Chapter 2
Fundamentals of Dislocation Dynamics Simulations

Ryan B. Sills, William P. Kuykendall, Amin Aghaei, and Wei Cai

2.1 Overview

When crystalline solids undergo plastic deformation, line defects known as dislocations move, multiply, and react with one another. The overall mechanical properties of the crystal in this plastic regime are governed by these dislocation processes. Dislocation dynamics (DD) is a modeling approach that aims to simulate the motion and interaction of these dislocation lines to gain insights concerning the mechanical properties of the material.

Dislocation lines are defects whose core widths are at the scale of the crystal lattice. The length scale over which dislocation structures evolve is, however, many orders of magnitude larger than the interatomic distance. A classical example is the formation of dislocation cells; at moderate to large amounts of plastic deformation, dislocation networks are known to form cellular structures, with an average cell size on the order of 1 μm (see Fig. 2.1a). Hence, any model which hopes to inform our understanding of bulk plastic deformation—for example, understanding the temperature dependence of the stress–strain curves shown in Fig. 2.1b—must

R.B. Sills (✉)
Sandia National Laboratories, Livermore, CA 94550, USA

Stanford University, Stanford, CA 94305, USA
e-mail: rbsills@sandia.gov

W.P. Kuykendall
Stanford University, Stanford, CA 94305, USA

Sandia National Laboratories, Livermore, CA 94550, USA
e-mail: wpkuyken@sandia.gov

A. Aghaei • W. Cai
Stanford University, Stanford, CA 94305, USA
e-mail: am.aghaei@gmail.com; caiwei@stanford.edu

C.R. Weinberger, G.J. Tucker (eds.), *Multiscale Materials Modeling for Nanomechanics*, Springer Series in Materials Science 245,
DOI 10.1007/978-3-319-33480-6_2

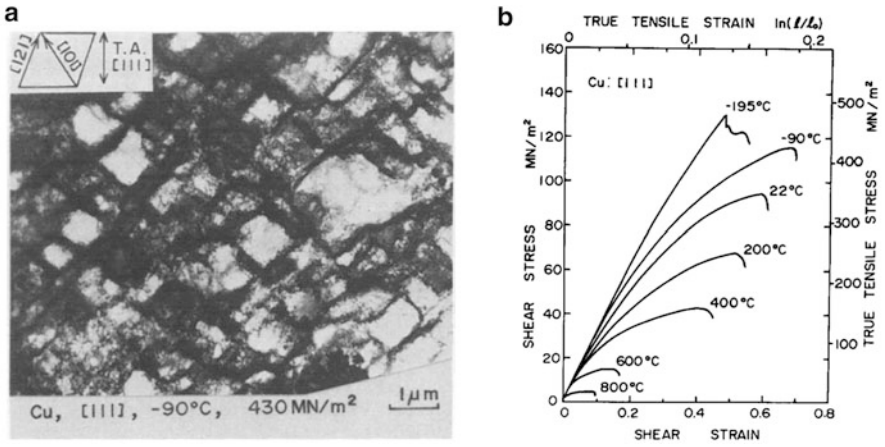

Fig. 2.1 Examples of dislocation plasticity. (**a**) Cellular dislocation structure in single crystal copper after tensile loading in the [1 1 1] direction at −90°C and (**b**) stress–strain curves for single crystal copper at various temperatures. Reproduced from [48] with permission from Elsevier

simulate a material volume above this length scale. Using an atomistic approach would require the simulation of more than 10^{10} atoms, a simulation size which is prohibitively expensive even for the most modern computational tools. This gap in scale necessitates a new model at the so-called mesoscale: dislocation dynamics.

The idea behind the DD approach is that because plastic deformation is dominated by the motion and interaction of dislocation lines, one only needs to consider the dislocation lines, rather than the locations of all of the atoms, to understand the plastic behavior of a material. Taking such an approach enables simulations with length scales of 10 μm and time scales of 1 ms. As with all mesoscale approaches, DD requires the input of multiple physical models to describe the various behaviors of the dislocation lines, meaning that much information must be provided either from experiments or more fundamental models. Unlike other mesoscale models of plasticity which consider the dislocation density in terms of a homogenized field, in DD dislocation lines are treated explicitly so that individual dislocation–dislocation interactions can be properly captured.

Much of the theory that feeds into the models that describe the dislocation lines has been established for many decades, as has the concept of DD itself [34, 53]. However, only recently have large-scale simulations been made possible with the inception of modern computational tools. Despite these many advances, DD remains a challenging tool to use, often requiring hundreds of computer cores for a single simulation of a short duration of physical time relative to experiments.

The remainder of the chapter will be organized as follows. First, in Sect. 2.2 we will discuss the basic features of the DD formulation. In Sect. 2.3, we will then discuss how to run a DD simulation all the way from inputs to outputs, and show a few examples. Section 2.4 will discuss DD's place in the hierarchy of material models. Finally, Sect. 2.5 will present topics of current research and challenges that the DD community need to overcome to enable more widespread use of the tool.

2.2 Fundamentals

In order to simulate the motion and interaction of dislocation lines, a number of algorithms, rules, and procedures have been developed. In this section, we break these features into two groups. First, we discuss the most basic ingredients necessary to conduct a DD simulation: how driving forces are exerted on dislocations (2.2.2.1), how to determine dislocation velocities given these forces (2.2.2.2), discretization and adaptive remeshing of the dislocation lines (2.2.2.3), time integration of the equations of motion (2.2.2.4), and how dislocations can collide and react (2.2.2.5). We will then introduce more advanced aspects of DD simulations: how to handle dislocation junctions and intersections (2.2.3.1), different types of boundary conditions (2.2.3.2), how screw dislocations can change their glide plane through cross-slip (2.2.3.3), and a brief discussion of two-dimensional DD simulations (2.2.3.4). These features are presented in the flowchart shown in Fig. 2.3. We begin, however, with a discussion of the overall problem formulation.

