is given in the legend. We use a volume ratio of 0.25 (25%) as the threshold for eliminating the full liquid/solid, i.e., if the solid/liquid phase is less than 25%, the interface structure will be regarded as full liquid/solid. Such cases are filtered out by the automated data analysis. The solid–liquid volume ratio as function of strain is plotted for all strained interface structures (Fig. 6(a)). One can see in Fig. 6(a) that the first interface structure has a strain ∼−2.0% and contains more than 75% of solid phase. This strain state is thus removed in the data analysis.

Except for the first interface structure, the volume ratios of all other interface structures are within the "safe" region, 0.25–0.75, as shown in Fig. 6(a). That means they are in the solid–liquid coexistence phase. However, we found that in some cases, i.e., under large strains, even though the volume ratio is well located in the "safe" region, the dependence of the solid–liquid volume ratio as function of strain behaves abnormal. An example is shown in Fig. 6(a) for strains larger than +5.0%. In this strain regime also the pressure-strain dependence shows an abnormal nonlinear behavior (Fig. 6(b)). We therefore carefully analyzed the respective interface structures. We find that under larger tensile strain some atoms in the interface structure move apart and form void-like structures. The presence of these voids misguides the detection algorithm resulting in incorrect melting points. To overcome this issue, we apply a Voronoi analysis to calculate the void volume of each atom in the interface structure. The dependence of void volume on strain is plotted in Fig. 6(c): the orange dot is the maximum void volume, $V_{\mathrm{void}}^{\mathrm{max}}$, the black dot is the mean void volume, $V_{\mathrm{void}}^{\mathrm{mean}}$, and the black dashed line is $2 \times \frac{1}{n}\sum V_{\mathrm{void}}^{\mathrm{mean}}$, where n is the number of the strained interface structures, i.e., 21 for the current loop calculations. We find that $V_{\mathrm{void}}^{\mathrm{max}}$ is generally below the black dashed line. However, once the applied strain exceeds the elastic deformation limit, it shifts substantially above the black dashed line.

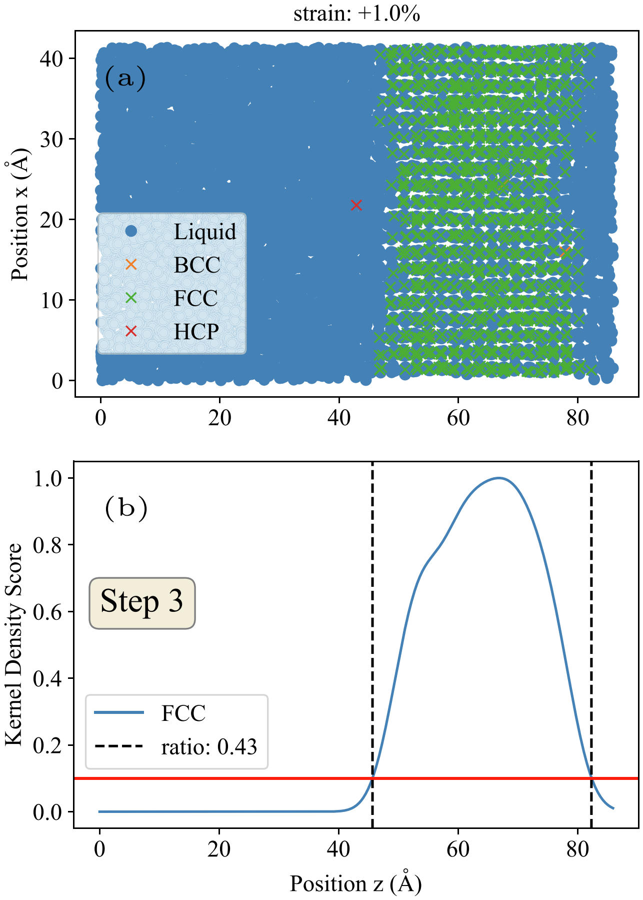

Fig. 5. (a) Two-dimensional snapshot of the (xz) interface structure. The green marks represent the solid fcc phase and the blue ones the liquid phase. (b) Kernel density score of the solid phase (blue line) along the z direction of the interface structure. The red line marks the threshold that defines the solid/liquid interface. The phase below/above the red line is detected as the liquid/solid. The two vertical black dashed lines mark the border between the solid and the liquid phase. Ratio 0.43 means that 43% volume percent remain in the fcc solid phase. The example shown here is for fcc Al with the litEAM potential [14] at strain +1.0% and $T = 926$ K. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

A detailed analysis revealed that this shift is a clear indication of voids forming in the liquid part of the interface structure. By applying this automatized algorithm interface structures containing voids can be easily detected. This is shown in Fig. 6(c) for strains between 5.0% and 8.0%. Fig. 6(d) shows an example of an interface structure with a void. This interface structure corresponds to the blue dots in Fig. 6(a)–(c).

