The Coupled Atomistic/Discrete-Dislocation Method in 3d
Part I: Concept and Algorithms

G. Anciaux[a,1,*], T. Junge[a], M. Hodapp[b], J. Cho[a], J.-F. Molinari[a], W. A. Curtin[b]

[a] *Ecole Polytechnique Fédérale de Lausanne (EPFL)*
Faculté ENAC-IIC, LSMS (http://lsms.epfl.ch/)
CH-1015 Lausanne, Switzerland

[b] *Ecole Polytechnique Fédérale de Lausanne (EPFL)*
Faculté STI, LAMMM (http://lammm.epfl.ch/)
CH-1015 Lausanne, Switzerland

Abstract

The Coupled Atomistic/Discrete-Dislocation (CADD) method is a concurrent multiscale technique that couples atomistic and discrete dislocation domains with the ability to pass dislocations seamlessly between domains. CADD has been demonstrated only in 2d plane-strain problems, for which each individual dislocation is either entirely atomistic or entirely discrete. Here, a full 3d implementation of CADD is presented, with emphasis on the algorithms for handling the description of dislocation lines that span both atomistic and continuum domains, so-called *hybrid* dislocations. The key new features of the method for 3d are (i) the use of an atomistic template of the dislocation core structure to transmit the proper atomistic environment of a continuum dislocation to the atomistic domain for hybrid dislocations and (ii) a staggered solution procedure enabling evolution of the hybrid dislocations. The method naturally requires calibration of discrete-dislocation Peierls stresses and mobilities to their atomistic values, implementation of a dislocation detection algorithm to identify atomistic dislocations, and computation of continuum dislocation displacement fields that provide boundary conditions for the atomistic problem. The method is implemented using the atomistic code LAMMPS and the discrete dislocation code ParaDiS within the LibMultiscale environment developed by the lead authors, and so has all the advantages of these widely-used high-performance open-source codes. Validation and application of CADD-3d are presented in companion papers.

1. Introduction

The mechanical behavior of metals underpins the structural performance of components and systems across the entire spectrum of modern technologies. As demands on material performance per unit cost or weight increase, there is an increasing need to understand the origins of mechanical behavior as well as a need to design new materials with improved properties (e.g. yield strength, ductility, fatigue or corrosion resistance). This requires, in turn, the study of metals at increasingly smaller sizes – at the level of the dislocation defects that are the carriers of plastic deformation and at the atomic level where the intimate interactions of dislocations, cracks, solutes, and other defects control the observed macroscopic behavior. Furthermore, it is now widely recognized that “plasticity” depends on the size of the region being deformed, generally obeying the adage “smaller is stronger”. Such size effects are due to the fundamental structuring of the dislocation defects over mesoscopic scales on the order of microns. Thus, for accuracy and realism, plasticity must be modeled simultaneously across several scales, which presents conceptual and operational challenges.

Capturing material behavior at several scales requires the development and use of multiscale models. So-called “hierarchical” methods use existing models at individual scales and pass appropriate parameters from one scale to another. Due to their simplicity - there is no direct coupling of scales - hierarchical

[*] Corresponding author
[1] *E-mail address:* guillaume.anciaux@epfl.ch

methods are the most commonly used approaches and should be used whenever possible. Hierarchical methods rely, however, on the strong assumption that there exists a clear separation of scales of phenomena. Unfortunately, such a separation often does not apply in plasticity because phenomena at the atomistic scale, dislocation (mesoscale), and continuum scales are strongly coupled in determining observed macroscopic behavior such as crack growth, hardening, or embrittlement. The issue of scale separation has been tackled by the development of so-called "concurrent" methods that seamlessly integrate material descriptions at several scales within a single computational framework. Such methods are far more difficult to develop due to the need for handling a continuous dialog between domains at different scales. It is implicit in all concurrent models that the higher-scale be sufficiently accurate for problems of inerest, and this requires suitable calibration of the higher scale problem; the quality of the calibration is important for overall accuracy but is not intrinsic to the coupling method itself. Two existing methods capable of handling non-trivial plasticity phenomena operating at multiple scales simultaneously are (i) the "Coupled Atomistic/Discrete-Dislocation" (CADD) model [33] where a fully atomistic domain communicates and exchanges dislocations with a mesoscale dislocation dynamics regime, (ii) the "Coupled Discrete-Dislocation/Crystal-Plasticity" method [37] where a discrete dislocation domain is directly coupled to a domain described by continuum plasticity. These methods operate only within the limits of a 2d plane-strain approximation. While these multiscale methods enable the investigation of many physical plasticity problems, the vast majority of problems require a full three-dimensional description of the dislocation plasticity. The CAC model [39, 40] is a more recent model in which a coarse-graining procedure starting from the atoms leads to a continuum domain that can support 3d dislocation defects activated on slip planes depending on the chosen mesh. The CAC method has the nice feature of preserving the correct anisotropic elasticity, and it automatically embues the coarse-grained dislocations with various properties such as a Peierls stress and mobility. On the other hand, the properties of the coarse-grain dislocations (mobility, Peierls stress, line tension) generally differ from those of the fully-atomistic dislocations in a scale-dependent manner, and so a method in which higher-scale dislocation models carefully match the atomistic dislocations has clear benefits. In addition, coarse-graining involves an artificial smearing of the dislocation Burgers vector. This influences interactions among dislocation segments at distances comparable to the coarse-graining scale, leading to possible loss of fidelity for some critical dislocation processes such as junction formation. Thus, higher scale methods that preserve dislocation interactions at small scales again have clear benefits.

In the present work, we introduce a new concurrent multiscale method, three-dimensional Coupled Atomistic/Discrete-Dislocations (CADD-3d), in which there is an intimate and seamless coupling between molecular dynamics (MD) and discrete dislocation dynamics (DDD). Molecular dynamics is to be used in local domains where dislocations nucleate or strongly interact with other atomistic defects such as cracks or grain boundaries, *i.e.* where an atomistic resolution is essential. Outside of such regions, the evolving plasticity (network of dislocations) is described by DDD at a much lower computational cost with no loss in fidelity of the plasticity evolution beyond the usual assumptions of the well-established DDD method. The main challenge for CADD-3d is the simultaneous description of different parts of the *same* dislocation line spanning the two different domains (MD and DDD) at the same instant of time; we refer to such dislocations as hybrid dislocations. We present a coupling that is robust in providing a correct atomistic description when the dislocations intersect the coupling interface, such that the atomistic domain responds as if embedded in an immense atomistic domain, *i.e.* atoms do not "know" that other parts of the material are actually treated by an entirely different method because the forces on the atoms are (almost) exactly the forces that would be generated in the immense (computationally unfeasible) atomistic simulation.

The remainder of this paper is organized as follows. In the next section, we outline the general domain decomposition problem in the presence of hybrid dislocations and the staggered algorithm for evolving both domains as the hybrid dislocations move. In Section 3, we discuss the implementation of proper boundary conditions on the atomistic domain for a hybrid dislocation. We assume that readers are familiar with the underlying methods (DDD and MD), but for completeness Appendice A provides brief reviews of the fundamental methods and standard implementations. We summarize the method in Section 4, with quasistatic validation of key aspects presented in the companion paper CADD-3d.II and dynamic validation and example problems presented in companion paper CADD-3d.III.

2. The CADD-3d algorithm

CADD-3d couples MD, DDD, and linear elasticity concurrently and assumes a decomposition of the computational domain Ω into a convex atomistic domain Ω_{A} and a non-convex continuum domain Ω_{C} separated by a sharp interface $\partial\Omega_{\mathrm{I}}$, see Figure 1.

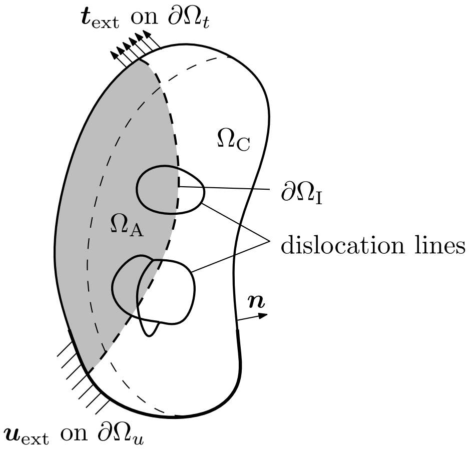

$\boldsymbol{t}_{\mathrm{ext}}$ on $\partial\Omega_t$
Ω_{C}
$\partial\Omega_{\mathrm{I}}$
Ω_{A}
dislocation lines
$\boldsymbol{n}$
$\boldsymbol{u}_{\mathrm{ext}}$ on $\partial\Omega_u$

Figure 1: The computational domain Ω is decomposed into a convex atomistic domain Ω_{A} and a continuum domain Ω_{C}, separated by a sharp interface $\partial\Omega_{\mathrm{I}}$. Tractions $\boldsymbol{t}_{\mathrm{ext}}$ or displacements $\boldsymbol{u}_{\mathrm{ext}}$ are applied on the external traction and displacement boundaries $\partial\Omega_{\mathrm{t}}$ and $\partial\Omega_{\mathrm{u}}$. Dislocation lines or networks are free to span and propagate across the domain.

As in any *force-based* concurrent atomistic/continuum coupling method, the displacements of the atoms coinciding with the interface $\partial\Omega_{\mathrm{I}}$ serve as displacement boundary conditions for the continuum domain. The solution of the continuum domain, given those boundary conditions and external boundary conditions $\boldsymbol{t}_{\mathrm{ext}}$ and $\boldsymbol{u}_{\mathrm{ext}}$, then provides a displacement field that is used to determine the positions of atoms in a so-called pad region Ω_{P} surrounding the atomistic domain but lying physically inside the continuum domain (omitted in Figure 1 for clarity, but shown at a later stage in Figure 2). The positions of the pad atoms then serve as displacement boundary condition for the MD problem. Although fixed within the MD problem, the pad atoms exert the correct non-local atomic forces on the interior atoms that should arise from the atoms represented by the continuum domain. The interior atoms then move according to the total atomic forces acting on them. The pad thickness is thus slightly larger than the cut-off radius of the interatomic potential, so that all atoms in Ω_{A} "feel" a full complement of surrounding atomic neighbors.

In the original plane-strain CADD, each dislocation is uniquely contained in one domain or the other, and so the domain decomposition involves two well-defined boundary value problems, one for MD and one for DDD, at any instant. In 3d, problems containing only fully atomistic and/or fully continuum dislocations, i.e. no hybrid dislocations, can be treated in exactly the same manner. In plane-strain CADD, special algorithms are used to transfer dislocations back and forth between the two domains. In 3d, portions of dislocation loops can move between the two domains, with other portions of the same loop residing inside the two domains (i.e. hybrid dislocations are created), as shown in Figures 2a, 2b, 2c. The previous passing algorithms are incapable of handling this situation, and the formulation of the domain-decoupling in the presence of hybrid dislocations has not been established; doing so is the main goal of this paper.

Figure 1 shows a three-dimensional view of a generic coupled domain. For visual clarity, we will explain the CADD algorithm using a schematic problem in which all dislocations are confined within one single slip plane. Then, a two-dimensional representation of the problem in a cross section of Ω can be shown without loss of detail or generality. Figure 2a shows a single dislocation loop partially located within the continuum domain Ω_{C} and partially within Ω_{A}. The entire domain $\Omega = \Omega_{\mathrm{A}} \cup \Omega_{\mathrm{C}}$ is bounded by $\partial\Omega = \partial\Omega_{\mathrm{u}} \cup \partial\Omega_{\mathrm{t}}$, on which displacement and traction are imposed, respectively. Internal boundaries connect the atomistic and continuum domain: the internal boundary of Ω_{C} is the sharp interface $\partial\Omega_{\mathrm{I}}$ while the internal boundary for the atomistic domain is defined by the finite-thickness pad region Ω_{P}.