2.2.1 Problem Formulation

The basic idea behind DD is to embed the physics of dislocations into a set of governing equations that can be solved for the positions of a network of dislocation lines, given an initial dislocation configuration, boundary conditions, and loading conditions. The positions of the lines are described by the vector $\mathbf{r}(s,t)$, where s is a scalar parameter dictating the location along the lines, as shown in Fig. 2.2a, and t denotes time. Because we seek a tool that can obtain a solution in arbitrary settings (e.g., many dislocation lines loaded multiaxially), we will need to discretize our system in both space and time, and employ numerical methods to solve the governing equations. Figure 2.2b shows an example of discretization in space.

As we will discuss, many things can exert forces on dislocations. These forces can be broken into drag forces, which resist dislocation motion, and driving forces, which promote it. Additionally, dislocation lines are known to have effective masses, giving rise to inertial forces [51]. In many crystalline materials under a broad range of conditions, however, drag forces intrinsic to the crystal lattice are orders of magnitude larger than the inertial forces, making dislocation motion overdamped [51]. This means that in the overall equations of motion, we can neglect inertial terms altogether, and simply require that the total driving force balance the total drag force, i.e.,

$$\sum \mathbf{F}_{\text{drag}}(\mathbf{v},s) + \sum \mathbf{F}_{\text{drive}}(s) = 0 \tag{2.1}$$

where $\mathbf{v}$ is the dislocation velocity

$$\mathbf{v} = \frac{\partial \mathbf{r}(s,t)}{\partial t} \tag{2.2}$$

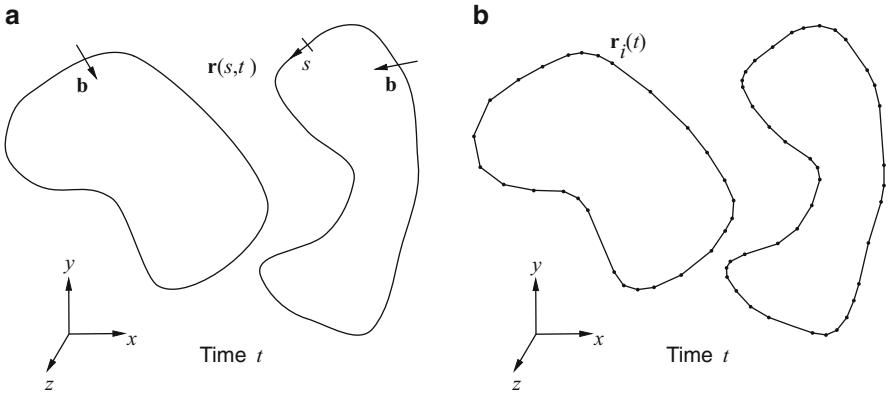

Fig. 2.2 Position of a pair of dislocation loops at time t. (**a**) Continuous representation described by position vector $\mathbf{r}$ as a function of parameter s. (**b**) Discrete representation using node-based discretization (see Sect. 2.2.2.3) described by position vectors of nodes $\mathbf{r}_i$

and the summations are over all drag and driving force contributions. Usually, in dislocation dynamics, Eq. (2.1) can be explicitly solved for $\mathbf{v}$ and restated as

$$\mathbf{v} = \mathbf{M}(\mathbf{F}^{\text{tot}}_{\text{drive}}) \tag{2.3}$$

where $\mathbf{F}^{\text{tot}}_{\text{drive}}[\mathbf{r}(s), \boldsymbol{\sigma}_{\text{ext}}, \ldots] = \sum \mathbf{F}_{\text{drive}}$ is the total driving force as a function of parameter s, dependent upon the dislocation position $\mathbf{r}(s)$, the externally applied stress $\boldsymbol{\sigma}_{\text{ext}}$, and any other features which exert driving forces. The function $\mathbf{M}(\cdot)$, which provides the velocity given a total driving force, is called the *mobility law*. The final governing equation of motion can be written as

$$\frac{\partial \mathbf{r}(s,t)}{\partial t} = \mathbf{g}\left[\mathbf{r}(s), \boldsymbol{\sigma}_{\text{ext}}, \ldots\right], \tag{2.4}$$

where $\mathbf{g} \equiv \mathbf{M}\left(\mathbf{F}^{\text{tot}}_{\text{drive}}\left[\mathbf{r}(s), \boldsymbol{\sigma}_{\text{ext}}, \ldots\right]\right)$ is an operator which computes the velocity $\mathbf{v}(s)$ from a given dislocation structure $\mathbf{r}(s)$ and loading condition.

2.2.2 *Basic Features*

A flowchart depicting the major steps in a DD simulation is presented in Fig. 2.3. We will now discuss each of these steps in turn.

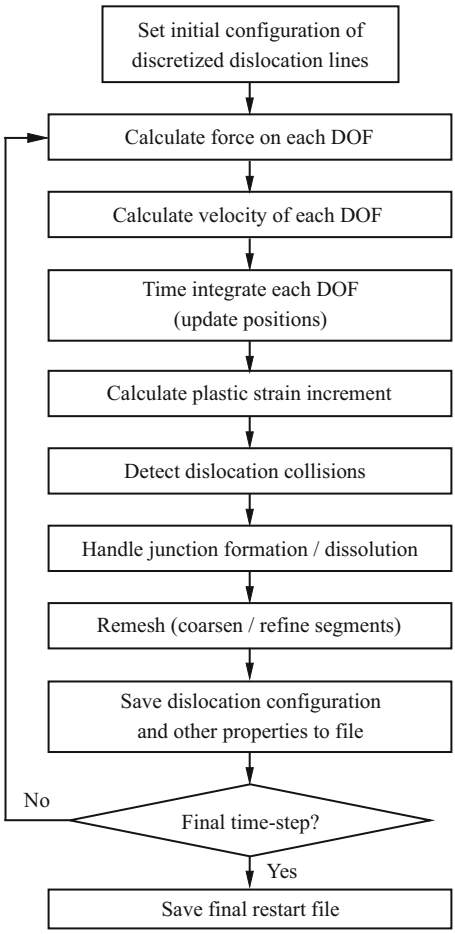

Fig. 2.3 Flowchart showing the basic steps for a dislocation dynamics code. Note that the force and velocity of each degree of freedom (DOF) may be computed multiple times per time step depending on the time integration scheme used (see Sect. 2.2.2.4)