In summary, to capture the melting point with high numerical precision, three mechanisms need to be carefully monitored: (i) ratio between solid and liquid phase, (ii) pressure-strain dependence and (iii) void volume, see Fig. 6(a)–(c). Monitoring these quantities provides a

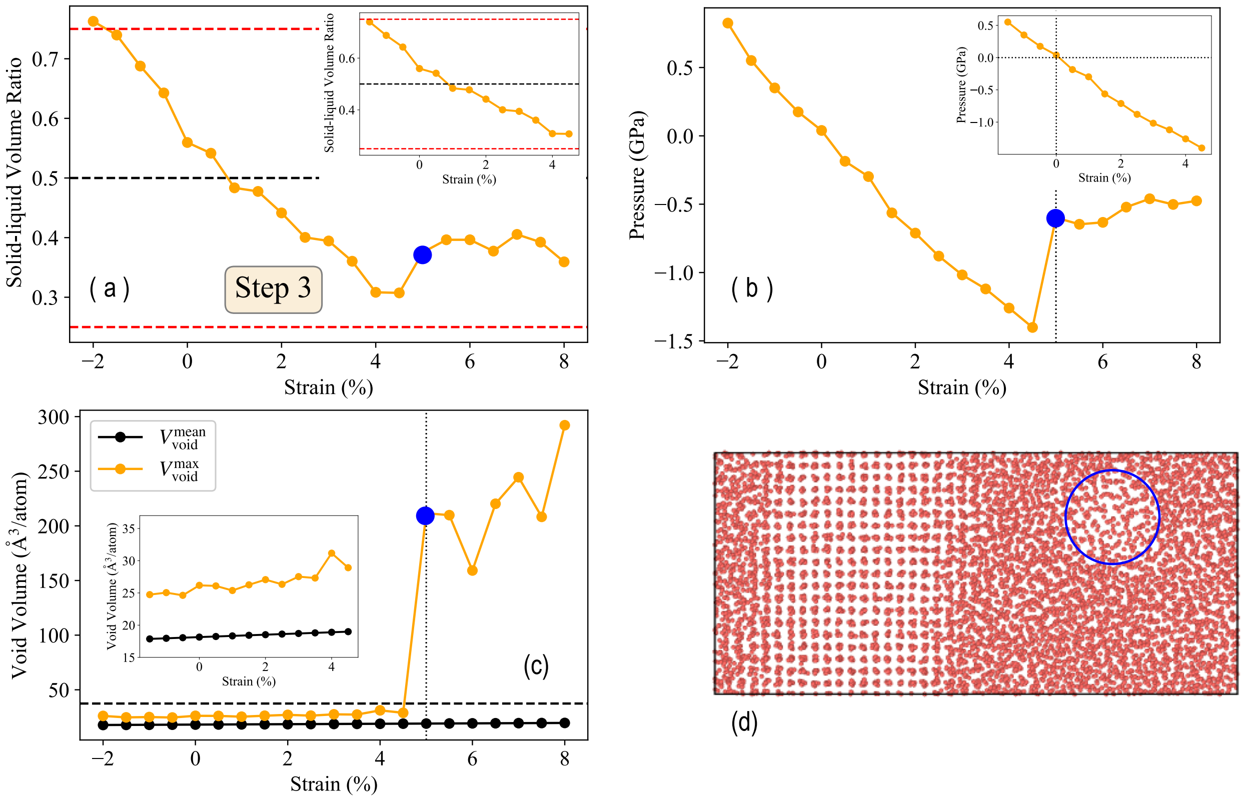

Fig. 6. (a) Solid–liquid volume ratio as function of strain. The black dashed line marks the region where the solid and liquid phases coexist in an equal volume ratio. The red dashed lines mark the threshold for filtering out full solid (0.75) and full liquid (0.25). In the chosen example the first dot has a value larger than 75%. Therefore, this interface structure is detected as a full solid phase and skipped from the data analysis. (b) Pressure-strain dependence obtained from the strained interface calculations. Note that the pressure-strain dependence becomes abnormal for strains larger than +5.0%. (c) Maximum void volume (orange dots), $V_{\mathrm{void}}^{\mathrm{max}}$, as function of strain. The black dots mark the mean value of the void volumes for a given strain value, $V_{\mathrm{void}}^{\mathrm{mean}}$. The black dashed line is the average over all strain values, $2\times\frac{1}{n}\sum V_{\mathrm{void}}^{\mathrm{mean}}$, where n is the number of the strained interface structures. (d) Interface structure with a void. The void is marked by a blue circle. The blue dot shown in (a), (b) and (c) corresponds to this structure. Inset: Plot of the filtered data. The results shown here are from a fcc Al calculation with the litEAM potential [14] at $T = 898$ K (first loop). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