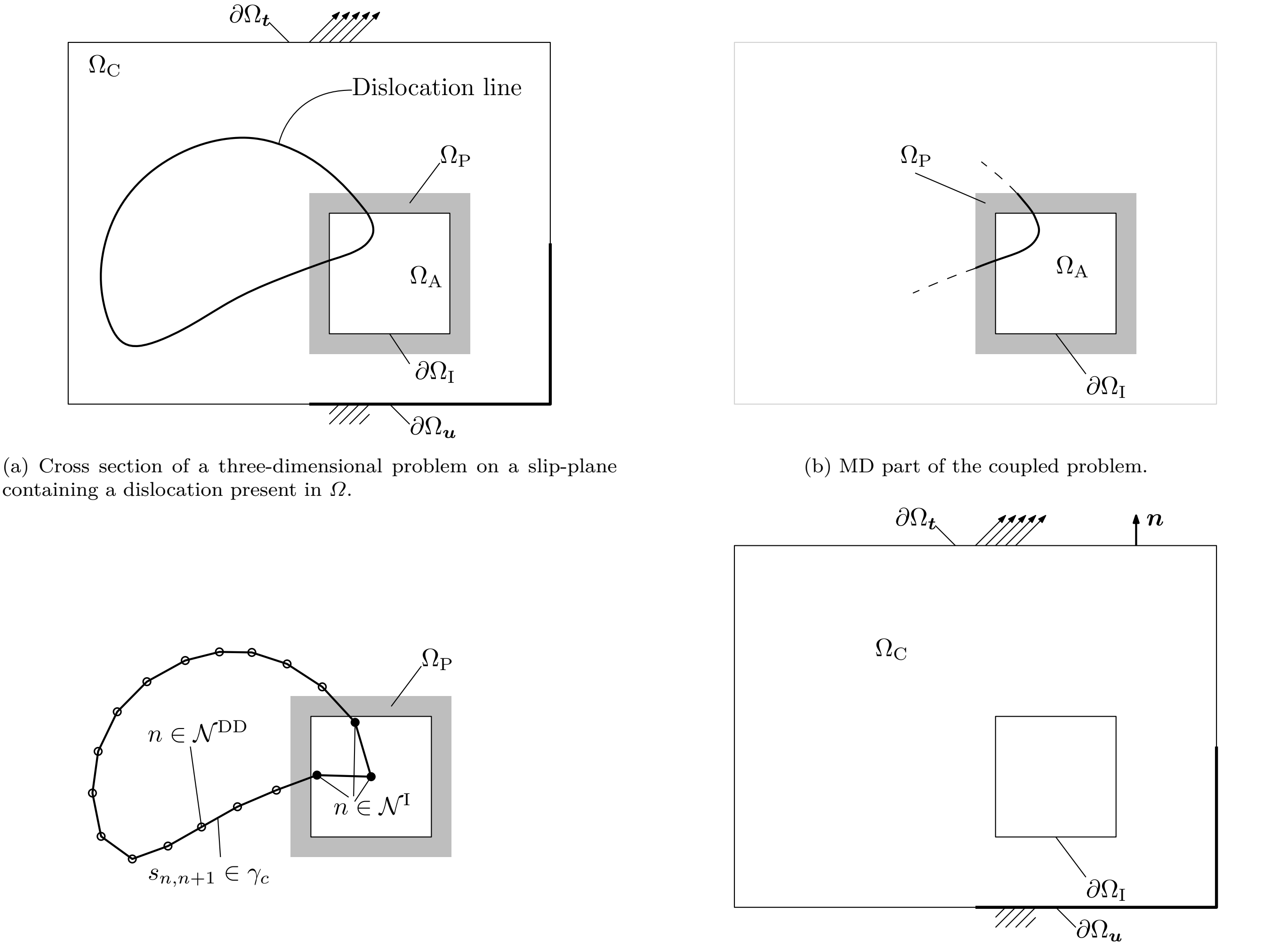
CADD-3d solves the DDD mechanics problem involving hybrid dislocations at any instant of time using a well-established approach [38, 7, 9]. For the DDD problem, the boundary conditions at the atomistic/continuum interface are displacement boundary conditions defined by the positions of the interface atoms. These displacements transmit all information about deformation and defects inside the atomistic domain to the continuum domain. With this boundary condition specified, the displacements and stresses in the DDD domain are then solved using the standard superposition method. The continuum portions of any hybrid dislocations terminate on the boundary $\partial\Omega_{\mathrm{I}}$. To form the closed loops required for the computation of $\tilde{\bullet}$ fields, additional fictitious segments are added outside of the DDD domain as shown in Figure 2c. The added continuum segments outside the DDD domain have no influence on the final solution, even if they overlap with the atomistic domain, because their fields are canceled exactly by superposition as a result of the DDD procedure (see Weygand et al. [38]). Therefore, we are free to close all loops at the center of the atomistic domain, which is convenient for management of all fictitious segments. The superposition method divides the DDD problem into two sub-problems, see Figures 2c and 2d. The first is an infinite-domain problem consisting of all dislocation loops in an infinite homogeneous elastic space (Figure 2c), with the stress and displacement fields solving this problem denoted as $\tilde{\boldsymbol{\sigma}}(\boldsymbol{x})$ and $\tilde{\boldsymbol{u}}(\boldsymbol{x})$. The second is a boundary value problem on the DDD domain now containing no dislocations and subject to the boundary conditions (i) $\hat{\boldsymbol{t}} = [\boldsymbol{t}(\boldsymbol{x}) - \tilde{\boldsymbol{\sigma}}(\boldsymbol{x})\,\boldsymbol{n}(\boldsymbol{x})]$ on the boundary $\partial\Omega_{\mathrm{t}}$ where traction boundary conditions $\boldsymbol{t}$ are applied and (ii) $\hat{\boldsymbol{u}} = (\boldsymbol{u}(\boldsymbol{x}) - \tilde{\boldsymbol{u}}(\boldsymbol{x}))$ on the boundary $\partial\Omega_{\mathrm{u}} \cup \partial\Omega_{\mathrm{I}}$ where displacement boundary conditions $\boldsymbol{u}$ are applied. This second problem is solved numerically and yields the smooth fields $\hat{\boldsymbol{u}}(\boldsymbol{x})$ and $\hat{\boldsymbol{\sigma}}(\boldsymbol{x})$ throughout the DDD domain (since $\hat{\boldsymbol{\sigma}}$ and $\hat{\boldsymbol{u}}$ have no singularity in $\partial\Omega_{\mathrm{u}} \cup \partial\Omega_{\mathrm{I}}$). The full solution throughout the DDD domain is then the superposition of the fields of the two sub-problems, $\boldsymbol{u}(\boldsymbol{x}) = \tilde{\boldsymbol{u}}(\boldsymbol{x}) + \hat{\boldsymbol{u}}(\boldsymbol{x})$ and $\boldsymbol{\sigma}(\boldsymbol{x}) = \tilde{\boldsymbol{\sigma}}(\boldsymbol{x}) + \hat{\boldsymbol{\sigma}}(\boldsymbol{x})$. We specifically reiterate that the closure of any hybrid dislocation loops outside the DDD domain is arbitrary as far as the continuum displacement fields are concerned.

CADD-3d solves the MD mechanics problem involving hybrid dislocations at any instant of time as follows. As in other atomistic/continuum coupling methods, and as stated above, the displacement fields due to applied boundary conditions and discrete dislocations in the DDD region are imposed on the so-called "pad atoms" in domain Ω_{P} that surrounds the true MD domain Ω_{A}. That is, the pad atom positions are controlled entirely by the DDD problem and provide displacement boundary conditions for the inner MD domain Ω_{A}. Use of the continuum fields is valid for all dislocation segments lying in the DDD domain away from the pad region, but is woefully inadequate for the hybrid dislocation segments actually penetrating the pad region and entering into the MD region. We discuss this crucial issue in considerable detail in section 3 but here simply state that the continuum Volterra field of the DDD problem is enriched around the center line of the hybrid dislocation by an atomistic corrective field, the *core template*, such that the pad displacements provide an accurate atomistic representation of the true dislocation core structure.

The key aspects of the CADD-3d algorithm for dealing with hybrid dislocations are then best illustrated by considering the evolution of a straight hybrid dislocation (see Figure 3a). In the atomistic domain, the dislocation exists implicitly via local atomic positions. In the continuum domain, the dislocation exists explicitly as a sequence of discrete dislocation nodes and segments, and the dislocation properties are calibrated to match the atomistic system. Under a uniform driving stress (Peach-Koehler force on the dislocation), the straight dislocation should remain straight and there is no need to solve the elasticity problem. The evolution of the straight dislocation is achieved as illustrated schematically in Figure 3b and the algorithm executed at each time integration step is as follows.

Starting at $t = t_n^+$,[2] the dislocation line is straight. The nonlinear core region of the MD dislocation (pink shaded area) is matched by the pad displacements using the *core template* (blue shaded area). The free DDD nodes – represented by blue squares – are subject to the Peach-Koehler forces (see A.3.2) illustrated by the blue arrows. An additional continuum dislocation segment extends into the atomistic domain, having been detected by established methods for finding atomistic dislocation lines. The terminating end node of the DDD dislocation is shown by the red circle inside the MD domain and is considered immobile in the DDD computation. In the first step, the dislocation advances by an increment Δl along most of its length, in both DDD and MD domains, except in the region close to the interface where boundary conditions are

[2] The exponent $()^+$ indicates the time instantly after the force calculation phase, an exponent $()^-$ instantly before coupling the positional degrees of freedom.



$\partial\Omega_{\boldsymbol{t}}$
Ω_{C}
Dislocation line
Ω_{P}
Ω_{A}
$\partial\Omega_{\mathrm{I}}$
$\partial\Omega_{\boldsymbol{u}}$
Ω_{P}
Ω_{A}
$\partial\Omega_{\mathrm{I}}$
Ω_{P}
$n \in \mathcal{N}^{\mathrm{DD}}$
$n \in \mathcal{N}^{\mathrm{I}}$
$s_{n,n+1} \in \gamma_c$
$\partial\Omega_{\boldsymbol{t}}$
$\boldsymbol{n}$
Ω_{C}
$\partial\Omega_{\mathrm{I}}$
$\partial\Omega_{\boldsymbol{u}}$

(a) Cross section of a three-dimensional problem on a slip-plane containing a dislocation present in Ω.

(b) MD part of the coupled problem.

(c) DDD sub-problem solving the stresses $\tilde{\boldsymbol{\sigma}}$ and displacements $\tilde{\boldsymbol{u}}$ for dislocations in infinite homogeneous elastic medium.

(d) Boundary value sub-problem of DDD problem solving for the corrective stresses $\hat{\boldsymbol{\sigma}}$ and displacements $\hat{\boldsymbol{u}}$ of the DDD-domain subject to corrected applied external tractions $\hat{\boldsymbol{t}} = \boldsymbol{t} - \tilde{\boldsymbol{\sigma}}\,\boldsymbol{n}$ on $\partial\Omega_{\mathrm{t}}$ and corrected applied displacements $\hat{\boldsymbol{u}} = \boldsymbol{u} - \tilde{\boldsymbol{u}}$ on $\partial\Omega_{\mathrm{u}}$ and $\partial\Omega_{\mathrm{I}}$.

Figure 2: Decomposition of the computational problem into their component sub-problems in the CADD method.

enforced, *i.e.* the pad atoms and the detected and constrained DDD node retain the positions of $t = t_n^+$ at $t = t_{n+1}^-$. Then, the new MD dislocation location is detected and the constrained end node is advanced to the new position along the true atomistic dislocation line. The DDD segment attached to the constrained node is then recomputed, re-straightening the discrete dislocation line at $t = t_{n+1}^+$. The atomistic template in the pad is then updated according to the new continuum dislocation line. The MD dislocation line is still non-straight at the end of this sequence (framed area in the figure) but the core template provides a strong driving force to advance this small portion of the atomistic dislocation in the next time step. In practice, the dislocation advances by only a small fraction of a Burgers vector per time step $\Delta l \ll ||B||$ and the non-linear core template is typically several Burgers vectors in diameter, so the non-straight portion of dislocation is actually quite small (the figure is exaggerated) and negligible. With this schematic and short description, we can now turn to the precise algorithm in the next section.

2.1. Integration in time

In computational mechanics, one typically chooses to evolve dynamic systems in time using some predictor-corrector-type integration scheme (*e.g.*, velocity-Verlet for MD, explicit or implicit Newmark β for finite-element method (FEM)) because of their excellent energy conservation properties. Such schemes have the following steps in common: i) at the beginning of a time step, the new positions (along with predicted velocities) for the end of the time step are computed based on the forces in the current configuration, ii) the forces are evaluated for the new configuration, iii) the predicted velocities at the end of the time step

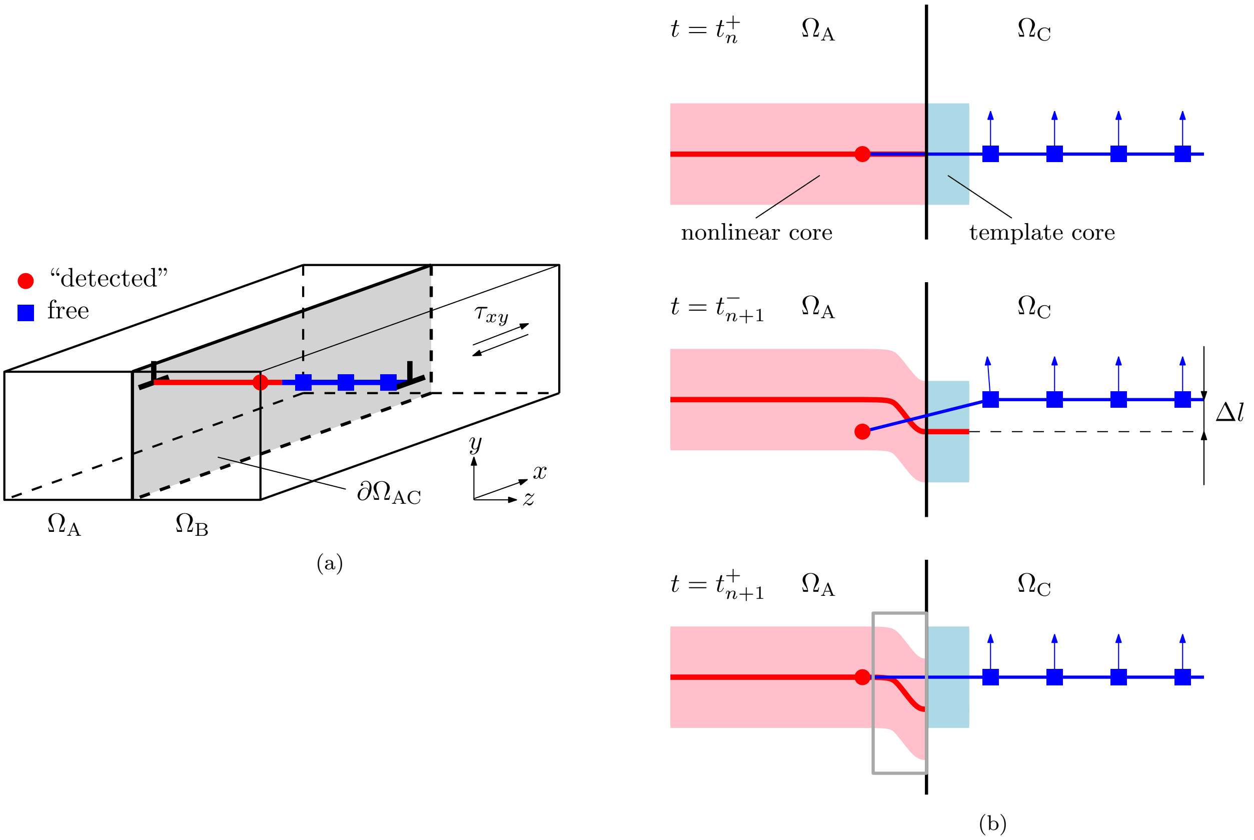


Figure 3: (a) Simulation setup for straight hybrid dislocation. The blue squares represent regular free DDD nodes. The constrained "detected" node, represented by the red disk, is moved by the coupling scheme, not the DDD method. The interface (marked grey) is intersected at a right angle. (b) Hybrid dislocation pinning at the interface.

are corrected. The coupling of systems integrated with predictor-corrector schemes is straight-forward and consists of synchronizing those displacement degrees of freedom that serve as internal mutual boundary conditions between sub-problems after the predictor step and the force degrees of freedom that serve as boundary conditions after the force evaluation step.

The case of CADD-3d adds two complications. First, force-based coupling of a dynamic atomistic system to a dynamic elastic system is inherently unstable [24], and second, DDD codes do not have a predictable nor constant time step. For instance, the integration time step for the DDD sub-problem may be cut short if an event such as a collision between DDD nodes is detected [38, 7, 9]. As a result, we only consider semi-dynamic cases (molecular dynamics in the atomistic domain and statics in the continuum). We thus present the CADD-3d algorithm for consistently coupling a MD sub-problem integrated in time using the velocity-Verlet scheme to a static elastic (*e.g.* FEM) sub-problem and a DDD sub-problem that performs an unknown number of sub-time steps per global system time step. Algorithm 1 shows a detailed overview of the operations and their order of application, listed by sub-problem.