2.2.2.1 Driving Forces

Many features of crystalline solids can apply driving forces to dislocation lines. These forces can be divided into two categories: forces arising from local stress fields (Peach–Koehler forces) and forces due to dislocation self interaction.

To determine the driving force exerted on a dislocation by a stress field $\boldsymbol{\sigma}(s)$ applied at position s, the Peach–Koehler expression is commonly used. It gives that the force per unit length, $\mathbf{F}(s)$, is [42]

$$\mathbf{F}(s) = (\boldsymbol{\sigma}(s) \cdot \mathbf{b}) \times \boldsymbol{\xi}(s), \tag{2.5}$$

where $\mathbf{b}$ is the Burgers vector of the dislocation and $\boldsymbol{\xi}(s)$ is the direction of the dislocation line at s (which varies with position for curved dislocations). Hence, any feature of a crystalline solid that results in a stress field can exert forces on

dislocations. The most common sources of stress in dislocation dynamics simulations are applied stresses due to loading of the simulation cell and stresses from other dislocations, which decay as $1/r$, where r is the distance from the dislocation [1]. The latter means that to determine the total force on a dislocation segment, we must consider the force exerted by every other segment in the simulation cell. This gives rise to an $\mathcal{O}(N^2)$ computation if all pair-wise interactions are computed explicitly (N is the number of segments), and makes DD difficult to implement efficiently. Other possible origins of stress include solute atoms, precipitates or inclusions, and free surfaces or secondary phase boundaries. The material system of interest will decide which of these must be considered.

With nanomaterials, free surface effects are especially important, since the small specimen size means every dislocation is near a free surface. For this reason, we will briefly discuss the nature of forces generated by free surfaces. In elasticity theory, for simplicity, stress expressions for dislocations are usually derived in a homogeneous, infinite medium. When these expressions are then used in finite media, they result in nonzero traction forces at the surfaces, violating the traction-free boundary condition at the surfaces. To correct this, a set of so-called image tractions must be applied to the surface. These image tractions render the surface traction-free, but additionally produce their own *image stress field*, which can also exert forces on dislocations. Thus, the problem of a finite solid requires that the image stresses be determined for the given geometry and distribution of dislocations; this generally has to be done numerically, and we defer further discussion of image solvers (which compute the image field) to Sect. 2.2.3.2. We will discuss an example with a cylindrical specimen in Sect. 2.3.5.

The above discussion applies to forces arising externally from the dislocation line. In addition to these effects, the dislocation line can exert a force on itself. This self-force can be thought of as resulting from the energy of the dislocation line, and has two contributions. The first contribution is elastic, and can be computed using a number of approaches, such as the non-singular theory of dislocations [15]. The second contribution is due to nonlinear interatomic interactions at the dislocation core, and we shall refer to it as the *core force*. Core forces can influence the dislocation line in two ways. First, the core force will try to reduce the length of the dislocation line, since the total core energy scales with the line length. Second, because the core energy varies with line character (i.e., edge and screw dislocations have different energies), the core force will exert a torque on the line, trying to rotate it into its orientation of lowest energy. One approach for determining the core force is to derive it from the core energy. The core energy per unit length, E_c, of a dislocation line can be calculated using atomistic or first-principles methods as a function of the character angle θ (the angle between the Burgers vector and line direction). Alternatively, it is common in DD simulations to use an approximate analytical model to describe the core energy. For example, in the deWit and Koehler model [25] the core energy varies as

$$E_c(\theta) = \mathscr{E} b^2 \left(\frac{1}{1-\nu} \sin^2 \theta + \cos^2 \theta \right) \tag{2.6}$$

where ν is Poisson's ratio, b is the magnitude of the Burgers vector, and $\mathscr{E}$ is a parameter that controls the magnitude of the core energy; this is the same way the line energy varies according to elasticity theory for an isotropic solid. Often $\mathscr{E}$ is approximated as $\mathscr{E} \approx \alpha\mu$, where μ is the shear modulus and α is a material parameter in the range $0.1 - 0.5$ [45]. Given this function $E_c(\theta)$, the core force can be determined using a number of approaches. Our preference is to calculate the core force after the dislocation lines have been discretized, and hence we postpone further discussion of core forces until Sect. 2.2.2.3.

2.2.2.2 Mobility Laws

As we discussed in Sect. 2.2.1, mobility laws serve as constitutive equations in dislocation dynamics simulations, relating the total driving force per unit length acting on a dislocation line to its velocity. Since the movement of a dislocation is strongly material dependent, mobility laws must be constructed with a specific material system in mind [9, 14]. The mobility of a dislocation line is commonly dependent upon the dislocation character, direction of motion, the crystallographic plane on which the dislocation can move conservatively—known as the *glide plane*—and the temperature. The goal of a mobility law is to express these dependencies in terms of an explicit function for the velocity given a total driving force per unit length. Usually, this means determining the drag force exerted by the crystal lattice on a dislocation. In this section we explain how mobility laws can be obtained, and provide an example of a mobility law for face-centered cubic (FCC) crystals.