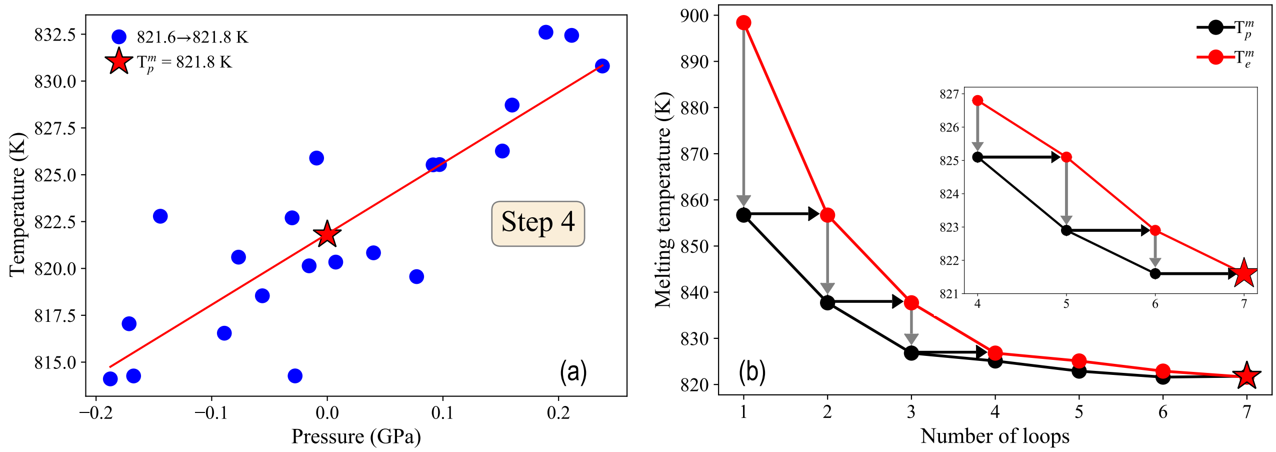

Fig. 7. (a) Temperature–pressure dependence from the last loop. The solid line is linear fit of the data points. In the legend the number left to the arrow is the estimated melting point, $T_{\mathrm{e}}^{\mathrm{m}}$. The one to the right is the predicted melting point, $T_{\mathrm{p}}^{\mathrm{m}}$. (b) Convergence of the predicted melting point as function of the number of loops. The black dot is the predicted melting temperature $T_{\mathrm{p}}^{\mathrm{m}}$. The red one is the estimated melting temperature $T_{\mathrm{e}}^{\mathrm{m}}$. The grey vertical arrow from $T_{\mathrm{e}}^{\mathrm{m}}\rightarrow T_{\mathrm{p}}^{\mathrm{m}}$ marks one loop. The black horizontal arrow indicates that the previous predicted melting point is used as the estimated melting point for the next loop. The inset enlarges the temperature difference between $T_{\mathrm{p}}^{\mathrm{m}}$ and $T_{\mathrm{e}}^{\mathrm{m}}$ from loop 4 to loop 7. In both (a) and (b) the red star is the final predicted melting point from the last loop (blue dots in (a)). The results shown here are for fcc Al with the TOR-TILD potential [9]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

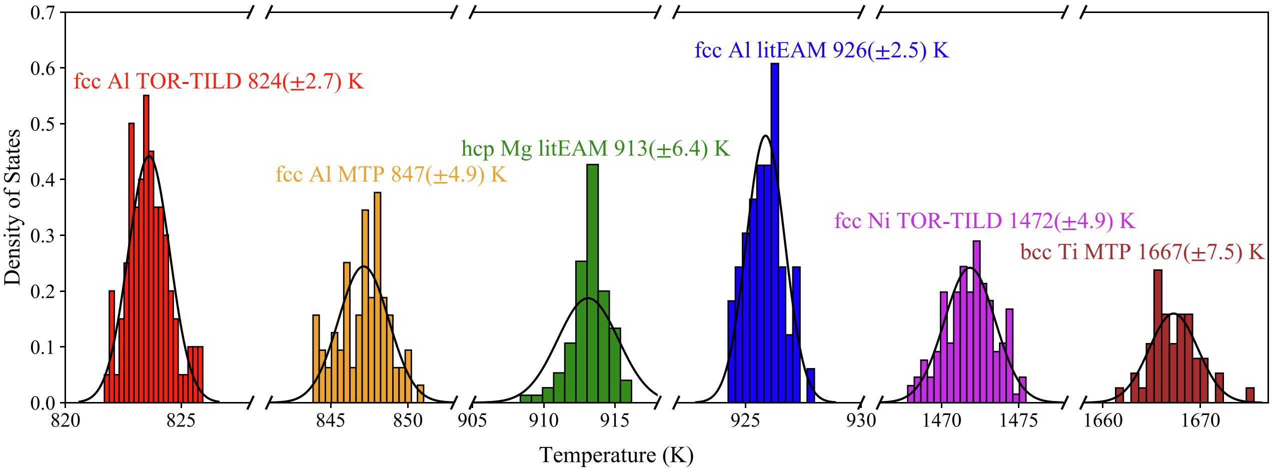

Fig. 8. Distribution of the predicted melting points of fcc Al, Ni, bcc Ti and hcp Mg and for different potentials. The value (in the parentheses) is three times the standard deviation of the distribution providing a 99.7% confidence.