Phase $\textcircled{1}$ updates the free positional degrees of freedom for the two dynamic domains; in the MD sub-problem, phase $\textcircled{1}$ uses the atomic forces $\boldsymbol{f}_\mathrm{a}^k$, positions $\boldsymbol{r}_\mathrm{a}^k$ and velocities $\boldsymbol{v}_\mathrm{a}^k$ on all free atoms (*i.e.*, atoms in Ω_A, but not for the pad atoms in Ω_P) from the previous time step ($t = t_k$) to execute the predictor part of the velocity-Verlet scheme, which includes updating the atomic positions to $\boldsymbol{r}_\mathrm{a}^{k+1}$ (at $t = t_{k+1}$) and predicted velocities to $\boldsymbol{v}_\mathrm{a}^{k+1/2}$ in the MD region Ω_A and computing the displacement $\boldsymbol{u}_n^{k+1}$ of those atoms coinciding with FEM nodes on the interface $\partial\Omega_\mathrm{I}$. In the DDD subproblem, the new dislocation segments $\boldsymbol{s}_\mathrm{e}^{k+1}$ of all free dislocation segments (*i.e.*, segments that have at least one node in the continuum domain Ω_C, but not the pinned segments that are fully in the atomistic domain Ω_A) are computed based on the previous segments $\boldsymbol{s}_\mathrm{e}^k$ and the Peach-Koehler forces $\boldsymbol{f}_\mathrm{d}^k$ on the nodes the continuum domain Ω_C in the free part of the dislocation network. The DDD update runs in one or more sub-time steps, see Appendix A.3.

The first coupling phase, $\textcircled{c1}$, uses the updated free atomic positions $\boldsymbol{r}_\mathrm{a}^{k+1}$ to detect dislocation lines in

Algorithm 1: General CADD-3d algorithm: Quantities subscripted $*_a$, $*_d$, or $*_n$ relate to atomic, dislocation nodal or finite-element nodal degrees of freedom, respectively, and $*_e$ relates to dislocation segments. The superscript $*^k$ indicates the time step.

Phase	Molecular Dynamics	Dislocation Dynamics	Linear Elasticity
①	Predictor	Update segments	
Input:	$\boldsymbol{f}^k_\mathrm{a}, \boldsymbol{r}^k_\mathrm{a}, \boldsymbol{v}^k_\mathrm{a}\ \forall a \in \mathit{\Omega}_\mathrm{A}$	$\boldsymbol{f}^k_\mathrm{d}\ \forall d \in \mathit{\Omega}_\mathrm{C}$ $\boldsymbol{s}^k_\mathrm{e}\ \forall e \in \gamma^\mathrm{c}$	
Output:	$\boldsymbol{r}^{k+1}_\mathrm{a}\ \forall a \in \mathit{\Omega}_\mathrm{A}$ $\boldsymbol{v}^{k+1/2}_\mathrm{a}\ \forall a \in \mathit{\Omega}_\mathrm{A}$ $\boldsymbol{u}^{k+1}_\mathrm{n}\ \forall n \in \partial\mathit{\Omega}_\mathrm{I}$	$\boldsymbol{s}^{k+1}_\mathrm{e}\ \forall e \in \gamma^\mathrm{c}$	
$\textcircled{c1}$	Detect atomistic dislocations		
Input:	$\boldsymbol{r}^{k+1}_\mathrm{a}\ \forall a \in \mathit{\Omega}_\mathrm{A}$		
Output:	$\boldsymbol{s}^{k+1}_\mathrm{e}\ \forall e \in \gamma_\mathrm{AD}$		
$\textcircled{c2}$		Compute $\tilde{\bullet}$ contributions	
Input:		$\boldsymbol{s}^{k+1}_\mathrm{e}\ \forall e$	
Output:		$\tilde{\boldsymbol{\sigma}}^{k+1}_n\ \forall n \in \partial\mathit{\Omega}_\mathrm{t}$ $\tilde{\boldsymbol{u}}^{k+1}_n\ \forall n \in \partial\mathit{\Omega}_\mathrm{I} \cup \partial\mathit{\Omega}_\mathrm{u}$ $\tilde{\boldsymbol{u}}^{k+1}_a\ \forall a \in \mathit{\Omega}_\mathrm{P}$	
$\textcircled{c3}$			Compute $\hat{\bullet}$ boundary corrections
Input:			$\tilde{\boldsymbol{\sigma}}^{k+1}_n\ \forall n \in \partial\mathit{\Omega}_\mathrm{t}$, $\boldsymbol{u}^{k+1}_n, \tilde{\boldsymbol{u}}^{k+1}_n\ \forall n \in \partial\mathit{\Omega}_\mathrm{I} \cup \partial\mathit{\Omega}_\mathrm{u}$
Output:			$\hat{\boldsymbol{t}}^{k+1}_n\ \forall n \in \partial\mathit{\Omega}_\mathrm{t}$, $\hat{\boldsymbol{u}}^{k+1}_n\ \forall n \in \partial\mathit{\Omega}_\mathrm{I} \cup \partial\mathit{\Omega}_\mathrm{u}$
②			Solve elasticity
Input:			$\hat{\boldsymbol{t}}^{k+1}_n\ \forall n \in \partial\mathit{\Omega}_\mathrm{t}$, $\hat{\boldsymbol{u}}^{k+1}_n\ \forall n \in \partial\mathit{\Omega}_\mathrm{I} \cup \partial\mathit{\Omega}_\mathrm{u}$
Output:			$\hat{\boldsymbol{\sigma}}^{k+1}_d\ \forall d \in \mathcal{N}^\mathrm{DD}$, $\hat{\boldsymbol{u}}^{k+1}_n\ \forall n \in \mathit{\Omega}_\mathrm{C}$ $\hat{\boldsymbol{u}}^{k+1}_a\ \forall a \in \mathit{\Omega}_\mathrm{A}$
③	Compute forces	Compute forces	
Input:	$\boldsymbol{r}^{k+1}_\mathrm{a}\ \forall a \in \mathit{\Omega}_\mathrm{A}$, $\tilde{\boldsymbol{u}}^{k+1}_a, \hat{\boldsymbol{u}}^{k+1}_a\ \forall a \in \mathit{\Omega}_\mathrm{P}$	$\boldsymbol{s}^{k+1}_\mathrm{e}$, $\forall e$ $\hat{\boldsymbol{\sigma}}^{k+1}_d\ \forall d \in \mathcal{N}^\mathrm{DD}$	
Output:	$\boldsymbol{f}^{k+1}_\mathrm{a}\ \forall a \in \mathit{\Omega}_\mathrm{A}$	$\boldsymbol{f}^{k+1}_\mathrm{d}\ \forall d \in \mathcal{N}^\mathrm{DD}$	
④	Corrector		
Input:	$\boldsymbol{v}^{k+1/2}_\mathrm{a}, \boldsymbol{f}^{k+1}_\mathrm{a}\ \forall a \in \mathit{\Omega}_\mathrm{A}$,		
Output:	$\boldsymbol{v}^{k+1}_\mathrm{a}\ \forall a \in \mathit{\Omega}_\mathrm{A}$		

the atomistic domain Ω_{A} (see Appendix C for details). The lines are discretized into discrete dislocation segments $\boldsymbol{s}_{\mathrm{e}}^{k+1}$ with their nodes in Ω_{A} pinned to serve as boundary conditions for the DDD sub-problem.

During the subsequent coupling phase, $\textcircled{c2}$, the entire network of segments $\boldsymbol{s}_{\mathrm{e}}$ is evaluated to compute the stress contribution $\tilde{\boldsymbol{\sigma}}_{n}^{k+1}$ on the tractions assembled on boundary nodes in $\partial\Omega_{\mathrm{t}}$ and the displacement contribution to the FEM displacements $\tilde{\boldsymbol{u}}_{n}^{k+1}$ on the interface nodes in $\partial\Omega_{\mathrm{I}}$ and the displacement boundary nodes in $\partial\Omega_{\mathrm{u}}$, as well as to the atomic displacements $\tilde{\boldsymbol{u}}_{a}^{k+1}$ of the pad atoms in Ω_{P}.

The final coupling phase, $\textcircled{c3}$ applies the principle of superposition in order to correct the boundary conditions of the elastic problem: the $\tilde{\bullet}$ boundary contributions obtained in phase $\textcircled{c2}$ are used to compute the corrected traction $\hat{\boldsymbol{t}}_{n}^{k+1}$ on the traction boundary $\partial\Omega_{\mathrm{t}}$ and displacements $\hat{\boldsymbol{u}}_{n}^{k+1}$ on the interface $\partial\Omega_{\mathrm{I}}$ and the displacement boundary $\partial\Omega_{\mathrm{u}}$.

At this stage, the continuum domain has consistently updated boundary conditions for the end of the time step ($t = t_{k+1}$). These are now used in phase ② to compute the static equilibrium in the continuum domain Ω_{C}. The solution yields the $\hat{\bullet}$ displacement contribution for the FEM nodes $\hat{\boldsymbol{u}}_{n}^{k+1}\,\forall n \in \Omega_{\mathrm{C}}$ and pad atoms $\hat{\boldsymbol{u}}_{a}^{k+1}\,\forall a \in \Omega_{\mathrm{P}}$, and the $\hat{\bullet}$ stress contribution on the dislocations nodes $\hat{\boldsymbol{\sigma}}_{d}^{k+1}\,\forall d \in \Omega_{\mathrm{C}}$. At the end of this phase, all positional degrees of freedom (in particular, the positions of the pad atoms and the pinned dislocation nodes), and the $\hat{\bullet}$ stress contributions at dislocation nodes are up to date for $t = t_{k+1}$, with one exception: in the case where dislocation segments intersect the pad and the interface, some of the pad and interface atoms are contained in the highly non-linear dislocation core, where their displacements are poorly described by the DDD solution. The positions of these atoms need to be corrected and we developed the concept of the *core template* for that purpose (details in Section 3).

The forces for the dynamic sub-problems are computed in phase ③. The MD sub-problem uses the positions of the free atoms $\boldsymbol{r}_{\mathrm{a}}^{k+1}\,\forall a \in \Omega_{\mathrm{A}}$ and the superposition principle for the pad atoms $\boldsymbol{r}_{\mathrm{a}}^{k+1} = \boldsymbol{r}_{\mathrm{a}}^{0} + \tilde{\boldsymbol{u}}_{n}^{k+1} + \hat{\boldsymbol{u}}_{n}^{k+1}\,\forall a, n \in \Omega_{\mathrm{P}}$ to evaluate the forces on free atoms $\boldsymbol{f}_{\mathrm{a}}^{k+1}\,\forall a \in \Omega_{\mathrm{A}}$. This superposition is valid as long as no dislocation core crosses the pad region. Coupling the displacement field of dislocation cores needs corrective terms that are presented in section 3. The DDD sub-problem uses the segments $\boldsymbol{s}_{\mathrm{e}}^{k+1}$ together with the $\hat{\bullet}$ stress contributions on the free dislocation nodes $\hat{\boldsymbol{\sigma}}_{d}^{k+1}\,\forall d \in \Omega_{\mathrm{C}}$ to evaluate the Peach-Koehler forces on those same free nodes $\boldsymbol{f}_{\mathrm{d}}^{k+1}\,\forall d \in \Omega_{\mathrm{C}}$.

Finally, using the updated atomic forces $\boldsymbol{f}_{\mathrm{a}}^{k+1}$, the free atoms velocities $\boldsymbol{v}_{\mathrm{a}}^{k+1}\,\forall a \in \Omega_{\mathrm{A}}$ are modified, following the predictor-corrector scheme, in phase ④ by applying the corrector step of the velocity-Verlet scheme. We show in the companion paper CADD-3d.III that one can take advantage of the slower timescale of dislocation motion with respect to atomistic motion and improve Algorithm 1's efficiency by executing multiple MD time steps per DDD step.

2.2. Dislocation detection

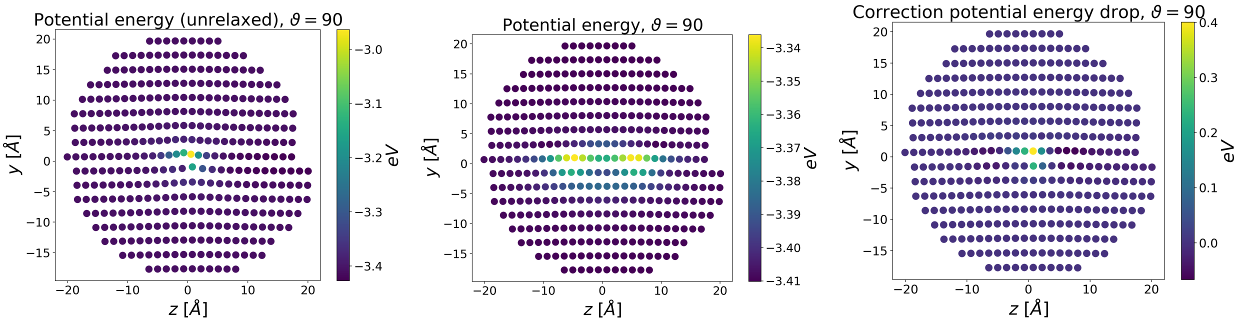
We mentioned the detection of dislocation segments in the atomistic domain (phase $\textcircled{c1}$ in Algorithm 1). Detection is a shorthand notation for the extraction of a line description of dislocation segments from a set of atoms within which dislocation cores are embedded. A detection method is present in the original two-dimensional version of CADD [see 33, for details], but since only edge dislocations exist and on a reduced set of slip planes (three for a hexagonal close-packed plane), the problem is simpler than in three dimensions . For the more general slip systems and character angles existing in 3d, we have adapted the method proposed by Stukowski et al. [36] and Stukowski and Albe [35]. The *Dislocation Extraction Algorithm* (DXA) performs Burgers circuit analyses and computes a set of line segments representing the entire dislocation network. To the best of our knowledge, DXA has never been applied for on-going dislocation network evolution as needed in CADD-3d. We have consequently developed a variant of DXA specifically tailored for our purposes. The basic ideas of the detection algorithm are summarized in Appendix C.

2.3. Static resolution

For solving static problems, one can employ standard optimization approaches using gradients. As a matter of fact, the gradients of the total energy are the force vectors $\boldsymbol{f}_{\mathrm{a}}$, $\boldsymbol{f}_{\mathrm{d}}$ necessary to compute optimal search directions. These could be exploited in conjugate gradient-type minimization schemes, if the total potential energy is corrected to avoid ghost forces [32]. In that case, the algorithm is changed by removing step $\textcircled{4}$ and changing step $\textcircled{1}$ for the construction of the new exploratory direction and the line search. However, an important difficulty could arise due to the detection stage $\textcircled{c1}$. This is intrinsically non-smooth and could prevent convergence by creating oscillations in the position of the detected segments. Studying convergence properties of a global optimization procedure is beyond the scope of the present series of papers. Instead, a domain decomposition and an alternative schwartz procedure where each model is equilibrated with boundary conditions fixed from the other counter parts is adopted. This strategy has been employed in the companion paper CADD-3d.II and will be described in more details there.