Linear mobility laws are commonly used. The viscous drag forces experienced by dislocations in crystalline solids, due, for instance, to phonon dispersion, are often proportional to the dislocation velocity [45]. Hence, a linear mobility model can be written as

$$\mathbf{M}(\mathbf{F}^{\text{tot}}_{\text{drive}}) = \mathscr{B}^{-1}(s) \cdot \mathbf{F}^{\text{tot}}_{\text{drive}}(s), \tag{2.7}$$

where $\mathscr{B}(s)$ is a drag coefficient tensor (with dimensions [mass]/([length][time])) and is strongly material dependent. The components of $\mathscr{B}(s)$ account for the various features affecting the dislocation drag coefficient. If more than one mechanism exerts linear drag on a dislocation, the net drag coefficient is the sum of the drag coefficients for each mechanism.

As an example of a linear mobility law, we consider the case of FCC crystals (using the same model as [12]). Excluding the possibility of cross-slip (to be discussed separately in Sect. 2.2.3.3), dislocations in FCC metals are confined to glide on $\{1\,1\,1\}$ planes; climb motion out of the glide plane requires the diffusion of vacancies into or out of the core, and is generally negligible at temperatures less than one-third of the melting point [42]. This glide confinement is a reflection of the dissociated core structure in FCC metals. The glide constraint can be enforced by setting the components of $\mathscr{B}$ coupled to out-of-plane motion to very large values.

This can lead to an ill-conditioned system, however, and it is numerically easier to project out climb motion by simply zeroing the velocity components in the direction of the glide plane normal; we will represent motion within the glide plane with the superscript ^{g}. Additionally, we often find with FCC metals that the drag coefficient is isotropic with respect to dislocation character (screw versus edge). Therefore, we can write the FCC mobility law as

$$\mathbf{v} = \mathbf{v}^g = \frac{\mathbf{F}^g}{B}, \tag{2.8}$$

where B is the isotropic drag coefficient and is typically between 10^{-5} and 10^{-4} Pa s for FCC metals [52]. With other materials, such as body-centered cubic (BCC) crystals, the drag coefficient is not isotropic and the glide constraint is not as strictly obeyed (for screw or near-screw dislocations), so that $\mathscr{B}$ will have to take a more complex form [12].

In many settings, a linear mobility law is inappropriate. For example, at low-to-moderate temperatures with BCC metals, the motion of screw dislocations is a thermally activated process; it occurs by the formation and movement of so-called kink pairs in the dislocation line. In this case, thermal activation theory should be used [52], which generally leads to a nonlinear mobility law. Nonlinear mobility laws have also been proposed to incorporate material effects besides lattice friction. For instance, solute atoms are known to exert drag forces on dislocations. A number of researchers have proposed nonlinear mobility laws that incorporate these effects [62, 95], and DD simulations have been conducted by approximating solute drag as a constant "back stress" which is subtracted from the driving force [69] (i.e., a ramp function mobility law).

2.2.2.3 Line Discretization and Remeshing

To employ numerical methods, we need to discretize the dislocation lines so that the overall dislocation structure is characterized by a set of nodes (or segments) and a data structure defining the connectivity between them. Discretization allows us to focus on a finite number of degrees of freedom (DOF), rather than an infinite number of points along the dislocation lines. Since dislocation lines can change their shape significantly during a simulation, and the total length of dislocation lines often increases, we also need to implement remeshing algorithms to modify the discretization when necessary. Dislocation lines can be discretized in a number of ways. Across the major DD codes, there are two general approaches to line discretization: lattice-based discretization and node-based discretization. Here we will discuss both. Major features of the two approaches are shown in Fig. 2.4.

In the lattice-based approach (used in the codes microMegas [24] and TRIDIS [103]), a grid of computational points, i.e., a lattice, often with a simple cubic structure of spacing a, is predefined throughout the simulation cell. Based on the structure of this lattice, a finite set of dislocation orientations is then selected

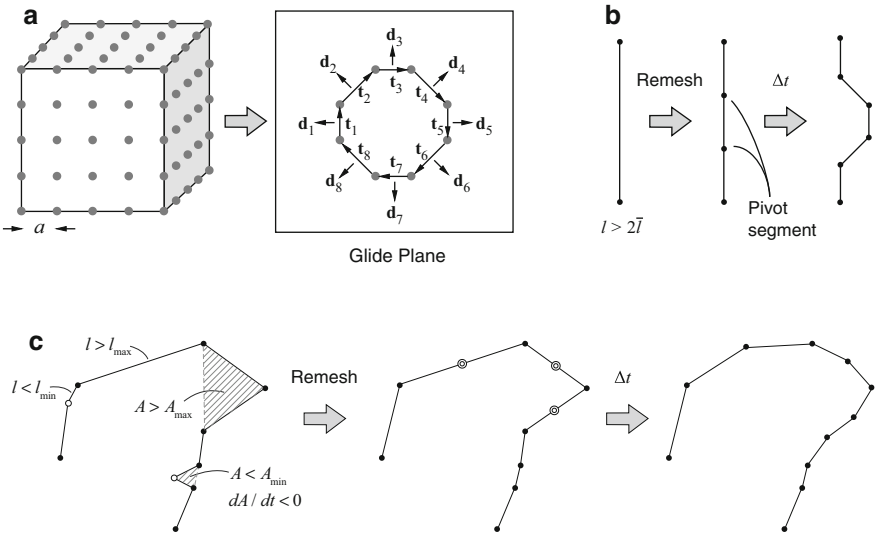

Fig. 2.4 Schematic depictions of (**a**,**b**) lattice-based and (**c**) node-based discretization. (**a**) The lattice grid used to define the segment directions $\mathbf{t}_i$ and movement directions $\mathbf{d}_i$. (**b**) Remeshing when a segment exceeds twice the average length $\bar{l}$, and response of pivot nodes after a time step Δt is taken. (**c**) Nodes are inserted (bullseye nodes) when $l > l_{\max}$ or $A > A_{\max}$, and removed (unfilled circles) when $l < l_{\min}$, or $A < A_{\min}$ with the area shrinking ($dA/dt < 0$)