Table 1
Melting points (in K) of fcc Al, Ni, bcc Ti, and hcp Mg for various interatomic potentials. The error bar is given by the standard deviation of the mean (see text).

	FCC	FCC	BCC	HCP
	Al	Ni	Ti	Mg
TOR-TILD	824 ± 0.1	1472 ± 0.1	–	–
litEAM	926 ± 0.1	–	–	913 ± 0.2
MTP	847 ± 0.2	–	1667 ± 0.4	–
Exp	933[a]	1728[b]	1941[c]	923[d]

[a] Ref. [22].
[b] Ref. [23].
[c] Ref. [24].
[d] Ref. [25].

Table 2
Melting points (in K) of fcc Al calculated with potentials from NIST interatomic potentials repository using our automatized interface method. Each melting point is obtained from a single random seed calculation.

Potential	T^m	Potential	T^m
Ref. [26]	207.5	Ref. [36]	928.9
Ref. [27]	577.2	Ref. [37]	932.5
Ref. [28]	846.3	Ref. [38]	942.5
Ref. [29]	870.5	Ref. [39]	950.5
Ref. [30]	871.0	Ref. [40]	1040.8
Ref. [31]	879.7	Ref. [41]	1040.9
Ref. [32]	884.0	Ref. [42]	1041.9
Ref. [33]	923.1	Ref. [43]	1043.3
Ref. [34]	925.9	Ref. [44]	1214.2
Ref. [35]	926.0	Ref. [45]	1352.9
Exp. [22]	933		

firm basis to automatically discard unphysical data points such as full solid/liquid data points or interface structures with voids.

Step 3.4: It may happen that all strained interface structures turn into full liquid/solid, i.e., all data points will be filtered out by Step 3.3. In this case, the estimated melting temperature needs to be decreased/ increased and Step 2 and Step 3 need to be recalculated with the newly estimated melting point. The automatized interface method easily handles this complex situation by automatically detecting structures, making a decision for decreasing or increasing the estimated temperature and performing a new loop calculations with the newly estimated temperature. Decreasing/increasing the temperature by 5% with respect to the previous estimated melting temperature worked well for all considered cases.

Step 4: Predict the melting point T_p^m from the temperature–pressure dependence obtained from Step 3 and compare it with the estimated melting point T_e^m, see Fig. 1. If the difference between T_p^m and T_e^m is larger than the convergence criterion (1 K is used for all the simulations in the present work), repeat the loop from Step 2 to Step 4 with the newly predicted melting point ($T_e^m = T_p^m$) until the difference reaches this criterion.

At the end of Step 4, a final melting point at $P = 0$ GPa is obtained from the temperature–pressure dependence of the last loop, as shown in Fig. 7(a) for fcc Al with the TOR-TILD potential [9]. The estimated melting point T_e^m for this loop is 821.6 K and the predicted melting point T_p^m 821.8 K. The difference is 0.2 K and smaller than the given convergence criterion 1 K. Therefore, the loop is terminated. The obtained final melting point is marked by the red star in Fig. 7 (a). The procedure and the parameters outlined above guarantee a smooth convergence of the predicted melting points, T_p^m and T_e^m. A representative example is shown in Fig. 7(b). To better follow the convergence towards the end Fig. 7(b) shows a zoom-in of temperature difference between T_p^m and T_e^m from loop 4 to 7.

After going from Step 1 to Step 4, a single predicted melting point is obtained from one initial random seed, for example, 821.8 K for fcc Al with the TOR-TILD potential, the red star in Fig. 7. As mentioned in the introduction, using different initial random seeds results in small changes in the solid–liquid volume ratio and then in the predicted melting point. This is a finite time effect. Therefore, the algorithm has to run over a large number of calculations with different initial random seeds to perform a statistical analysis and obtain error bars.

The detailed outline of the various steps clearly shows that manually applying the interface method to predict melting points with high numerical precision is not feasible on a routine or high-throughput basis. Thus conventionally often rather rough convergence criteria (in the order of a few 10 K) are used to reduce the manual work load. With our fully automatized procedure even a criterion of 1 K can be easily attained.