3. Accurate atomic boundary conditions for hybrid dislocations: the *Core Template*

The continuum Volterra solution for the dislocation displacement field is a poor representation of the true field within the dislocation core. For a hybrid dislocation in the pad region, i.e. those dislocation segments passing from the continuum region to the atomistic region, the use of the Volterra solution creates serious artifacts. That is, if the pad atom displacements are imposed using the Volterra DDD solution, then real MD atoms near the core will be subject to large spurious forces due to the error in the pad atom positions. Such errors are especially egregious for fcc dislocations that dissociate into two partials separated by a stacking fault. Figure 4 illustrates this point by considering the per-atom potential energy profile obtained for an edge dislocation in fcc aluminum in the cases when a) the Volterra displacement field is used and b) after full atomic relaxation is permitted resulting in two partial dislocations separated by a stacking fault. In Figure 4c is shown the energy change per atom upon relaxation, showing that the Volterra solution introduces spurious energy reaching 0.4 eV. Minimizing the effect of these spurious energies and the associated forces represents one of the main efforts in the present work.



(a)

(b)

(c)

Figure 4: Per atom potential energy extracted from a section normal to an edge dislocation line ($\vartheta = 90^{circle}$) in fcc aluminum modeled with the Mendelev EAM potential [28], with a) the configuration given by the Volterra expression, b) the relaxed and dissociated core structure and c) the difference of both that shows the energy reduction provided by the relaxation.

The goal of an accurate coupling is to have the real MD atoms experience accurate forces due to realistic atomic displacements in the pad atoms. This can be achieved by enriching the continuum displacement field with the displacement field of the true atomistic dislocation when and where such enrichment is necessary. We envision the solution as follows. Let us imagine that any dislocation is described by a true discrete atomistic core of displaced atoms within some inner radius R^{core} and by a continuum displacement field beyond R^{core}. The entire DDD methodology would then proceed unchanged, but with the $\hat{\boldsymbol{u}}$ field being enriched by an atomistic description $\Delta\tilde{\boldsymbol{u}}^{\text{corr}}$ when the distance to the dislocation line for $r < R^{\text{core}}$. With such an enriched field, $\tilde{\boldsymbol{u}}$ $+\Delta\tilde{\boldsymbol{u}}^{\text{corr}}$, the imposition of the pad atom displacements automatically includes the atomistically-enriched core structure for any dislocations within R^{core} of the pad region, and especially for hybrid dislocations that pass into the atomistic region. We now recognize, however, that the atomistically-enriched field for $r < R^{\text{core}}$ is in fact only necessary for precisely those dislocation segments intersecting the pad region. Outside of this region, the dislocations communicate with the pad through the displacement field $\hat{\boldsymbol{u}} + \tilde{\boldsymbol{u}}$, and interact with each other through the stress field $\hat{\boldsymbol{\sigma}} + \tilde{\boldsymbol{\sigma}}$. There is thus no need for the atomic resolution of the core in regions away from the pad region. In fact, the entire development of the DDD method is based on the understanding that the dislocation core structures are not important for modeling of long-range dislocation interactions. Therefore, the atomistically-enriched field is, in practice, only necessary for the hybrid dislocation segments in the pad region itself. We refer to this atomistic enrichment field as the *core template correction* because the computed atom positions provide a template of a realistic atomistic dislocation core environment for the real atoms in the MD domain. A related approach was used in plane-strain CADD for dislocations approaching the coupling interface, and was shown to significantly reduce the magnitude of spurious Peach-Koehler forces on both atomistic and continuum dislocations, see [13].

Figure 5 illustrates the application of the template correction in the case where a dislocation line intersects the interface. Without a template, the pad atoms have reference positions of the crystalline lattice with displacements given by the superposition of the corrective field $\hat{\boldsymbol{u}}$ and the infinite-space field $\tilde{\boldsymbol{u}}$ of all the dislocations, see (A.8). The true atomic displacements of the pad atoms (displacement field in $\varOmega_{\mathrm{P}}$) can be represented as a sum of contributions (after phase ②) as

$$\boldsymbol{u}_a = \hat{\boldsymbol{u}}(\boldsymbol{x}_a) + \tilde{\boldsymbol{u}}(\boldsymbol{x}_a, \gamma^{\mathrm{c}}, \gamma^{\mathrm{v}}) + \varDelta\tilde{\boldsymbol{u}}^{\mathrm{corr}}(\boldsymbol{x}_a) \quad \forall \boldsymbol{x}_a \in \varOmega_{\mathrm{P}}, \tag{1}$$

where the unknown corrective term $\varDelta\tilde{\boldsymbol{u}}^{\mathrm{corr}}(\boldsymbol{x}_a)$ accounts for all the non-linear material response around the dislocation core. The *exact core template* correction $\varDelta\tilde{\boldsymbol{u}}^{\mathrm{corr}}(\boldsymbol{x}_a)$ for any atom a is then the error of the semi-analytic contribution $\tilde{\boldsymbol{u}}(\boldsymbol{x}_a)$ of a dislocation network in a system free of elastic strain ($\hat{\boldsymbol{u}}(\boldsymbol{x}_a) = \mathbf{const}$),

$$\varDelta\tilde{\boldsymbol{u}}^{\mathrm{corr}}(\boldsymbol{x}_a) = \boldsymbol{u}_a - \tilde{\boldsymbol{u}}(\boldsymbol{x}_a), \tag{2}$$

Far from any dislocation core, non-linear effects become negligible, and the small-strain linear elasticity solution $\tilde{\boldsymbol{u}}$ converges with the atomistic solution $\boldsymbol{u}$. With $r(\boldsymbol{x})$ defined as the distance between $\boldsymbol{x}$ and the nearest dislocation core, this is expressed as

$$\lim_{r(\boldsymbol{x})\to+\infty} ||\varDelta\tilde{\boldsymbol{u}}^{\mathrm{corr}}(\boldsymbol{x})|| = 0. \tag{3}$$

We can thus define a practical core radius R^{core} as the smallest radius for which

$$||\varDelta\tilde{\boldsymbol{u}}^{\mathrm{corr}}(\boldsymbol{x})|| < \kappa \tag{4}$$

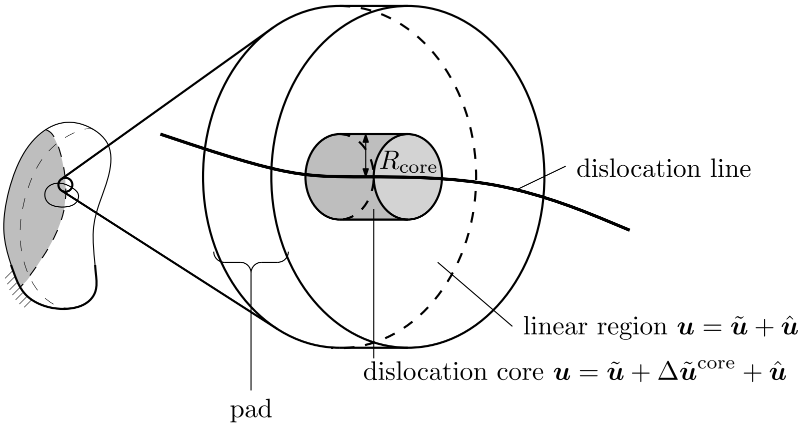
for a suitably chosen tolerance $\kappa > 0$. Outside of R^{core}, the atomistic corrections are deemed negligible (errors in displacement less than κ). An error estimation of the template enables the choice of a suitable core radius as detailed in Appendix B. The core template is only imposed within R^{core}, as indicated by the gray region in Figure 5. More formally, the dislocation core region around a core segment s is the domain

$$\varOmega_s^{\mathrm{core}}(\kappa) = \left\{\boldsymbol{x} \in \varOmega_{\mathrm{P}} : \mathrm{dist}(s, \boldsymbol{x}) < R^{\mathrm{core}}\right\}, \tag{5}$$

where dist defines the euclidian distance between a line and a point. Also, we assume that dislocations are sparse in the pad such that no point is within the core region of more than one core, *i.e.* $\bigcap\limits_s \varOmega_s^{\mathrm{core}}(\kappa) = \varnothing$. Complicated cases, such as dislocation junctions, or structural changes due to applied stresses are not taken into account in the present algorithm. Algorithms for passing of junctions, for instance, remain to be developed along the lines of dislocation passing in CADD2d [33].

3.1. Implementation of core templates

To implement a continuous core template correction field for arbitrary character angles, we define a continuous function $\boldsymbol{u}(\boldsymbol{x})$ that interpolates the displacement field between (point-wise defined) atomistic core structures at various character angles. The true core structure of a dislocation at a position $\boldsymbol{x}$ depends on the local character angle $\vartheta(\boldsymbol{x})$, the curvature around $\boldsymbol{x}$, and the applied stress (which distorts the core, allowing it to overcome the Peierls barrier when the applied stress equals the Peierls stress). The use of DDD implicitly assumes a small dislocation curvature over the scale of the dislocation core size. It possible, but not necessary at this stage, to model the stress dependence of the core structure, which is mainly important for the atoms within a few ångströms of the center of the (partial) dislocation(s). This also excludes core structural changes (large displacements) due to large stresses. These assumptions, however, allow us



R_{core}
dislocation line
linear region $\boldsymbol{u} = \tilde{\boldsymbol{u}} + \hat{\boldsymbol{u}}$
dislocation core $\boldsymbol{u} = \tilde{\boldsymbol{u}} + \Delta\tilde{\boldsymbol{u}}^{\text{core}} + \hat{\boldsymbol{u}}$
pad

Figure 5: Schematic of dislocation core in the pad. For most of the pad, the linear elastic discrete dislocation dynamics displacement solution is adequate, in the vicinity of the dislocation line, however, adding the core template $\varDelta\tilde{\boldsymbol{u}}^{\text{corr}}$ is required to minimise spurious forces.

to consider straight dislocation segments at given fixed character angles to build the *core template*. A set of such templates, with various character angles, can be conveniently precomputed and then interpolated during the coupled simulation. It is possible in principle to create a library of core templates for dislocations under Escaig (non-glide) stresses. Such stresses may not always be negligible but standard DDD models do not represent the dislocation as two dissociated partials, and therefore the effects of Escaig stresses are not addressed here.

A set of straight atomistic dislocation cores spanning the full range of character angles[3] is thus generated using molecular statics minimization, c.f. [10] for details. The atomic displacements around the dislocation line are extracted as illustrated on Figure 6b. For any given character angle ϑ_i, the atomic displacement $\boldsymbol{u}_{\vartheta_i}(\boldsymbol{x}_a)$ varies along the dislocation line $\boldsymbol{\xi}$. Because the dislocation is straight, we choose a coordinate system with line vector $\boldsymbol{\xi} = \boldsymbol{x}$ coinciding with the x axis and the normal to the slip plane $\boldsymbol{n} = \boldsymbol{y}$ coinciding with the y axis (c.f. Figure 6a). We set the origin on the dislocation line so that all atomic positions $\boldsymbol{x}_a = (x_a, y_a, z_a)$ are relative to the dislocation core. We then project the atomic positions onto the z-y plane as shown in Figures 6(c, d), which reveals that the core template correction (i) can indeed safely be approximated as a function of z, y coordinates only and (ii) is always sufficiently smooth to validate the approximation that excludes the variations in the $\boldsymbol{\xi}$ direction. Note that the density of points in the projection normal to $\boldsymbol{\xi}$ varies with the dislocation character angle. With the atomic positions projected onto two dimensions, it is straightforward to subtract $\tilde{\boldsymbol{u}}(\boldsymbol{x}_a)$ from the displacements, triangulate the residual displacements, and use first-order finite elements to interpolate the point-wise correction term $\varDelta\tilde{\boldsymbol{u}}^{\text{corr}}_{\vartheta_i}(\boldsymbol{x}_a)$. This procedure yields a continuous correction field $\varDelta\tilde{\boldsymbol{u}}^{\text{corr}}_{\vartheta_i}(\boldsymbol{x})$ at each character angle ϑ_i. For an arbitrary character angle $\vartheta_i \leq \vartheta < \vartheta_{i+1}$, we simply interpolate linearly between the two nearest solutions as

$$\varDelta\tilde{\boldsymbol{u}}^{\text{corr}}_{\vartheta} = \alpha\, \varDelta\tilde{\boldsymbol{u}}^{\text{corr}}_{\vartheta_i} + (1-\alpha)\, \varDelta\tilde{\boldsymbol{u}}^{\text{corr}}_{\vartheta_{i+1}} \qquad \text{with} \quad \alpha = \frac{\vartheta_{i+1} - \vartheta}{\vartheta_{i+1} - \vartheta_i},$$

The near-elimination of any spurious forces on hybrid dislocations at the atomistic/continuum interface using this continuous interpolated solution for the core template correction is assessed quantitatively in the companion papers CADD-3d.II and CADD-3d.III.

4. Discussion

In this first paper of a series of three papers, we have presented the key concepts and algorithmic details for the intimate coupling of a fully atomistic domain to a surrounding domain described by a nodal discrete dislocation dynamics domain in full 3d. The goal of this CADD-3d model is to achieve full atomistic

[3] Because, the *core template* will vary with the character angle one has to consider as many as possible.