and only these orientations are considered, as shown in Fig. 2.4a. These orientations define a unique set of straight line segments used to represent the dislocation lines (denoted as $\mathbf{t}_i$ in Fig. 2.4a). Dislocation motion is only considered in the direction orthogonal to each of these orientations (denoted as $\mathbf{d}_i$ in Fig. 2.4a). In this way, the segments are the degrees of freedom of the model. As the dislocation structure evolves, two different configurations of the dislocation lines are considered. The actual configuration is stored as the segments move continuously through space. When computing interaction forces and considering dislocation junctions, however, the actual configuration is projected onto the nearest set of lattice grid points in order to simplify the computations. Remeshing proceeds by dividing segments into smaller segments connected by "pivot segments" based on the user-specified average segment length, $\bar{l}$. The pivot segments initially have zero-length and extend along the direction set by the motion of their neighbors, allowing segments of new orientations to form, as depicted in Fig. 2.4b. In the lattice-based approach, the fidelity of the discretization is controlled by the spacing of the lattice grid, a, the number of line orientations allowed, and by the specified average dislocation segment length, $\bar{l}$.

With the node-based approach, dislocation lines are discretized according to a set of nodes and shape functions that connect the nodes, with the simplest case being linear shape functions that result in straight line segments. In this approach, any dislocation orientation is allowed and dislocation segments can move

in any direction (consistent with their mobility law). In contrast to lattice-based discretization, in the node-based approach the nodes are the fundamental degrees of freedom. Only a single dislocation configuration is considered at a given time; the same configuration which is evolved in time is used for force calculations. Node-based codes have been written using linear segments (MDDP [64], NUMODIS [73], ParaDiS [5], PARANOID [89]) and cubic splines (PDD [35]) to connect the nodes. Given the greater versatility of the node-based approach, a larger set of remesh rules must be specified. For example, in ParaDiS two criteria are used for remeshing: segment lengths and the area enclosed by adjacent segments [9]. Both minimum ($l_{\min}$, $A_{\min}$) and maximum ($l_{\max}$, $A_{\max}$) values are specified for each, and nodes are added or removed to bring the dislocation structure into compliance with these ranges (see Fig. 2.4c).

In order to evolve the dislocation structure, we need to compute the forces acting on the segments or nodes. Generally, the forces per unit length discussed in Sect. 2.2.2.1 vary with position along the lines. To get the total force acting on node or segment i, we need to integrate the force along the line. In this respect, lattice-based and node-based discretization differ slightly. With lattice-based models, since the segments are the fundamental degrees of freedom, we need to calculate the total force acting on a segment with the line integral

$$\mathbf{f}_i = \int_{C_i} \mathbf{F}(s) dL(s) \tag{2.9}$$

where C_i denotes segment i. Note that a lower case $\mathbf{f}$ denotes a force, and an upper case $\mathbf{F}$ denotes a force per unit length. Node-based codes, on the other hand, require the total force acting on the nodes. This is determined in terms of the line shape function $N_i^j(s)$ which describes the contribution to node i from segment j as

$$\mathbf{f}_i^j = \int_{C_j} N_i^j(s) \mathbf{F}(s) dL(s). \tag{2.10}$$

For example, with linear segment j connecting nodes i and k, $N_i^j(s) = s$ where $s = 0$ at node k and $s = 1$ at node i. The total force on node i is then the sum of the contributions from each of the segments it is attached to:

$$\mathbf{f}_i = \sum_j \mathbf{f}_i^j. \tag{2.11}$$

These expressions are valid if the force per unit length acting along the line is known. However, in the case of the core force, determining the force per unit length is not very straightforward. Instead, it is easier to derive the force acting on a segment or node directly from the core energy per unit length expression, $E_c(\theta)$ [9]. Given $E_c(\theta)$, we can compute the total core energy E_{core} for a given discretized dislocation structure by summing the contribution from each segment, and then find

the corresponding nodal or segment forces with

$$\mathbf{f}_i = -\frac{\partial E_{\text{core}}}{\partial \mathbf{r}_i}. \tag{2.12}$$

where $\mathbf{r}_i$ is the position vector of node or segment j.

Summing the Peach–Koehler and self-force contributions gives us the total force acting on a node or segment. However, mobility laws are usually written in terms of the force per unit length acting on the line. The force per unit length needed to evaluate the mobility law can be determined with

$$\mathbf{F}_i = \frac{\mathbf{f}_i}{\mathscr{L}_i}, \tag{2.13}$$

where $\mathscr{L}_i$ is a line length that depends on the discretization method. For the lattice-based approach, $\mathscr{L}_i$ is simply the length of segment i, $\mathscr{L}_i = l_i$. With node-based discretization, the following approximation[1] is commonly used: $\mathscr{L}_i = \sum_k l_{ik}/2$, where l_{ik} is the length of the segment connecting nodes i and k and the summation is over all nodes k connected to node i.

Now we have discretized the dislocation structure, and discussed the calculation of driving forces and subsequent velocity determination through the mobility law. Next we need to focus on evolving the positions of the nodes or segments, and the underlying dislocation structure they represent, in time.