3. Results and discussions

We have implemented, benchmarked and executed the highly automated interface method in pyiron (see details in Appendix A). In this fully automatized approach the crystal structure is automatically linked to the stable phase of the chemical element. Therefore, for stable unary phases only two mandatory parameters enter: the chemical element and its interatomic potential. For metastable or dynamically unstable phases, the crystal structure needs to be provided as an additional parameter. Having provided these parameters all calculations from Step 1 to Step 4 run automatically without any interruption or user interaction. To demonstrate the generality, efficiency and flexibility of the automatized interface method, we performed a representative set of melting point calculations. Examples include fcc Al with the TOR-TILD potential [9], MTP, and the litEAM potential [14], fcc Ni with the TOR-TILD potential [9], and hcp Mg with the litEAM potential [15]. Dynamically unstable bcc Ti with a machine learning potential, i.e., MTP, has been selected as an example because the bcc phase of Ti remains stable from 1155 K up to its melting point. The number of different initial random seeds simulations depends on the specific interatomic potential and varies between 50 to 150. For each single initial random seed simulation, 50 000 MD steps and 1 fs timestep are used. A $10 \times 10 \times 20$ supercell is used for fcc Al and Ni, $14 \times 14 \times 28$ for bcc Ti and $20 \times 20 \times 20$ for hcp Mg. Fig. 8 shows that the distribution of the predicted melting points is Gaussian, i.e., normal distribution. The standard deviation σ of this distribution approaches a finite value and does not go to zero when increasing the number of samples n due to a finite time effect [21]. To estimate the error of the melting temperature we therefore use the standard deviation of the mean $\sigma/\sqrt{n}$. This error estimate is listed in Table 1 for the various potentials. Achieving errors below 1 K requires a statistical analysis over dozens of runs. If one only needs a rough estimate, a single run may be sufficient. To estimate the error with a 99.7% confidence we use the triple standard deviation (see error values given in Fig. 8). The largest error is observed in the case of bcc Ti (MTP potential) with $\sim$7.5 K. For all other cases shown in Fig. 8, the error for a single calculation is substantially smaller. A single run estimate of the melting temperature takes a few hours. To be specific, computing Al with the litEAM potential [14] takes about 4.5 CPU hours on 8 CPU cores. Therefore, a single random seed simulation allows to easily perform a high-throughput analysis of the melting points. As an example we performed such a study for all Al potentials in the NIST repository.

The predicted melting points are summarized in Table 2. As evident from the large scatter of the melting temperatures for the identical element (Al), melting temperatures are not yet routinely used in constructing and fitting empirical potentials. Having now an efficient and easy-to-use approach to routinely obtain precise melting temperatures for empirical potentials will allow to include also this critical quantity in the fitting process. The automated algorithm also links to our previously developed TOR-TILD methodology [8], which allows to efficiently and accurately compute melting points fully from *ab initio*. Within the TOR-TILD method, the interface method is the first critical step and the one not difficult to automize.

4. Conclusions

In summary, we described, analyzed and benchmarked a fully automatized procedure to compute melting points for arbitrary interatomic potentials. pyiron [13], an open source integrated development environment with focus on computational material science, proofed to dramatically ease prototyping, testing and running the high-throughput simulations. To employ the automated interface method on the stable unary phases, users need to provide only two mandatory parameters: chemical species and interatomic potential. For metastable and dynamically unstable phases, the crystal structure of that phase has to be provided as an additional parameter.

As a matter of fact, the prediction of melting temperatures using interatomic potentials strongly depends on the quality of the potential. Having a robust, computationally efficient and easy-to-use approach to determine melting temperatures for any chemical element and potential provides a critically needed tool to benchmark and improve the quality of empirical potentials. The interface method automated in this study is also a key part in advanced methodologies that combines empirical potential simulations and *ab initio* calculations to compute melting temperatures. An example is the recently developed TOR-TILD methodology [8,9]. The fully automatized interface method opens the way towards an automized determination of melting points for *ab initio* calculations.

Our fully automatized approach has been only tested for unary phases. Principally, it can be also applied to compounds. A critical challenge for such an application is that the melting points of the components in the compound can be very different. It can happen that the element having low melting point already melts and those having high melting points still remains solid. Therefore, further development is necessary in order to apply this approach to binary or multi-component compounds. This investigation is still ongoing.

5. Data availability

The fully automated approach has been implemented into a pyiron-based Jupyter notebook. This notebook is available at https://www.mpie.de/4008196/Software.

CRediT authorship contribution statement

Li-Fang Zhu: Methodology, Investigation, Data curation, Visualization, Writing - original draft, Writing - review & editing. **Jan Janssen:** Software, Methodology, Data curation, Investigation. **Shoji Ishibashi:** Investigation. **Fritz Körmann:** Funding acquisition, Validation. **Blazej Grabowski:** Funding acquisition, Supervision. **Jörg Neugebauer:** Conceptualization, Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

We would like to thank Won-Seok Ko for his assistance with the interface method, Osamu Waseda for the kernel-density based solid–liquid detector, and Andrew Ian Duff and Alexander Shapeev for their help in fitting potentials. Funding by the European Research Council (ERC) under the EU’s Horizon 2020 Research and Innovation Programme (Grant No. 639211), by the Max-Planck Research network on “Big-data-driven Material science” (BigMax), NWO/STW (VIDI grant 15707) and by Deutsche Forschungsgemeinschaft (DFG, 405621217) are gratefully acknowledged.