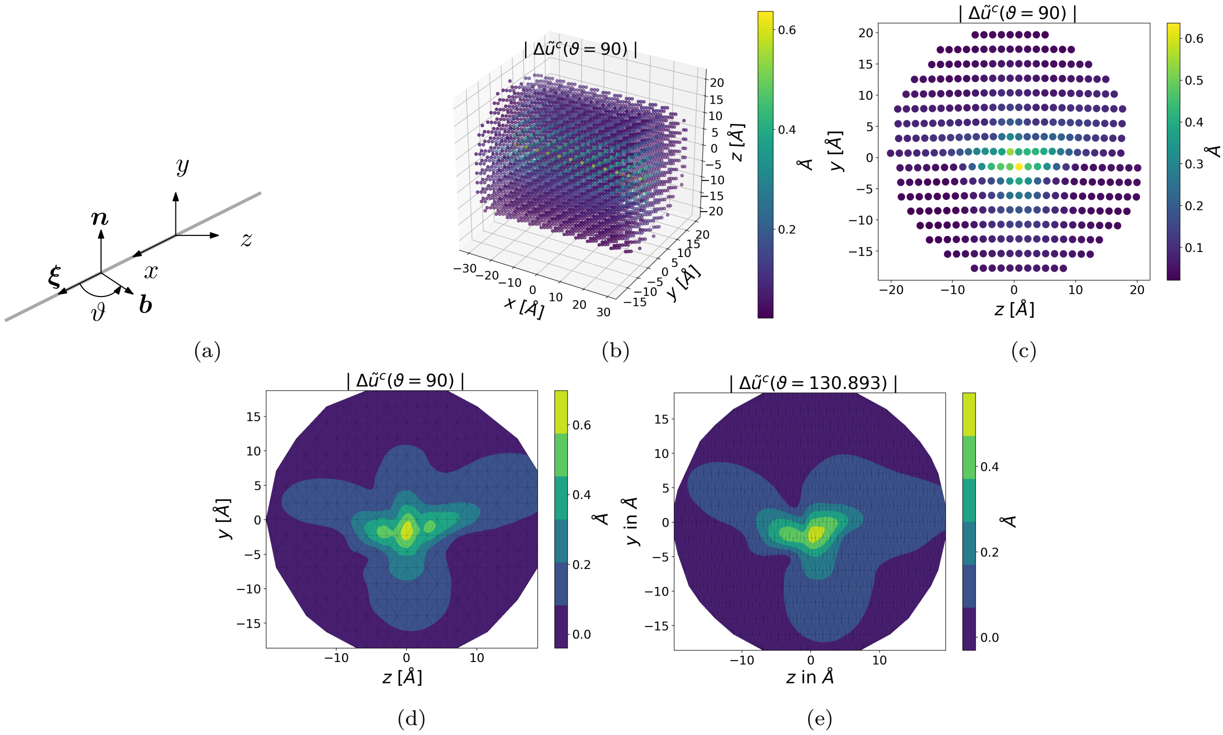


Figure 6: (a) Geometric setup for core template extraction. The heavy grey line represents the straight infinite dislocation line. Because the line direction $\boldsymbol{\xi}$ coincides with the x axis and the normal to the slip plane $\boldsymbol{n}$ with the y axis, the Burgers vector $\boldsymbol{b}$ is constrained to be in the $x-z$ plane. The character angle of the dislocation is ϑ. The extracted configurations are for a dislocation line in aluminum as described by the Mendelev EAM potential [28] following Cho et al. [10] approach. (b) The template correction $\Delta\tilde{\boldsymbol{u}}^{\mathrm{corr}}$ for a cylinder of atoms around a straight dislocation core. (c) The template correction $\Delta\tilde{\boldsymbol{u}}^{\mathrm{corr}}$ of the edge dislocation. (d) Interpolated template correction $\Delta\tilde{\boldsymbol{u}}^{\mathrm{corr}}$ of the edge dislocation (e) Interpolated template correction $\Delta\tilde{\boldsymbol{u}}^{\mathrm{corr}}$ of the mixed angle dislocation $\vartheta = 130.893$.

resolution in regions where complex atomistic deformations are occuring that cannot be described solely by dislocation dynamics but that are strongly influenced by, and also influence, surrounding dislocation plasticity. The methodology proceeds by domain decomposition, reducing the problem to relatively standard MD and DDD problems that are coupled through mutual evolving displacement boundary conditions at the atom/continuum interface. We have focused particularly on treatment of hybrid dislocations, i.e. those dislocations that span the atomistic/continuum interface, with a careful description of both the algorithm for evolving the entire system and the core-template algorithm for minimizing spurious forces on atoms near the interface.

Accuracy of the method is largely dictated by the quality of the calibration of the DDD material parameters to the corresponding atomistic values. With accurate calibration, CADD-3d harnesses the power of both the MD code LAMMPS and the DDD code ParaDis. Current approximations in the most commonly-used version of ParaDis (isotropic elasticity, small-strain deformation, core energy models, the functional forms for mobility laws, kinks/jogs due to dislocation junctions) can be systematically improved. The widely-available version of ParaDis is easily modified to change the dislocation mobility law and core energy, while an anisotropic elasticity code also exists. Thus, the accuracy of the DDD method that underlies CADD-3d can be systematically advanced toward full fidelity, leaving only coupling errors of CADD-3d itself. In the second paper of this series, we thus study the quasistatic behavior of dislocations in Al, which is fairly isotropic and for which we have previously determined an accurate core energy, enabling careful assessment of errors introduced by the coupling method. In the third paper, we study the dynamic evolution of various dislocation configurations in Al using carefully calibrated mobility laws, enabling further assessment of the CADD-3d method and demonstrating its use in modeling complex multiscale plasticity problems.

We have not discussed here the passing of dislocations in and out of the atomistic domain, i.e. the creation of hybrid dislocations. In the third paper, we introduce a method that is suitable for problems consisting only of collections of interacting dislocations without any other defects. This limit is, however, quite useful in many cases. In particular, CADD-3d can be used to insert an MD box of atoms anywhere and anytime into any fully DDD problem to analyze the atomistic behavior in a local region where the existing DDD method may be inadequate. We will discuss more general passing algorithms in future work.

Appendix

A. The Models

A.1. Linear elastic continuum Mechanics

Linear elasticity is certainly the simplest model that can provide a prediction of the stresses and strains for a body subjected to constraints such as forces, tractions and displacements. When we consider the body/domain as $\mathit{\Omega}_\mathrm{C}$, the partial differential equation to be solved is classically

$$\forall \boldsymbol{x} \in \mathit{\Omega}_\mathrm{C}, \quad \sigma_{ij,j}(\boldsymbol{x}) = 0 \tag{A.1}$$

under the boundary conditions

$$\forall \boldsymbol{x} \in \partial\mathit{\Omega}_\mathrm{t}, \quad \boldsymbol{\sigma}(\boldsymbol{x})\boldsymbol{n}(\boldsymbol{x}) = \boldsymbol{t}_\mathrm{ext}(\boldsymbol{x}) \tag{A.2}$$

$$\forall \boldsymbol{x} \in \partial\mathit{\Omega}_\mathrm{u}, \quad \boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}_\mathrm{ext}(\boldsymbol{x}) \tag{A.3}$$

where $\boldsymbol{u}_\mathrm{ext}$ and $\boldsymbol{t}_\mathrm{ext}$ are prescribed displacements and tractions respectively, and $\boldsymbol{n}$ is the outward normal to the boundary $\partial\mathit{\Omega}_\mathrm{t}$. Under the small strain hypothesis, the constitutive law relating small strains $\boldsymbol{\varepsilon}$ with stresses can be formulated as:

$$\forall \boldsymbol{x} \in \mathit{\Omega}_\mathrm{C}, \quad \sigma_{ij}(\boldsymbol{x}) = C_{ijkl}\varepsilon_{kl}(\boldsymbol{x}) \tag{A.4}$$

where C_{ijkl} is a constant fourth order tensor. For the purpose of the proposed coupling method, a body not containing any dislocation, *i.e.* where no plastic deformation occurred, will be modeled with linear elasticity. The elasto-static equations can naturally be solved by means of the FEM. It is also possible to consider a boundary integral formulation when Green's functions complying with the boundary requirements are accessible. We want to emphasize here that the proposed approach is in principle agnostic to the employed linear elastic solver as long as Neumann and Dirichlet boundary conditions with anisotrope constitutive laws can be handled.

A.2. Molecular dynamics

We use classical molecular dynamics to simulate matter at the atomic scale by modelling point mass atoms subject to interatomic potentials. Born and Oppenheimer [6] formulated this approximation which is well funded if the nuclei move much slower than the electrons (*i.e.*, in mechanics at moderate temperatures without chemical reactions). In such a case, the distribution of electrons adapts in a quasi-static manner to the positions of the nuclei, and the interatomic forces (*i.e.* forces between nuclei) depend only on their respective positions. It can be shown that the forces are therefore conservative and derive from a potential. The force $\boldsymbol{f}_i$ acting on any atom i of a system of N atoms is:

$$\boldsymbol{f}_i = -\nabla_{\boldsymbol{r}_i} U^a(\boldsymbol{r}_1, \boldsymbol{r}_2, \cdots, \boldsymbol{r}_N), \tag{A.5}$$

where U^a is the potential energy of the system and $\boldsymbol{r}_j,\ j \in \{1, 2, \cdots, N\}$ are the atomic positions. The governing equation of a system modeled by molecular dynamics is the Newton's second law:

$$m_i\ddot{\boldsymbol{r}}_i = \boldsymbol{f}_i, \tag{A.6}$$

where m_i is the mass of the i-th atom. Different (explicit) integration schemes such as Verlet and Leap-Frog are commonly used to integrate (A.6) in time, but also any classical mechanics integrator such as Newmark's corrector-predictor algorithm could be used. More details can be found in Griebel et al. [19], Rapaport [31], Berendsen [4].

MD is conceptually very simple and the main difficulty in its use is the sheer size of molecular systems. As an illustration, the number N of atoms in $V = 1\,\mathrm{mm}^3$ of pure aluminum is

$$N = N_\mathrm{a}\frac{\varrho V}{m_\mathrm{Al}} \approx 1.8 \cdot 10^{20}, \tag{A.7}$$

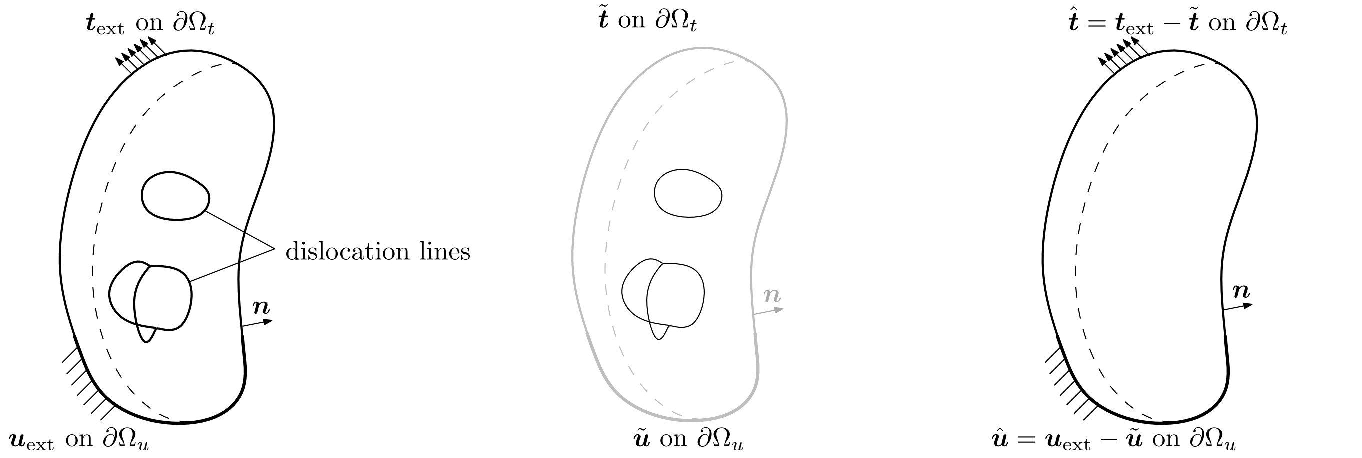
where N_a is Avogadro's constant, ϱ and m_Al are the solid density and the molar mass respectively. Such a number of atoms is at least ten orders of magnitude larger than what a single computer nowadays can handle. Molecular dynamics relies heavily on high performance computing using parallel machines and is restricted to small simulation boxes.

Furthermore, the time steps necessary in the integration (A.6) is of the order of $1\,\mathrm{fs} = 1 \times 10^{-15}\,\mathrm{s}$, which is extremely small from a mechanics point of view.

A.3. Discrete dislocation mechanics

As previously mentioned, dislocations are structural defects within an atomic lattice. Under loading conditions, these can undergo motion that can be modeled. This is the direction taken by discrete dislocation mechanics where dislocations are approximated with segments and where force-displacement relations allow a proper dynamical evolution.

Dislocation mechanics models [17, 9] target the problem illustrated in Figure A.7a. A volume $\varOmega$ with boundary $\partial\varOmega$ is subject to an external traction $\boldsymbol{t}_\mathrm{ext}$ on a part $\partial\varOmega_t$ of its boundary and an imposed displacement $\boldsymbol{u}_\mathrm{ext}$ on the remainder $\partial\varOmega_u = \partial\varOmega \setminus \partial\varOmega_t$ of the boundary.


$\boldsymbol{t}_\mathrm{ext}$ on $\partial\Omega_t$
dislocation lines
$\boldsymbol{n}$
$\boldsymbol{u}_\mathrm{ext}$ on $\partial\Omega_u$
$\tilde{\boldsymbol{t}}$ on $\partial\Omega_t$
$\boldsymbol{n}$
$\tilde{\boldsymbol{u}}$ on $\partial\Omega_u$
$\hat{\boldsymbol{t}} = \boldsymbol{t}_\mathrm{ext} - \tilde{\boldsymbol{t}}$ on $\partial\Omega_t$
$\boldsymbol{n}$
$\hat{\boldsymbol{u}} = \boldsymbol{u}_\mathrm{ext} - \tilde{\boldsymbol{u}}$ on $\partial\Omega_u$

(a) Full initial and boundary value problem: The volume $\varOmega$ contains a dislocation network. External tractions $\boldsymbol{t}_\mathrm{ext}$ act on $\partial\varOmega_t$ and a displacement $\boldsymbol{u}_\mathrm{ext}$ is imposed on $\partial\varOmega_u$.

(b) Dislocation network in infinite space. The displacement $\tilde{\boldsymbol{u}}$ is evaluated on $\partial\varOmega_u$, and the traction $\tilde{\boldsymbol{t}} = \tilde{\boldsymbol{\sigma}}\,\boldsymbol{n}$ on $\partial\varOmega_t$.

(c) Boundary conditions are corrected for the elastic problem by subtracting the contributions of the dislocation network that violate the boundary conditions.

Figure A.7: Discrete dislocations problem, linearity is exploited to use superposition of $\tilde{\bullet}$ and $\hat{\bullet}$ solutions.