2.2.2.4 Time Integration

As shown in Sect. 2.2.1, dislocation line motion is governed by a partial differential equation (PDE) in time (Eq. 2.4). After discretizing the dislocation lines, we can write this governing equation in terms of the motion of the nodes or segments, converting the PDE into a coupled system of N ordinary differential equations (ODEs). For example, in the nodal representation we have

$$\frac{d\mathbf{r}_i}{dt} = \mathbf{g}_i\left(\left\{\mathbf{r}_j\right\}, \boldsymbol{\sigma}_{\text{ext}}, \ldots\right) \tag{2.14}$$

where $\mathbf{r}_i$ is the 3×1 position vector of node i and brackets denote the set of all nodes. In DD, we solve these ODEs using *time integration*, an approach where the solution is found over a series of sequential time steps. Many methods exist for time integrating coupled systems of ODEs, and in this section we discuss a few in the context of DD. In the following, for clarity we will assume $\left\{\mathbf{r}_j\right\}$ is the only argument of $\mathbf{g}(\cdot)$.

The simplest time integration scheme is the forward Euler method, which has the following form:

[1] A more rigorous definition can be written in terms of the line shape functions [9, 35].

$$\mathbf{r}_i^{k+1} = \mathbf{r}_i^k + \Delta t\, \mathbf{g}_i\left(\left\{\mathbf{r}_j^k\right\}\right). \tag{2.15}$$

Superscripts denote the time step number and Δt is the time step size. In this scheme we assume that the nodes maintain their current velocities over the duration of the time step, and update their positions accordingly. The forward Euler method is commonly used in DD simulations. One issue with this approach is that the error it introduces is unknown (without additional numerical methods). All time integration schemes introduce error and we must ensure this error does not overwhelm the solution. A simple method that provides an error estimate is the Heun method:

$$\mathbf{r}_{i,0}^{k+1} = \mathbf{r}_i^k + \Delta t\, \mathbf{g}_i\left(\left\{\mathbf{r}_j^k\right\}\right) \tag{2.16a}$$

$$\mathbf{r}_{i,l+1}^{k+1} = \mathbf{r}_i^k + \frac{\Delta t}{2}\left[\mathbf{g}_i\left(\left\{\mathbf{r}_j^k\right\}\right) + \mathbf{g}_i\left(\left\{\mathbf{r}_{j,l}^{k+1}\right\}\right)\right] \tag{2.16b}$$

$$e = \max_i \|\mathbf{r}_{i,l+1}^{k+1} - \mathbf{r}_{i,l}^{k+1}\|. \tag{2.16c}$$

Eq. (2.16a) is the forward Euler "predictor" and Eq. (2.16b) is the trapezoidal method "corrector." The corrector can be applied arbitrarily many times using a fixed-point iteration, with the second subscript denoting the iterate number, until the error estimate of Eq. (2.16c) falls below some user-specified tolerance. If the solution does not converge in a prespecified number of iterations, the time step must be reduced and the method applied anew. Note that in addition to providing an error estimate, the Heun method is globally second order accurate, meaning the solution converges as $\mathcal{O}(\Delta t^2)$, whereas the forward Euler method is only first order accurate, $\mathcal{O}(\Delta t)$. The Heun method is the default time integrator in ParaDiS.

Time integration turns out to be a challenging problem in DD, and is an active area of research. We defer discussion of more advanced topics, such as implicit time integration and subcycling, to Sect. 2.5.

2.2.2.5 Dislocation Collisions

When dislocation lines collide, they can react and form junctions or annihilate. The resulting junction formation and annihilation events can significantly influence the evolution of the dislocation structure. Hence, detecting and handling collision events reliably is important. To detect the collision of dislocation lines, a number of approaches have been developed. The simplest is a proximity-based algorithm, which assumes two lines have collided if they come within a user-defined minimum distance of each other. This approach can miss collisions, however, if dislocation lines are displaced too far in a time step. More advanced algorithms can safeguard against missing collisions [96]. Once a collision is detected, the appropriate topological changes must be made. The conservation of the Burgers vector must be invoked to determine the Burgers vectors of resulting segments. For instance, if two segments with opposite Burgers vectors collide they will annihilate with each other.

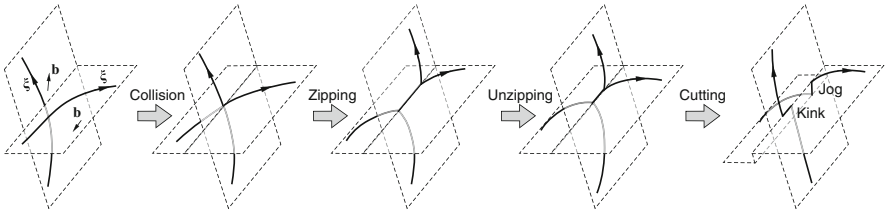

Fig. 2.5 Schematic showing the process of dislocation line collision, the zipping of a junction, subsequent unzipping, and then final dissolution after the dislocations cut each other. The cutting results in the formation of a jog and a kink

2.2.3 Additional Aspects

The fundamentals presented in Sect. 2.2.2 provide the basic toolset necessary to run a simple DD simulation. For example, the Frank–Read source simulations presented in Sect. 2.3.5.1 can be conducted using these methods. More advanced simulations require additional details, some of which are presented in this section.

2.2.3.1 Junctions and Dislocation Intersections

The discussion of dislocation collisions in Sect. 2.2.2.5 does not consider how to handle the formation and dissolution of dislocation junctions; we will elaborate these details here. When two dislocation lines moving in different planes collide, one of two things may occur. They may cut through each other and continue their motion, potentially producing Burgers-vector-sized steps on the lines known as jogs or kinks (depending on whether they are out of or within the glide planes, see Fig. 2.5). Or, they may zip together and react to annihilate or form a junction. Even if a junction does form, it may be ripped apart if a large enough force is applied, and the lines may cut each other and continue on as if the junction had never existed; this process is depicted in Fig. 2.5. Accurately capturing these behaviors is important because sessile (immobile) junctions (often referred to as locks) and dislocation intersections are thought to play vital roles in work hardening.