Appendix A. Implementation in pyiron

As mentioned, the interface method from Step 1 to Step 4 has been implemented into a pyiron-based [13] Jupyter notebook [46]. LAMMPS [47] is the MD-engine to perform all automatic calculations. All optional parameters for the automated approach are provided in the first part of the notebook, such as interatomic potential, element, crystal structure (generally commented out), number of atoms, the first temperature window and run time steps for obtaining the first estimate in Step 1, run time steps for *NPT* calculations during setting up the interface structures in Step 2, the boundary to filter out fully solidified/liquid structures, strain range, timestep, and run time steps for *NPT* and *NVE* calculations in Step 3, convergence criterion for Step 4, and the number of CPU cores. As noted before, for the application of our fully automatized interface method to stable unary phases, only two parameters have to be explicitly set by the user: element and potential. For metastable or dynamically unstable phases, its crystal structure needs to be provided. All other parameters are well-tested default values. They provide a compromise between robustness of convergence and computational efficiency.

To perform the automatized interface method calculations, a Jupyter notebook can be downloaded here [https://www.mpie.de/4008196/Software]. The notebook can be executed after modifying the potential and element. Running directly the notebook the details of each step are automatically plotted and can be analyzed retrospectively. To further simplify the application of our tool, we also support execution with Snakemake [https://doi.org/10.1093/bioinformatics/bts480]. This requires an input file, which contains the two user-specified parameters, element and interatomic potential, in the following format:

```
{
  "config": [ "pair_style eam/alloy \n",
              "pair_coeff * * FeC.eam Fe C\n"
             ],
  "filename": ["./examples/bccFe/FeC.eam"],
  "species": ["Fe", "C"],
  "element": "Fe"
  #"crystalstructure": "bcc"
}
```

Here ”config” provides the LAMMPS parameters for loading the potential, ”filename” shows the location of the potential file, ”species” gives the species implemented in the potential, ”element” defines on which material the melting point calculations will be performed. The parameter ”crystalstructure” is only needed for metastable or dynamically unstable phases. The input file is in a JSON (JavaScript Object Notation) format that is easy to read and write for humans and easy to parse and generate for machines. After creating the input.json file, one can simply execute the protocol by using snakemake:

```
snakemake -use-conda -cores 1
```

The parameters defined in the input.json file will overwrite those in the Jupyter notebook. Using this approach, there is no need to interfere with any of the computational or technical details. Both approaches write the estimated melting points and predicted melting points from all loop calculations in a file output.json.

Appendix B. Finite size effects

With the fully automatized interface method the supercell size effect on predicting the melting point can be easily studied. Using fcc Al with the litEAM potential [14] as an example, we did finite size tests in both dimensions, the distance between the interfaces, i.e., z, and the size of the interface, i.e., $x(=y)$, as shown in Fig. B.9(a) and (b). Each point is obtained by averaging five melting points from five different initial random seeds calculations. Fig. B.9(a) shows that for a fixed $x(=y)$, when the distance between the interfaces is large enough, e.g., $z \geqslant 16$, the predicted melting temperatures remain constant with different z. However, for small interface-interface distances $z < 16$, onset of melting/ solidification is strongly enhanced resulting in a strong fluctuation of predicted melting points from different loops, which are thus difficult to converge. The underlying reason is that, as the periodic boundary condition is applied to the interface structure, z must be large enough to minimize the interface-interface interaction. In Fig. B.9(b), we therefore chose $z = 16$ to test the size effect of $x(=y)$ on the melting point. The shift of the predicted melting points from $4 \times 4 \times N$ to $10 \times 10 \times N$ is only ~4 K. We can therefore conclude that, as long as the interface structure is large enough to overcome a critical size for equilibration (mainly in terms of the interface-interface distance), the remaining finite size effects are rather small on the melting point prediction with the here utilized interface method. This is consistent with the findings of Morris *et al.*, who also did not observe a strong finite size effect for the interface method [4]. For our test case of fcc Al with the litEAM potential, we found that an interface structure based on a supercell size of $8 \times 8 \times 16$ (4096 atoms) is already sufficiently large to provide a reliable melting point prediction. For our final calculations $10 \times 10 \times 20$ supercells (8000 atoms) are used for fcc Al.

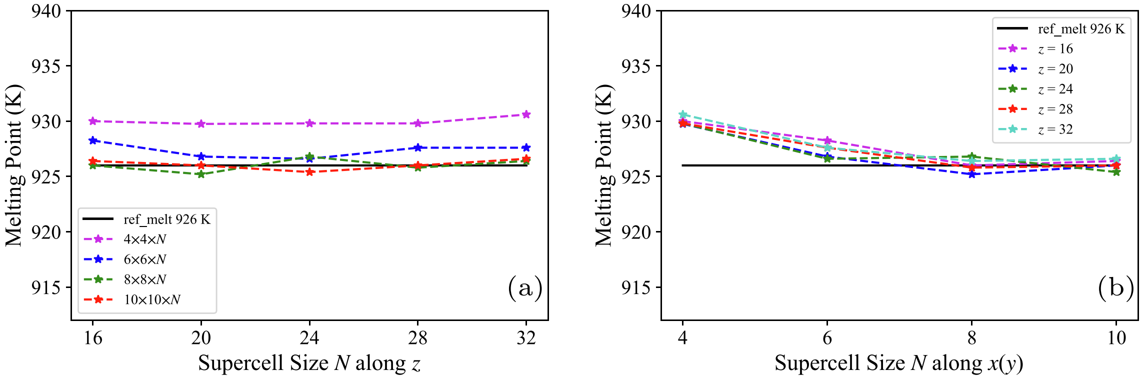

Fig. B.9. Supercell size effects (a) along z direction with different $x(y)$, (b) along $x(y)$ direction with different z, for fcc Al with the litEAM potential [14]. The black solid line in (a) and (b) marks the melting point evaluated for the potential. The dashed lines are from different supercell size simulations.