The volume contains a network of dislocations. In two-dimensional problems, the dislocation network is a set of individual straight parallel edge dislocations whereas three-dimensional networks are made of complex and curved paths. Only linear elastic material and small strains are considered, which allows to use the superposition principle, *i.e.* to write the displacement $\boldsymbol{u}$, strain $\boldsymbol{\varepsilon}$ and stress field $\boldsymbol{\sigma}$ as sums of an elastic contribution and a plastic contribution due to the dislocation network

$$\boldsymbol{u} = \hat{\boldsymbol{u}} + \tilde{\boldsymbol{u}}, \quad \boldsymbol{\varepsilon} = \hat{\boldsymbol{\varepsilon}} + \tilde{\boldsymbol{\varepsilon}}, \quad \boldsymbol{\sigma} = \hat{\boldsymbol{\sigma}} + \tilde{\boldsymbol{\sigma}}, \tag{A.8}$$

where $(\tilde{*})$ denotes contributions of the dislocation network in an infinite space and $(\hat{*})$ the elastic boundary-value contribution. The fields associated with the dislocation network are the superposition of the fields $\tilde{\boldsymbol{u}}_i$,

$\tilde{\boldsymbol{\varepsilon}}_i$, $\tilde{\boldsymbol{\sigma}}_i$ due to all the n dislocation segments,

$$\tilde{\boldsymbol{u}} = \sum_{i=1}^{n} \tilde{\boldsymbol{u}}_i, \quad \tilde{\boldsymbol{\varepsilon}} = \sum_{i=1}^{n} \tilde{\boldsymbol{\varepsilon}}_i, \quad \tilde{\boldsymbol{\sigma}} = \sum_{i=1}^{n} \tilde{\boldsymbol{\sigma}}_i. \tag{A.9}$$

The solid is considered infinite for the contributions of individual dislocation segments, see Figure A.7b and the displacements $\tilde{\boldsymbol{u}}$ and traction $\tilde{\boldsymbol{t}} = \tilde{\boldsymbol{\sigma}}\,\boldsymbol{n}$ are evaluated of $\partial\Omega_u$ and $\partial\Omega_t$, respectively. These fields are then used to correct the boundary conditions (see Figure A.7c), which may be solved using any suitable elasto-static method.

A.3.1. Dislocation strain energy and Peach-Koehler forces

The strain energy density caused by the presence of a dislocation in an infinite medium is at the source of the displacement field $\tilde{\boldsymbol{u}}$ and also of the effective force acting on the line. Let us name $\mathcal{C}$ the line path that characterizes the dislocations of interest. Then a purely linear elastic approach is accurate except if close to the dislocation line $\mathcal{C}$ where the stress field diverges. The classical approach to circumvent this problem is to regularize the strain field by spreading the Burgers vector over a distance a around the dislocation line. In the end we obtain the following total strain energy:

$$\begin{aligned} U^{c} &= \int \mathrm{d}U^{\mathrm{el}}(\boldsymbol{x}, a) d\boldsymbol{x} + \int \mathrm{d}U^{\mathrm{core}}(\boldsymbol{x}, a) d\boldsymbol{x} \\ &= U^{\mathrm{el}}(\mathcal{C}, a) + U^{\mathrm{core}}(\mathcal{C}, a) \end{aligned} \tag{A.10}$$

where we have named two contributions, U^{el} and U^{core}, as the non-singular elastic and core contributions to the strain energy, both depending on the path $\mathcal{C}$ and a regulating parameter a, since these volume integrals can be expressed as a line integral over $\mathcal{C}$ [22, 8]. By computing the variation of the energies U^{c} and U^{core} with respect to a change of dislocation configuration, we obtain an effective line load that applies to the dislocation line. The Peach-Koehler [22] formula provides the forces experienced by an infinitesimal dislocation segment when subject to a stress-field:

$$\boldsymbol{f}^{\mathrm{PK}}(\boldsymbol{x}) = (\boldsymbol{\sigma}(\boldsymbol{x}) \cdot \boldsymbol{b}(\boldsymbol{x})) \times \boldsymbol{\xi}(\boldsymbol{x}) \tag{A.11}$$

A.3.2. Dislocation discretization

Since dislocations are line defects, it is natural to decompose any such line into sub-elements. From classical definitions, a dislocation is characterized with a path and a Burgers vector which may vary along this line. In DDD, any dislocation is represented as a set of edges taken from γ that connect the nodes in the set $\mathcal{N}$. Any edge $e \in \gamma$ is a pair of indices stating the nodes connected by element e. Each node $n \in \mathcal{N}$ is having its coordinate labeled $\boldsymbol{x}_n$. An example of such a line description is illustrated in Figure A.8.

In implementations such as ParaDis [9], the dislocation lines are approximated using linear segments and it is convenient to define the Burgers vector as a per-segment and per-direction constant (see, e.g., Bulatov and Cai [7, Section 10.1.1]). Thus we denote $\boldsymbol{b}_{i,j}$ the constant Burgers vector along the segment that connects nodes i and j in direction $\boldsymbol{t}_{i,j}$. For this convention, it follows that $\boldsymbol{b}_{i,j} = -\boldsymbol{b}_{j,i}$ since $\boldsymbol{t}_{i,j} = -\boldsymbol{t}_{j,i}$, see inset in Figure A.8. The principle of conservation of the Burgers vector becomes that $\sum_j \boldsymbol{b}_{i,j} = \boldsymbol{0}$ across any node, see inset in Figure A.8. Note that this choice of implementation trivially guarantees that the Burgers vector is conserved along any simple loop iterated over in either direction.

The line force given in (A.11) can be integrated to nodal consistent elastic forces:

$$\boldsymbol{f}^{\mathrm{el}}_i = \oint_{\mathcal{C}} \boldsymbol{f}^{\mathrm{PK}}(\boldsymbol{x}) N_i(\boldsymbol{x}) \boldsymbol{dL}(\boldsymbol{x}) \tag{A.12}$$

where $N_i(\boldsymbol{x})$ stands for the shape function associated with node i. The next stage of discretization approximates the interaction stress as a sum of contributions from the segments by using the following property:

Property 1. *Stresses $\tilde{\boldsymbol{\sigma}}$ generated by a closed dislocation path. Let a dislocation closed-path $\mathcal{C}$ being discretized with segments taken from γ. Then the total contribution to the stress $\tilde{\boldsymbol{\sigma}}$ can be written as the sum*

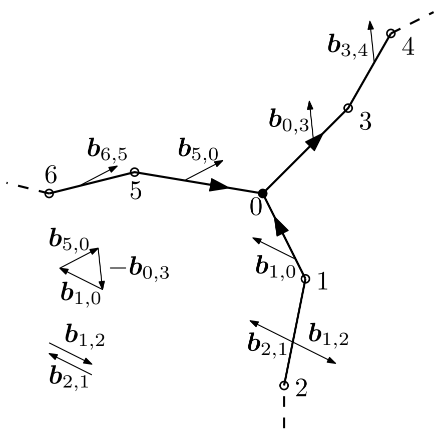


Figure A.8: Dislocation network represented as a set of “nodes” interconnected by straight segments. Empty circles represent simple computational discretization nodes. The full disk (node 0) is both a computational node and a physical node, since it represents a dislocation triple-juncture. Note that two neighboring segments have the same Burgers vector if the connecting node is a simple discretization node (*e.g.* $\boldsymbol{b}_{2,1} = \boldsymbol{b}_{1,0}$). The sum of all incoming Burgers vectors matches the sum of all outgoing Burgers vectors across a node.

of the contributions of the segments:

$$\forall x \quad \tilde{\boldsymbol{\sigma}}(\boldsymbol{x}) = \sum_{e\in\gamma} \tilde{\boldsymbol{\sigma}}_e(\boldsymbol{x})$$

With such a superposition property, the discrete Peach-Koehler becomes:

$$\boldsymbol{f}^{\mathrm{PK}}(\boldsymbol{x}) = \left[\sum_{e\in\gamma} (\tilde{\boldsymbol{\sigma}}_e(\boldsymbol{x}) \cdot \boldsymbol{b}) \times \boldsymbol{\xi}(\boldsymbol{x})\right] + (\hat{\boldsymbol{\sigma}}(\boldsymbol{x}) \cdot \boldsymbol{b}) \times \boldsymbol{\xi}(\boldsymbol{x}) \tag{A.13}$$

Discretizing the contour integral of (A.12) we obtain the following approximation[4]:

$$\boldsymbol{f}_i^{\mathrm{el}} = \sum_{e\in\gamma} \sum_{e'} \int_{e'} \left\{ [(\tilde{\boldsymbol{\sigma}}_e(\boldsymbol{x}) \cdot \boldsymbol{b}) \times \boldsymbol{\xi}(\boldsymbol{x})] + (\hat{\boldsymbol{\sigma}}(\boldsymbol{x}) \cdot \boldsymbol{b}) \times \boldsymbol{\xi}(\boldsymbol{x}) \right\} N_i(\boldsymbol{x}) \boldsymbol{dL}(\boldsymbol{x}). \tag{A.14}$$

Overall it can be summarized with the following formula for the linear elastic force:

$$\boldsymbol{f}_i^{\mathrm{el}} = \sum_{e\in\gamma} \tilde{\boldsymbol{f}}_{e,i}^{\mathrm{el}} + \hat{\boldsymbol{f}}_i \tag{A.15}$$

In order to include the self interaction, one has to take into account the core energy. This core energy density is usually assumed to depend on the core energy associated with the local character angle of the dislocation line. Also, it is possible again to express to volume integral as an integral over the line itself, which combined with the discretization of $\mathcal{C}$ leads to:

$$U^{\mathrm{core}}(\mathcal{C}, a) = \int \mathrm{d}U^{\mathrm{core}}(\boldsymbol{x}, a) d\boldsymbol{x} = \oint_{\mathcal{C}} W^{core}(\boldsymbol{\vartheta}(\boldsymbol{x}), a) dL(\boldsymbol{x}) \approx \sum_{e\in\gamma} W^{core}(\boldsymbol{\vartheta}_e, a) L_e \tag{A.16}$$

where L_e is the length of segment e. A precise functional structure of W^{core} is given in CADD-3d.II. The core energy contribution to the nodal force is consequently related to the length variation of the segments in γ:

$$\boldsymbol{f}_i^{\mathrm{core}} = \sum_{e\in\gamma} W^{core}(\boldsymbol{\vartheta}_e, a) \frac{\partial L_e}{\partial \boldsymbol{x_i}} = \sum_{e\in\gamma} \boldsymbol{f}_{e,i}^{\mathrm{core}} \tag{A.17}$$

Afterall we have the force acting on any dislocation node to be expressed as a sum of segment contributions

[4] Employing a quadrature rule will result in the same formulation.

with externally applied forces:

$$\forall i \in \mathcal{N} \quad \boldsymbol{f}_i = \sum_{e \in \gamma} \left[\boldsymbol{f}^{\text{core}}_{e,i} + \boldsymbol{f}^{\text{el}}_{e,i} \right] + \hat{\boldsymbol{f}}_i \tag{A.18}$$

A.3.3. Dislocation Mobility

The Peach-Koehler force field acts effectively on the line as a driving force. The dynamical evolution can be realized by integrating the configuration of the dislocation lines over time. In the over-damped regime, acceleration effects are neglected. Then the dislocation mobility is the relation existing between the dislocation velocity $\boldsymbol{v}$ and the Peach-Koehler force [7]:

$$\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{\mathcal{M}}_{\vartheta(\boldsymbol{x}),T}(\boldsymbol{f}^{\text{PK}}(\boldsymbol{x})) \tag{A.19}$$

where $\boldsymbol{\mathcal{M}}_{\vartheta,T}$ is a function that depends on the local character angle $\vartheta(\boldsymbol{x})$ and on the temperature T [22]. It accounts for the energy dissipated during over-damped dislocation motion. At moderate velocities (a fraction of the shear wave speed), the damping is due to phonon viscous forces which are mainly proportional to the temperature [26, 21, 18] leading to a linear mobility relation:

$$\boldsymbol{v}(\boldsymbol{x}) = \boldsymbol{\mathcal{B}}(\vartheta(\boldsymbol{x}), T)\boldsymbol{f}^{\text{PK}}(\boldsymbol{x}) \tag{A.20}$$

where $\boldsymbol{\mathcal{B}}$ is a tensor that depends on the dislocation character angle and temperature T. For convenience we will now drop the dependence on $\boldsymbol{x}$. In the case of FCC materials, it is classical to prevent climb motion[5] to account for the effect of Shockley partials. To that end the tensor $\boldsymbol{\mathcal{B}}$ can be defined as proportional to a projection onto the slip plane:

$$\boldsymbol{\mathcal{B}} = \Lambda_\vartheta \cdot (\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n}) \tag{A.21}$$

where $\boldsymbol{n}$ is the unit vector normal to the desired slip plane, and Λ_ϑ is a scalar. Then, by employing A.11 and A.21 into A.20 and rewriting thanks to cross product identities leads to:

$$\boldsymbol{v} = \Lambda_\vartheta \left[\boldsymbol{b} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{n} \right] \cdot (\boldsymbol{n} \times \boldsymbol{\xi}) \tag{A.22}$$

which shows that the infinitesimal dislocation velocity is in the direction $(\boldsymbol{n} \times \boldsymbol{\xi})$, which is normal to the dislocation line wihtin the slip plane. By defining the resolved shear stress as $\sigma_{res} = \frac{\boldsymbol{b} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{n}}{|\boldsymbol{b}|}$ we obtain the scalar mobility relation, which is used in the *ParaDis* code:

$$|\boldsymbol{v}| = \Lambda_\vartheta \sigma_{res} |\boldsymbol{b}| \tag{A.23}$$

Such a mobility law can be evaluated either from experiments, theory [14, 1, 23, 12, 16] or molecular dynamics simulations [25, 27, 29, 5, 11]. At moderate velocities, it was shown that Λ_ϑ was inversely proportional to temperature [29]). Furthermore, it is important to insist on the dependence of the mobility law with the dislocation character angle. Offline molecular dynamics calculation can extract a finite set of mobility-angles relations. Consequently, it is classical to interpolate Λ_ϑ in between the precomputed angles [11].

In the small scale cases considered in the companion paper III, it is important to acknowledge inertial effects, so that an effective mass has to be introduced:

$$\sigma_{res}|\boldsymbol{b}| = \mathcal{D}_\vartheta |\boldsymbol{v}| + m^d |\dot{\boldsymbol{v}^l}| \tag{A.24}$$

where m^d is an effective mass density (Kg/m) and where we have noted $\mathcal{D} = \Lambda^{-1}$. If one wants to limit motion when the resolved stress is below a Peierls stress, we obtain:

$$\left(\sigma_{res} - \sigma_{Peierls} \right) |\boldsymbol{b}| = \mathcal{D}_\vartheta |\boldsymbol{v}| + m^d |\dot{\boldsymbol{v}^l}| \tag{A.25}$$

[5] out of slip plane.