Considering this process in the context of DD, there are (at least) three different steps that need to be considered. First, the collision of dislocation lines needs to be detected, the result of which is a point junction between the two lines. The resulting point junction can then either zip together and form a proper junction, or split apart and possibly produce jogs and/or kinks. In some codes, the lines never formally react, and instead simply approach each other closely and align parallel to each other when forming a junction [24]. If the lines do formally react, the code must be able to detect whether the formation of a junction is favorable. This is typically done by applying an energy criterion to ensure that the system moves towards a state that maximizes its dissipation rate. Common examples include approximations

based on line energy arguments [108], tests to see if the involved lines are moving apart [90], and the principle of maximum dissipation [9], which approximates the dissipation rate as the dot product of the nodal force with the velocity and seeks to maximize it. If it is decided that the point junction should instead split in such a way that lines cut each other, there may be an energy barrier inhibiting this split due to, for example, the formation of jogs. This barrier can be accounted for in terms of a splitting rate through the use of thermal activation theory (see Sect. 2.2.3.3), or athermally in terms of a junction strength dictating the minimum stress that must be applied for the split to occur. As an example for the latter scheme, Kubin et al. [54] have developed the following law to determine the strength of a junction:

$$\tau_{\mathrm{j}} = \frac{\beta \mu b}{l_{\mathrm{u}}} \tag{2.17}$$

where μ is the shear modulus, l_{u} is the length of the dislocation arms surrounding the junction, and β is a material constant that must be determined from experiments or atomistic simulations. If a cutting event like this does occur, the resulting jogs can influence the mobility of the dislocation lines [42]. However, most DD codes do not account for the presence of jogs.

2.2.3.2 Boundary Conditions

As with any initial-boundary value problem, the boundary conditions (BCs) need to be stated in order to have a well-defined problem. The specific form of the BCs is dictated by the geometry of interest. The types of BCs used in DD simulations can be categorized into three groups as shown in Fig. 2.6: (a) infinite BCs, (b) periodic BCs, and (c) heterogeneous BCs.

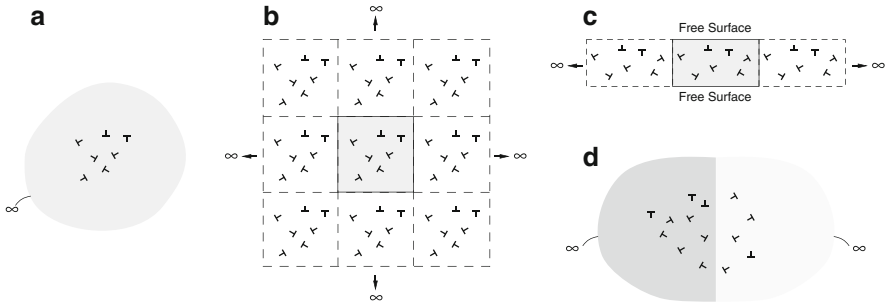

Fig. 2.6 Schematic depictions of different types of boundary conditions. (**a**) Infinite BCs, (**b**) periodic BCs, and heterogeneous BCs with (**c**) a free standing film with in-plane periodic BCs and (**d**) a bimaterial interface in an infinite medium. For simplicity, the dislocations are represented by the *perpendicular* symbol, even though the figure refers to 3D DD simulations

Infinite BCs (Fig. 2.6a) are the simplest, and correspond to the simulated dislocation lines being embedded in an infinite medium. Enforcement of an infinite BC in any coordinate direction requires simply that we allow dislocations to move an arbitrary distance along that axis. The stress expressions for dislocation lines (and even other defects) are known for an isotropic, homogeneous, infinite medium, so they may be implemented readily.

While infinite BCs provide a reasonable model for the behavior of dislocation lines far from free surfaces (i.e., in the bulk), it is computationally infeasible to keep track of all dislocation lines in an infinite medium that has a finite average dislocation density. This makes infinite BCs primarily useful for idealized test cases. Periodic BCs, in contrast, mimic an infinite medium while allowing for a nonzero average dislocation density. With periodic BCs, the simulation cell represents a so-called *supercell* which is repeated in all directions ad infinitum—Fig. 2.6b depicts this idea for a 2D geometry. The replicas surrounding the main simulation cell are called *images*. Periodic BCs provide a model for the simulation of bulk metals, where the material element being simulated is in the middle of a specimen many times larger than the cell. Any dislocation configuration or pattern, however, whose characteristic length scale is larger than the supercell cannot be captured with periodic BCs. To enforce periodic BCs, the total stress field due to every dislocation line in each of the infinite number of periodic images must be computed to determine the driving forces. In practice, only a finite number of images is considered, however, care must be taken to ensure the resulting stress field is well defined (due to conditional convergence [13, 55]). When a dislocation line crosses the supercell boundary, its next image over will enter the supercell from the opposing boundary. See [9] for a more detailed discussion of periodic boundary conditions.

The final type of boundary condition we will discuss applies to a much broader class of problems. In the case of a heterogeneous BC (Fig. 2.6c), some feature of the geometry breaks the homogeneity of the domain. Common examples are free surfaces, with geometries like cylinders, thin films, and half spaces, and bimaterial interfaces, as in the case of a layered material. As was discussed in Sect. 2.2.2.1, since analytic stress expressions generally only apply to an infinite, homogeneous medium, a corrective image stress field must be determined. Image stress solvers have been developed using the finite element method [100, 103, 108, 110], Fourier methods [32, 105, 107], and boundary element methods [26], as well as various other methods [29, 41, 49] to solve for the image field.