References

[1] J.H. Perepezko, The Hotter the Engine, the Better, Science 326 (2009) 1068.

[2] K. Lu, The future of metals, Science 328 (2010) 319.

[3] J. Mei, J.W. Davenport, Free-energy calculations and the melting point of Al, Phys. Rev. B 46 (1992) 21.

[4] J.R. Morris, C.Z. Wang, K.M. Ho, C.T. Chan, Melting line of aluminum from simulations of coexisting phases, Phys. Rev. B 49 (1994) 3109.

[5] D. Alfè, M.J. Gillan, G.D. Price, Complementary approaches to the ab initio calculation of melting properties, J. Chem. Phys. 116 (2002) 6170.

[6] D. Alfè, First-principles simulations of direct coexistence of solid and liquid aluminum, Phys. Rev. B 68 (2003), 064423 .

[7] U.R. Pedersen, F. Hummel, G. Kresse, G. Kahl, C. Dellago, Computing Gibbs free energy differences by interface pinning, Phys. Rev. B 88 (2013), 094101 .

[8] L.-F. Zhu, B. Grabowski, J. Neugebauer, Efficient approach to compute melting properties fully from ab initio with application to Cu, Phys. Rev. B 96 (2017), 224202 .

[9] L.-F. Zhu, F. Körmann, A.V. Ruban, J. Neugebauer, B. Grabowski, Performance of the standard exchange-correlation functionals in predicting melting properties fully from first principles: application to Al and magnetic Ni, Phys. Rev. B 101 (2020), 144108 .

[10] G.P. Purja Pun, Y. Mishin, Optimized interatomic potential for silicon and its application to thermal stability of silicene, Phys. Rev. B 95 (2017), 224103 .

[11] K. Ozaki, S. Fukutani, K. Honda, Effect of interatomic potential on melting point and thermal expansion of a transition metal, JSME Int. J. Ser. A 44 (2) (2001) 199.

[12] B.-J. Lee, W.-S. Ko, H.-K. K, E.-H. Kim, The modified embedded-atom method interatomic potentials and recent progress in atomistic simulations, Calphad 34 (4) (2010) 510.

[13] J. Janssen, S. Surendralal, Y. Lysogorskiy, M. Todorova, T. Hickel, R. Drautz, J. Neugebauer, pyiron: An integrated development environment for computational materials science, Comput. Mater. Sci. 163 (2019) 24.

[14] M.I. Mendelev, M.J. Kramer, C.A. Becker, M. Asta, Analysis of semi-empirical interatomic potentials appropriate for simulation of crystalline and liquid Al and Cu, Philos. Mag. 88 (12) (2008) 1723.

[15] D.Y. Sun, M.I. Mendelev, C.A. Becker, K. Kudin, T. Haxhimali, M. Asta, J.J. Hoyt, A. Karma, D.J. Srolovitz, Crystal-melt interfacial free energies in hcp metals: a molecular dynamics study of Mg, Phys. Rev. B 73 (2006), 024116 .

[16] A.V. Shapeev, Moment tensor potentials: a class of systematically improvable interatomic potentials, Multiscale Model. Simul. 14 (2016) 1153.

[17] C. Nyshadham, M. Rupp, B. Bekker, A.V. Shapeev, T. Mueller, C.W. Rosenbrock, G. Csányi, D.W. Wingateand, G.L. Hart, Machine-learned multi-system surrogate models for materials prediction, NPJ Comput. Mater. 5 (2019) 51.

[18] D. Faken, H. Jónsson, Systematic analysis of local atomic structure combined with 3D computer graphics, Comput. Mater. Sci. 2 (1994) 279.

[19] A. Stukowski, Visualization and analysis of atomistic simulation data with OVITO–the Open Visualization Tool, Model. Simul. Mater. Sci. Eng. 18 (2010), 015012 .

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, et al., Scikit-learn: Machine Learning in Python, JMLR 12 (2011) 2825.

[21] X.W. Zhou, S.M. Foiles, Uncertainty quantification and reduction of molecular dynamics models, open access peer-reviewed chapter, 2017.

[22] D.R. Lide (Ed.), CRC Handbook of Chemistry and Physics, 77th ed.,, CRC Press, New York, 1996–1997.

[23] A. Cezairliyan, A.P. Miiller, Melting Temperature of Nickel by a Pulse Heating Technique, Int. J. Thermophys. 5 (1984) 315.