A.3.4. Linear-elastic prediction of the displacement field due to a dislocation loop

In order to enforce displacement boundary conditions we know that we have to provide the displacement $\tilde{\boldsymbol{u}}$ on the $\partial\Omega_{\mathrm{u}}$. It was defined as the displacement resulting for the presence of dislocations in an infinite space. The work of Hirth and Lothe[22] provided the expression of such a displacement for an isotropic material in the case a triangular dislocation loop connecting three points A, B and C

$$\tilde{\boldsymbol{u}}(\boldsymbol{x}) = \frac{\boldsymbol{b}\Xi(\boldsymbol{x})}{4\pi} - \frac{1-2\nu}{8\pi(1-\nu)}[\boldsymbol{f}_{AB}(\boldsymbol{x}) + \boldsymbol{f}_{BC}(\boldsymbol{x}) + \boldsymbol{f}_{CA}(\boldsymbol{x})] + \frac{1}{8\pi(1-\nu)}[\boldsymbol{g}_{AB}(\boldsymbol{x}) + \boldsymbol{g}_{BC} + \boldsymbol{g}_{CA}(\boldsymbol{x})] \quad \text{(A.26)}$$

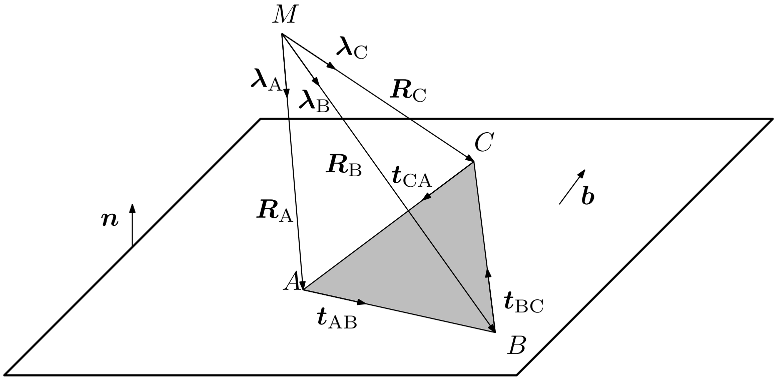

M
$\boldsymbol{\lambda}_{\mathrm{C}}$
$\boldsymbol{\lambda}_{\mathrm{A}}$
$\boldsymbol{\lambda}_{\mathrm{B}}$
$\boldsymbol{R}_{\mathrm{C}}$
C
$\boldsymbol{R}_{\mathrm{B}}$
$\boldsymbol{t}_{\mathrm{CA}}$
$\boldsymbol{b}$
$\boldsymbol{n}$
$\boldsymbol{R}_{\mathrm{A}}$
A
$\boldsymbol{t}_{\mathrm{BC}}$
$\boldsymbol{t}_{\mathrm{AB}}$
B

Figure A.9: Geometrical description of a triangular loop ABC lying on a glide plane normal to the vector $\boldsymbol{n}$ with a unique Burgers vector $\boldsymbol{b}$. The point M is where the displacement field has to be evaluated.

where Ξ is the solid angle formed by vectors $\boldsymbol{\lambda}_A, \boldsymbol{\lambda}_B, \boldsymbol{\lambda}_C$. Also $\boldsymbol{f}$ and $\boldsymbol{g}$ follow these expressions:

$$\boldsymbol{f}_{AB} = [\boldsymbol{b} \wedge \boldsymbol{t}_{AB}]\ln\left(\frac{\boldsymbol{R}_B}{\boldsymbol{R}_A} \cdot \frac{1+\boldsymbol{\lambda}_B \cdot \boldsymbol{t}_{AB}}{1+\boldsymbol{\lambda}_A \cdot \boldsymbol{t}_{AB}}\right)$$

$$\boldsymbol{g}_{AB} = [\boldsymbol{b} \cdot (\boldsymbol{\lambda}_A \wedge \boldsymbol{\lambda}_B)]\frac{\boldsymbol{\lambda}_A + \boldsymbol{\lambda}_B}{1+\boldsymbol{\lambda}_A \cdot \boldsymbol{\lambda}_B}$$

The meaning of the geometrical parameters $\boldsymbol{R}_A, \boldsymbol{R}_B, \boldsymbol{R}_C, \boldsymbol{\lambda}_A, \boldsymbol{\lambda}_B, \boldsymbol{\lambda}_C, \boldsymbol{t}_{AB}, \boldsymbol{t}_{BC}$ and $\boldsymbol{t}_{CA}$ are presented on Figure A.9. In the expression of $\tilde{\boldsymbol{u}}$, the solid angle Ω is the only term being discontinuous and will acknowledge the displacement jump through the glide plane of a FCC dislocation. Barnett and Baluffi [2, 3], proposed to decompose any dislocation line discretized by segments in γ into triangles as illustrated on Figure A.10. Based on this expression, Fivel and Depres [15] provide the algorithmic details of for the construction of the displacement fields for an arbitrary dislocation loop, including the case where several slip planes would be spanned by the same dislocation. This treatment allows to express displacement as a sum of contributions from segments which provides the following property.

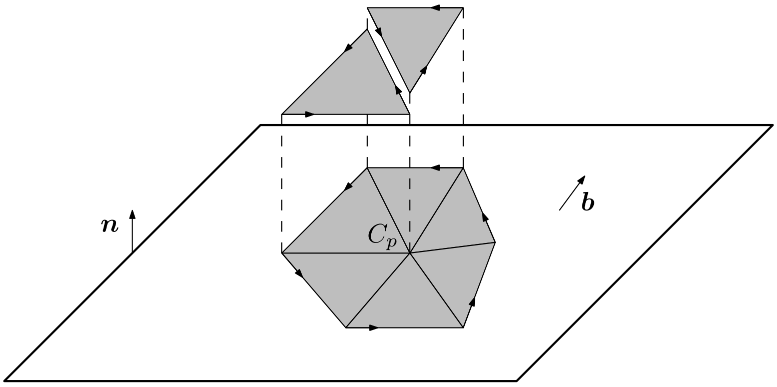


Figure A.10: Generic decomposition of a polygonal loop lying on a glide plane normal to the vector $\boldsymbol{n}$ with a unique Burgers vector $\boldsymbol{b}$. The principle is to choose a point C_p in the glide plane and to decompose the loop in triangles. The displacement produced by the superposition of the all the triangles is equivalent to that one produced by the polygonal loop.

Property 2. *Displacement $\tilde{\boldsymbol{u}}$ generated by a closed dislocation path. Let a dislocation path $\mathcal{C}$ be a closed loop discretized by the segments in γ. Then the displacement field created by $\mathcal{C}$ can be computed as a sum of contributions from each segment:*

$$\forall x \in \Omega \quad \tilde{\boldsymbol{u}}(\boldsymbol{x}) = \sum_{e \in \gamma} \tilde{\boldsymbol{u}}_e(\boldsymbol{x})$$

B. Error estimation for the core template

The data generated from a MD simulation can then be simplified: all atomic positions $\boldsymbol{x}_{0,i}$ are projected onto the (y, z) plane in order to obtain a two-dimensional set of points $\bar{\boldsymbol{x}}_{0,i}$

$$\bar{\boldsymbol{x}}_{0,i} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_{0,i}. \tag{B.1}$$

Let us consider the situation where the *core template* is used over a radius r_c. The total correction made can be expressed in cylindrical coordinates (the axis of the cylinder is the line direction):

$$\mathrm{Err}(r_\mathrm{c}) = \int_{x=0}^{L_x} \int_{\vartheta=0}^{2\pi} \int_{r=0}^{r_\mathrm{c}} \mid \Delta\tilde{\boldsymbol{u}}^{\mathrm{corr}}(r, x, \vartheta) \mid r \,\mathrm{d}r \,\mathrm{d}x \,\mathrm{d}\vartheta \tag{B.2}$$

where L_x is the size of the considered cylinder, such as the one presented in Figure 6(a). If we approximate the core template as a radial function:

$$\mathrm{Err}(r_\mathrm{c}) = 2\pi L_x \int_{r=0}^{r_\mathrm{c}} \mid \Delta\tilde{\boldsymbol{u}}^{\mathrm{corr}}(r) \mid r \,\mathrm{d}r \tag{B.3}$$

and by differentiation we obtain:

$$\mid \Delta\tilde{\boldsymbol{u}}^{\mathrm{corr}}(r_\mathrm{c}) \mid \approx \frac{1}{2\pi L_x r_\mathrm{c}} \frac{\partial \mathrm{Err}(r_\mathrm{c})}{\partial r_\mathrm{c}} \tag{B.4}$$

By extracting the error function numerically we can numerically estimate the dependency of the core template correction to the radius as presented on Figure B.11.

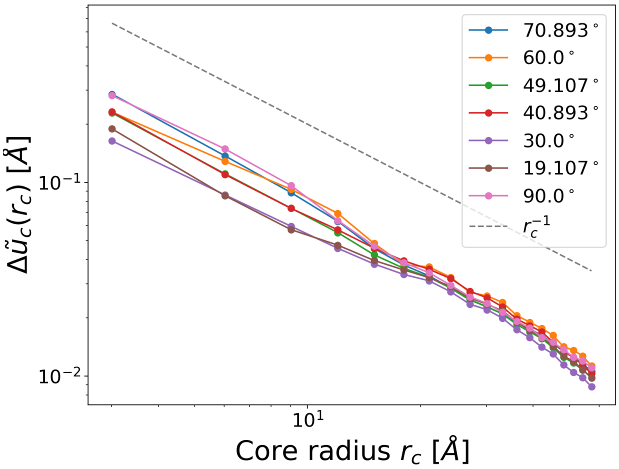


Figure B.11: Core template correction magnitude as a function of the distance to the dislocation line.

C. Dislocation Extraction Algorithm

The DXA requires the identification of reference lattice vectors corresponding to lattice vectors in the deformed configuration. The atomic reference positions Λ_{a} is assumed to be a subset of a Bravais lattice

$$\Lambda := \{\ a_1\,\boldsymbol{v}_1 + a_2\,\boldsymbol{v}_2 + a_3\,\boldsymbol{v}_3 \mid \forall\, i = 1, ..., 3\ a_i \in \mathbb{Z}\ \},$$

with basis vectors $\{\boldsymbol{v}_i\}_{i=1,...,3}$. For this purpose, we assume a tetrahedral tessellation (*i.e.* a three-dimensional Delaunay triangulation) of the reference lattice Λ_{a}. This definition is not unique, e.g. other types of elements could be used in almost the same manner. However, we found it most convenient to employ a tetrahedral tesselation from a computational point of view since efficient tools for are available off the shelf [6].

Since the reference lattice has translational symmetry there exists a finite set of distinct lattice vectors $\mathcal{X}$. We call elements $\boldsymbol{v} \in \mathcal{X}$ the *ideal lattice vectors*. Next we assume a tessellation $\mathcal{T}$ of the current configuration $\Lambda_{\mathrm{a}} + \boldsymbol{u}_{\mathrm{a}}$. We denote elements of $\mathcal{T}$, i.e. tetrahedra, faces and edges, by t, f and s. In order to identify "dislocated" tetrahedra, we seek to relate the edge vectors $\mathrm{d}\boldsymbol{y}_s$ in $\mathcal{T}$ to the ideal lattice vectors. This procedure is schematically depicted in Figure C.12(a) for a face-centered cubic (FCC) lattice. For the identification we use the criterion from [34] defined

$$\forall\ \mathrm{d}\boldsymbol{y}_s \qquad \mathrm{d}\boldsymbol{x}_s := \mathrm{Arg}\left\{ \min_{\boldsymbol{v}\in\mathcal{X}} \|\, \boldsymbol{v} - \mathrm{d}\boldsymbol{y}_s \,\| \right\} \tag{C.1}$$

which minimizes the Euclidean norm with respect to $\mathcal{X}$. We note that this procedure essentially detects full dislocations in the deformed lattice. However, it is also possible to detect partial dislocations by adding the ideal lattice vectors corresponding to the reference lattice structure in the stacking fault. This is shown schematically in Figure C.12(a) for the FCC lattice which incorporates a hexagonal closed-packed (HCP) structure in the stacking fault. Other criteria have been proposed, e.g. by comparing angles between lattice vectors as done by [20] but we have found (C.1) to be a very robust criterion if solely crystal slip is considered which suffices for our current purposes. However, we remark that the identification (C.1) is limited to single crystals and small lattice rotations. For the more general identification procedure we refer the reader to [36]. In the following we assume that each reference lattice vector is computed via the criterion (C.1). The true Burgers vector for each face $f \in \mathcal{T}$ can then be readily identified via

$$\boldsymbol{b}_f = \sum_{s\in f} \mathrm{d}\boldsymbol{x}_s. \tag{C.2}$$

Tetrahedra with $\boldsymbol{b}_f \neq \boldsymbol{0}$ for two or more faces are consequently defined as distorted tetrahedra (cf. Figure C.12(b)).

Algorithm 2 summarizes the relevant steps of our detection algorithm. In practice, we loop over all faces until we detect a non-zero Burgers vector. Subsequently we track the dislocation line through the crystal by repetitively identifying the next distorted neighbor of the previous tetrahedron. For each distorted tetrahedron we store its center of mass to a vector $\hat{p}$. We abort the tracking process if we arrive at the initial tetrahedron or at a boundary face. In order to generate the line representation from the point set $\hat{p}$ we have used a simple scheme by extracting only every N-th node into a reduced point set. A continuous line representation is then derived by connecting the elements of the reduced point set via piece-wise linear segments.

Acknowledgments

The authors thank the Fonds National Suisse for support of this work through the project 200021_140506 entitled "Coupled Atomistic/Discrete-Dislocations in 3d (CADD3d) and the European Research Council for the ERCstg UFO-240332.

[6] Our implementation of the DXA uses the *Computational Geometry Algorithms Library [30]*

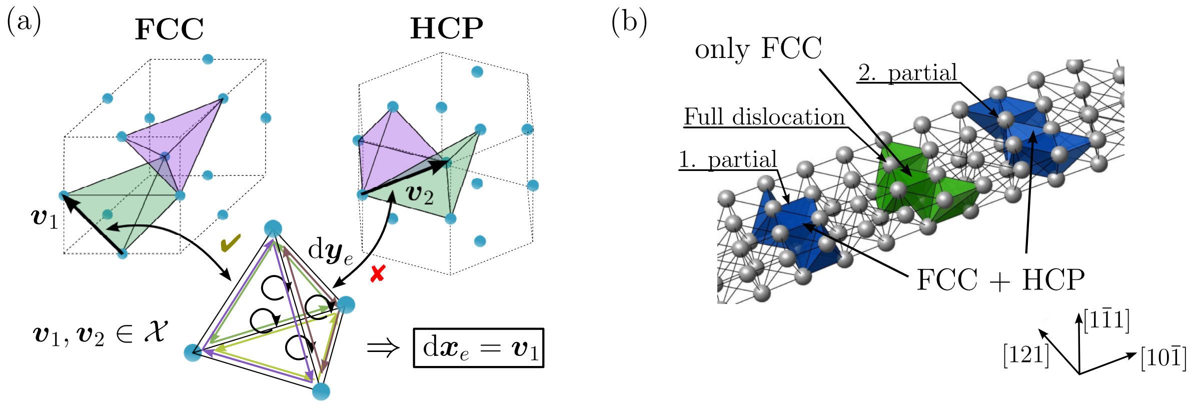

(a)
FCC
HCP
$\boldsymbol{v}_1$
$\boldsymbol{v}_2$
$\mathrm{d}\boldsymbol{y}_e$
✓
✗
$\boldsymbol{v}_1, \boldsymbol{v}_2 \in \mathcal{X}$
$\Rightarrow \boxed{\mathrm{d}\boldsymbol{x}_e = \boldsymbol{v}_1}$
(b)
only FCC
2. partial
Full dislocation
1. partial
FCC + HCP
[1$\bar{1}$1]
[121]
[10$\bar{1}$]

Figure C.12: Schematic illustration of the identification of reference lattice vectors corresponding to the lattice vectors in the current configuration.

Algorithm 2: Dislocation detection algorithm (**dislocDetn**)

```
Input: Deformed lattice $\Lambda_a + \boldsymbol{u}_a$
$\mathcal{X}$ ← define set of ideal lattice vectors;
$\mathcal{T}$ ← compute tetrahedralization of $\Lambda_a + \boldsymbol{u}_a$;
for all $e \in \mathcal{T}$ do
    $\mathrm{d}\boldsymbol{x}_s$ ← compute its reference lattice vector using the criterion (C.1);
end
$\mathcal{V}$ ← {};
for $f \in \mathcal{T}$ do
    $\boldsymbol{b}_f$ ← compute its Burgers vector via (C.2);
    if $\boldsymbol{b}_f \neq \boldsymbol{0}$ then
        $\hat{p}$ ← {};
        while dislocation line not closed do
            $\hat{p}$ ← add center of mass of $t_f$;
            $t_f$ ← identify distorted neighboring tetrahedra of $t_f$;
        end
        $\mathcal{V}$ ← add $\hat{p}$;
        eliminate distorted faces from $\mathcal{T}$;
    end
end
for all $\hat{p} \in \mathcal{V}$ do
    $\gamma_a$ ← compute a continuous line representation from the point set $\hat{p}$;
    $\Gamma_a$← map $\gamma_a$ to the reference configuration;
end
Output: Discrete atomistic dislocations $\gamma$
```

Bibliography

References

[1] Atkinson, W., Cabrera, N., 1965. Motion of a frenkel-kontorowa dislocation in a one-dimensional crystal. Phys. Rev. 138, A763–A766.

[2] Barnett, D., 1985. The displacement field of a triangular dislocation loop. Philosophical Magazine A 51, 383–387. arXiv:http://dx.doi.org/10.1080/01418618508237562.

[3] Barnett, D., Balluffi, R., 2007. The displacement field of a triangular dislocation loop - a correction with commentary. Philosophical Magazine Letters 87, 943–944. arXiv:http://dx.doi.org/10.1080/09500830701601748.

[4] Berendsen, H.J.C., 2007. Simulating the Physical World: Hierarchical Modeling from Quantum Mechanics to Fluid Dynamics. Cambridge University Press, New York, NY, USA.

[5] Bhate, N., Clifton, R., Phillips, R., 2002. Atomistic simulations of the motion of an edge dislocation in aluminum using the embedded atom method. AIP Conference Proceedings 620, 339–342.

[6] Born, M., Oppenheimer, R., 1927. Zur quantentheorie der molekeln. Annalen der Physik 84, 0457–0484.

[7] Bulatov, V., Cai, W., 2006. Computer Simulations of Dislocations (Oxford Series on Materials Modelling). Oxford University Press, USA.

[8] Cai, W., Arsenlis, A., Weinberger, C.R., Bulatov, V.V., 2006. A non-singular continuum theory of dislocations. Journal of the Mechanics and Physics of Solids 54, 561–587.

[9] Cai, W., Bulatov, V.V., Pierce, T.G., Hiratani, M., Rhee, M., Bartelt, M., Tang, M., 2004. Massively-parallel dislocation dynamics simulations, in: Kitagawa, H., Shibutani, Y. (Eds.), Iutam Symposium on Mesoscopic Dynamics of Fracture Process and Materials Strength, pp. 1–11.

[10] Cho, J., Junge, T., Molinari, J.F., Anciaux, G., 2015. Toward a 3d coupled atomistic and discrete dislocation dynamics simulation: dislocation core structures and Peierls stresses with several character angles in FCC aluminum. Advanced Modeling and Simulation in Engineering Sciences 2, 12.

[11] Cho, J., Molinari, J.F., Anciaux, G., 2017. Mobility law of dislocations with several character angles and temperatures in FCC aluminum. International Journal of Plasticity 90, 66–75.

[12] Ramos de Debiaggi, S., Caro, A., 1987. Phonons radiated by moving dislocations in disordered alloys. J. Phys. France 48, 1499–1504.

[13] Dewald, M., Curtin, W.A., 2006. Analysis and minimization of dislocation interactions with atomistic/continuum interfaces. Modelling and Simulation in Materials Science and Engineering 14, 497.

[14] Eshelby, J., 1956. Supersonic dislocations and dislocations in dispersive media. Proceedings of the Physical Society. Section B 69, 1013.

[15] Fivel, M., Depres, C., 2014. An easy implementation of displacement calculations in 3d discrete dislocation dynamics codes. Philosophical Magazine 94, 3206–3214.

[16] Flytzanis, N., Celli, V., Nobile, A., 1974. Motion of two screw dislocations in a lattice. Journal of Applied Physics 45, 5176–5181.

[17] van der Giessen, E., Needleman, A., 1995. Discrete dislocation plasticity - a simple planar model. Modelling and simulation in Materials Science and Engineering 3, 689–735.

[18] Gorman, J., Wood, D., Vreeland, T., 1969. Mobility of dislocations in aluminum. Journal of Applied Physics 40, 833–841.

[19] Griebel, M., Knapek, S., Zumbusch, G., 2007. Numerical Simulation in Molecular Dynamics: Numerics, Algorithms, Parallelization, Applications (Texts in Computational Science and Engineering). Springer.

[20] Hartley, C., Mishin, Y., 2005. Characterization and visualization of the lattice misfit associated with dislocation cores. Acta Materialia 53, 1313 – 1321.

[21] Hikata, A., Johnson, R., Elbaum, C., 1971. Interaction of dislocations with electrons and with phonons. Phys. Rev. B 4, 674–674.

[22] Hirth, J.P., Lothe, J., 1992. Theory of Dislocations. second ed., Krieger Publishing Company, Malabar, Florida, U.S.A.

[23] Ishioka, S., 1973. Steady motion of a dislocation in a lattice. Journal of the Physical Society of Japan 34, 462.

[24] Junge, T., Anciaux, G., Molinari, J.F., 2015. Dynamic stability of displacement-based atomistic/continuum coupling methods. Journal of the Mechanics and Physics of Solids 80, 103–120.

[25] Kang, K., Bulatov, V., Cai, W., 2012. Singular orientations and faceted motion of dislocations in body-centered cubic crystals. Proceedings of the National Academy of Sciences 109, 15174–15178. `arXiv:http://www.pnas.org/content/109/38/15174.full.pdf`.

[26] Leibfried, G., 1950. Über den einfluß thermisch angeregter schallwellen auf die plastische deformation. Zeitschrift für Physik 127, 344–356.

[27] Martínez, E., Marian, J., Arsenlis, A., Victoria, M., Perlado, J., 2008. Atomistically informed dislocation dynamics in FCC crystals. Journal of the Mechanics and Physics of Solids 56, 869 – 895.

[28] Mendelev, M.I., Asta, M., Rahman, M.J., Hoyt, J.J., 2009. Development of interatomic potentials appropriate for simulation of solid-liquid interface properties in al-mg alloys. Philosophical Magazine 89, 3269–3285.

[29] Olmsted, D., Louis, G., Curtin, W., Clifton, R., 2005. Atomistic simulation of dislocation mobility in Al,Ni and Al/Mg alloys. Modeling And Simulation in Materials Science and Engineering 13, 371–388.

[30] Project, T.C., 2017. CGAL User and Reference Manual. 4.9.1 ed., CGAL Editorial Board.

[31] Rapaport, D.C., 2004. The Art of Molecular Dynamics Simulation. Cambridge University Press.

[32] Shenoy, V.B., Miller, R., Tadmor, E.b., Rodney, D., Phillips, R., Ortiz, M., 1999. An adaptive finite element approach to atomic-scale mechanics—the quasicontinuum method. Journal of the Mechanics and Physics of Solids 47, 611–642.

[33] Shilkrot, L.E., Miller, R.E., Curtin, W.A., 2004. Multiscale plasticity modeling: coupled atomistics and discrete dislocation mechanics. Journal of the Mechanics and Physics of Solids 52, 755–787.

[34] Stukowski, A., 2014. A triangulation-based method to identify dislocations in atomistic models. Journal of the Mechanics and Physics of Solids 70, 314 – 319.

[35] Stukowski, A., Albe, K., 2010. Dislocation detection algorithm for atomistic simulations. Modelling and Simulation in Materials Science and Engineering 18, 025016.

[36] Stukowski, A., Bulatov, V.V., Arsenlis, A., 2012. Automated identification and indexing of dislocations in crystal interfaces. Modelling and Simulation in Materials Science and Engineering 20, 085007.

[37] Wallin, M., Curtin, W.A., Ristinmaa, M., Needleman, A., 2008. Multi-scale plasticity modeling: Coupled discrete dislocation and continuum crystal plasticity. Journal of the Mechanics and Physics of Solids 56, 3167–3180.

[38] Weygand, D., Friedman, L.H., Van der Giessen, E., Needleman, A., 2002. Aspects of boundary-value problem solutions with three-dimensional dislocation dynamics. Modelling and Simulation in Materials Science and Engineering 10, 437–468.

[39] Xiong, L., Deng, Q., Tucker, G., McDowell, D.L., Chen, Y., 2012a. A concurrent scheme for passing dislocations from atomistic to continuum domains. Acta Materialia 60, 899–913.

[40] Xiong, L., Deng, Q., Tucker, G.J., McDowell, D.L., Chen, Y., 2012b. Coarse-grained atomistic simulations of dislocations in Al, Ni and Cu crystals. International Journal of Plasticity 38, 86–101.

Acronyms

DDD	discrete dislocation dynamics
MD	molecular dynamics
FEM	finite-element method

CADD-3d glossary

$\varOmega_{\mathrm{A}}$	Molecular dynamics region
$\varOmega_{\mathrm{P}}$	Boundary condition of the the Molecular dynamics region. This is also referred as the pad region in the literature
$\partial\varOmega_{\mathrm{I}}$	Atomic region providing the boundary condition to the linear elastic model. This region is often referred as the interface
$\varOmega_{\mathrm{C}}$	Continuum region. It is the superposition of Dislocation dynamics and linear elastic regions
$\partial\varOmega_{\mathrm{u}}$	Exterior boundary condition of the Continuum region where displacements are prescribed
$\partial\varOmega_{\mathrm{t}}$	Exterior boundary condition of the Continuum region where tractions are prescribed
$\varDelta\tilde{\boldsymbol{u}}^{\mathrm{corr}}$	Core template correction field
$\varOmega^{\mathrm{core}}$	Region defining the core of one or several dislocations

Atomistic glossary

$\varLambda_{\mathrm{a}}$	Bravais lattice of atomistic domain
$\boldsymbol{f}_{\mathrm{a}}$	force on atom a
$\boldsymbol{v}_{\mathrm{a}}$	velocity of atom a
$\boldsymbol{r}_{\mathrm{a}}$	position of atom a
$\boldsymbol{u}_{\mathrm{a}}$	displacement of atom a
U^{a}	Total Potential energy of atoms

Dislocation dynamics glossary

$\tilde{\bullet}$	Field computed with infinite boundary conditions following [38]
$\hat{\bullet}$	Field computed from an elastic boundary value problem following [38]
R^{core}	Core Radius which is characterizing the non linear region around the dislocation line
U^{core}	Core energy, defined as the lineic energy density for a disc of radius R^{core} centered on the dislocation line
U^{el}	Strain energy due to a dislocation
γ	Set of dislocation segments

γ_{AD}	Dislocation segments located in the detection zone within the molecular dynamics region
γ^{c}	Dislocation segments with at least one end located in the continuum region
γ^{v}	Virtual dislocation segments employed to close the loops on $\partial\mathit{\Omega}_{\mathrm{I}}$
$\hat{\boldsymbol{t}}$	Corrected boundary traction for elasticity boundary value problem
$\tilde{\boldsymbol{u}}$	Displacement field due to the presence of dislocations within infinite boundary conditions
$\tilde{\varepsilon}$	Infinitesimal strain field due to the presence of dislocations within infinite boundary conditions
$\tilde{\boldsymbol{\sigma}}$	Stress tensor field due to the presence of dislocations within infinite boundary conditions
$\mathcal{N}$	Dislocation dynamics nodes
$\mathcal{N}^{\mathrm{DD}}$	Dislocation dynamics nodes associated to segments in γ^{c}
$\boldsymbol{f}_{\mathrm{d}}$	force on dislocation node d
$\boldsymbol{s}_{\mathrm{e}}$	dislocation segment
$\boldsymbol{f}^{\mathrm{PK}}$	Peach Koehler force
$\mathcal{C}$	Path describing the dislocation lines
$\boldsymbol{\xi}$	Normalized vector along the dislocation line
$\boldsymbol{b}$	Burgers vector
$\boldsymbol{\mathcal{B}}$	Characteristic tensor for the linear mobility law
$\mathit{\Lambda}$	Characteristic scalar in the FCC linear mobility law
$\boldsymbol{\mathcal{M}}$	Mobility function
m^d	Dislocation effective mass employed in the modified mobility law
σ_{res}	Resolved shear stress
$\sigma_{Peierls}$	Peierls stress

Continuum elasticity

$\hat{\boldsymbol{u}}$	Dislocation free displacement field
$\boldsymbol{u}$	Displacement field
$\boldsymbol{t}$	Boundary traction
ε	Infinitesimal strain field
$\boldsymbol{\sigma}$	Cauchy stress field
$\hat{\varepsilon}$	Dislocation free infinitesimal strain field
$\hat{\boldsymbol{\sigma}}$	Dislocation free stress tensor field
$\boldsymbol{u}_{\mathrm{n}}$	displacement of finite-element node n
U^c	Strain energy of the continuum