[24] P.D. Desai, Thermodynamic properties of titanium, Int. J. Thermophys. 8 (1987) 781.

[25] A.I. Efimov, L.P. Belorukova, I.V. Vasilkova, V.P. Chechev, Svoistva Neorganicheskih Soedinenii (Himiia, Leningrad, 1983) (inRussian).

[26] J.E. Angelo, N.R. Moody, M.I. Baskes, Trapping of hydrogen to lattice defects in nickel, Model. Simul. Mater. Sci. 3 (3) (1995) 289.

[27] X.W. Zhou, R.A. Johnson, H.N.G. Wadley, Misfit-energy-increasing dislocations in vapor-deposited CoFe/NiFe multilayers, Phys. Rev. B 69 (14) (2004), 144113 .

[28] J.M. Winey, A. Kubota, Y.M. Gupta, A thermodynamic approach to determine accurate potentials for molecular dynamics simulations: thermoelastic response of aluminum, Model. Simul. Mater. Sci. 17 (5) (2009) 55004.

[29] R.R. Zope, Y. Mishin, Interatomic potentials for atomistic simulations of the Ti-Al system, Phys. Rev. B 68 (2) (2003), 024102 .

[30] Y. Mishin, Atomistic modeling of the γ and $\gamma^{'}$-phases of the Ni-Al system, Acta Mater. 52 (6) (2004) 1451.

[31] X.-Y. Liu, J.B. Adams, Grain-boundary segregation in Al-10% Mg alloys at hot working temperatures, Acta Mater. 46 (10) (1998) 3467.

[32] X.-Y. Liu, F. Ercolessi, J.B. Adams, Aluminium interatomic potential from density functional theory calculations with improved stacking fault energy, Model. Simul. Mater. Sci. 12 (4) (2004) 665.

[33] A. Landa, P. Wynblatt, D.J. Siegel, J.B. Adams, O.N. Mryasov, X.-Y. Liu, Development of glue-type potentials for the Al-Pb system: phase diagram calculation, Acta Mater. 48 (8) (2000) 1753.

[34] X.-Y. Liu, C.-L. Liu, L.J. Borucki, A new investigation of copper's role in enhancing Al-Cu interconnect electromigration resistance from an atomistic view, Acta Mater. 47 (11) (1999) 3227.

[35] M.I. Mendelev, M. Asta, M.J. Rahman, J.J. Hoyt, Development of interatomic potentials appropriate for simulation of solid-liquid interface properties in Al-Mg alloys, Philos. Mag. 89 (34–36) (2009) 3269.

[36] X.-Y. Liu, P.P. Ohotnicky, J.B. Adams, C. Lane Rohrer, R.W. Hyland, Anisotropic surface segregation in Al-Mg alloys, Surf. Sci. 373 (2–3) (1997) 357.

[37] J.B. Sturgeon, B.B. Laird, Adjusting the melting point of a model system via Gibbs-Duhem integration: application to a model of aluminum, Phys. Rev. B 62 (22) (2000) 14720.

[38] M.I. Mendelev, D.J. Srolovitz, G.J. Ackland, S. Han, Effect of Fe segregation on the migration of a non-symmetric Σ5 Tilt Grain Boundary in Al, J. Mater. Res. 20 (1) (2005) 208.

[39] V.V. Zhakhovskii, N.A. Inogamov, Y.V. Petrov, S.I. Ashitkov, K. Nishihara, Molecular dynamics simulation of femtosecond ablation and spallation with different interatomic potentials, Appl. Surf. Sci. 255 (24) (2009) 9592.

[40] G.P. Purja Pun, V. Yamakov, Y. Mishin, Interatomic potential for the ternary Ni-Al-Co system and application to atomistic modeling of the B2–L10 martensitic transformation, Model. Simul. Mater. Sci. 23 (6) (2015), 065006 .

[41] G.P. Purja Pun, Y. Mishin, Development of an interatomic potential for the Ni-Al system, Philos. Mag. 89 (34–36) (2009) 3245.

[42] F. Apostol, Y. Mishin, Interatomic potential for the Al-Cu system, Phys. Rev. B 83 (5) (2011), 054116 .

[43] Y. Mishin, D. Farkas, M.J. Mehl, D.A. Papaconstantopoulos, Interatomic potentials for monoatomic metals from experimental data and ab initio calculations, Phys. Rev. B 59 (5) (1999) 3393.

[44] Y. Mishin, M.J. Mehl, D.A. Papaconstantopoulos, Embedded-atom potential for B2-NiAl, Phys. Rev. B 65 (22) (2002), 224114 .

[45] D. Farkas, C. Jones, Interatomic potentials for ternary Nb-Ti-Al alloys, Model. Simul. Mater. Sci. 4 (1) (1996) 23.

[46] F. Perez, B.E. Granger, Ipython: A system for interactive scientific computing, Comput. Sci. Eng. 9 (3) (2007) 21.

[47] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, J. Comput. Phys. 117 (1995) 1.