As a final note, we point out that these BCs can be combined. For instance, we may simulate a freestanding thin film [107] by employing periodic BCs in one or two coordinate directions and free surface BCs in the others (Fig. 2.6c).

2.2.3.3 Cross-Slip

Conservative dislocation motion occurs when no atomic diffusion is required and is termed dislocation glide. Nonconservative motion, on the other hand, requires the diffusion of vacancies and is referred to as dislocation climb. Assuming a dislocation

only moves by conservative glide (i.e., no climb), it is confined to motion within the plane that contains both its Burgers vector and its line direction—this defines its glide plane. In the case of a screw dislocation, because the Burgers vector is parallel to the dislocation line, a unique glide plane cannot be defined. In principle, a screw dislocation can glide in any plane that contains its Burgers vector. Most of the time, however, dislocations prefer to glide along a few families of crystallographic planes that minimize their core energy, and this sets the slip systems for that metal. For instance, in FCC metals dislocations usually glide in $\{1\,1\,1\}$ planes, giving each screw dislocation two viable glide planes. The process of a screw dislocation changing from one glide plane to another is called *cross-slip*. Cross-slip is thought to be an important feature of dislocation motion, and in this section we will briefly outline the key aspects relevant to DD.

Cross-slip is known to be a *thermally activated* process [79]. This means that there is an energy barrier associated with its occurrence, and this barrier can be overcome by thermal fluctuations. The rate at which a thermally activated event occurs can be approximated with an Arrhenius-type relationship [51]:

$$R = \nu_0 \exp\left(-\frac{E_b}{k_\mathrm{B} T}\right) \tag{2.18}$$

where R is the rate in number of events per unit time, E_b is the energy barrier, k_B is Boltzmann's constant, T is the absolute temperature, and ν_0 is the attempt frequency. Often, the attempt frequency is approximated as $\nu_0 = \nu_D(L/L_0)$, where ν_D is the Debye frequency, L is the length of the dislocation segment, and L_0 is a reference length. Thus, in order to determine the cross-slip rate at a specified temperature, one needs to know the energy barrier and the attempt frequency. Atomistic simulations have commonly been used to determine these quantities, often finding that the energy barrier is sensitive to the local stress state (see Sect. 2.4.1).

Using thermal activation theory, cross-slip can be implemented in DD as follows. We test for cross-slip events once during each time step. We loop over all dislocation lines, looking for segments that are of screw character. If a screw segment is found, the energy barrier is calculated based on the local stress state at that segment, with which the cross-slip rate can be determined using Eq. (2.18). The cross-slip probability is then simply $R\Delta t$, where Δt is the time step size. We then select a random number ζ uniformly distributed in [0,1], and cross-slip occurs if $R\Delta t > \zeta$. The most difficult aspect of implementing this model is determining how the energy barrier depends on the local conditions (e.g., stress, local dislocation configuration, etc.).

2.2.3.4 2-Dimensional Dislocation Dynamics

As we have shown, fully three-dimensional dislocation dynamics simulations are complex and computationally expensive. Consequentially, many researchers have sought to develop dislocation dynamics in two-dimensions [3, 8, 18, 21, 37, 39,

72, 93, 100, 111]. In two-dimensional dislocation dynamics (2DDD), dislocations are assumed to be infinitely long and straight, so that they can be represented by point objects in the plane perpendicular to the dislocation line; this approximation greatly reduces the number of degrees of freedom and removes the need to track the complex topology present in three dimensions. 2DDD codes run much faster and can achieve much larger amounts of plastic deformation than 3D codes. However, these advantages are offset by the limited subset of problems that can be faithfully represented in two-dimensions (e.g., fatigue problems where dislocations are often long and straight). Because many physical phenomena are absent in the 2D picture, additional physics, such as sources for multiplication and obstacles [8, 16, 100], must be added. While many important contributions to DD have been made in a 2D setting, we will not elaborate on 2DDD further.

2.3 Running a DD Simulation

Over the past several decades, DD has been utilized to study a range of problems in crystal plasticity. While the specific details surrounding each of these simulations vary, they all share a number of basic ingredients. In this section we will briefly discuss each of these ingredients, and then provide several case studies.

2.3.1 Types of Simulations

DD simulations can be categorized into two groups: (1) small-scale—those interested in the interactions and behavior of one or a few dislocation lines and (2) large-scale—simulations examining the collective behavior of many dislocations. Examples of small-scale simulations include the simulation of intersecting dislocation lines, junction formation, and junction dissolution [58, 60]; the interaction of dislocations with precipitates [80] and solutes [17, 69]; and the interaction of dislocations with free surfaces [49, 98, 106]. Simulations of large-scale collective behavior generally involve simulating the stress–strain response of a material, with examples including work hardening in bulk metals [5, 11, 22, 96], the plasticity of micropillars [2, 19, 88, 106, 109], and plasticity during nanoindentation [30, 103]. Details below will be presented in terms of the two simulation types.

2.3.2 DD Codes

There are currently about a dozen 3D dislocation dynamics codes in use. Here we will briefly discuss some of their differences to aid the user in making a selection. See [28], [52], or [78] for additional reviews of DD codes.

As discussed in Sect. 2.2.2.3, DD codes can be categorized as either lattice-based or node-based. In practice, each of these discretization schemes has its own advantages and disadvantages in terms of accuracy, computing efficiency, simplicity, and flexibility. A strength of lattice-based simulations is that force calculations and tracking of dislocation intersections are simplified, since only a finite set of dislocation configurations (dictated by the lattice) are considered [24, 52].

With the node-based scheme, since dislocation segments can take any arbitrary orientation, dislocation lines tend to be smoother. In contrast, with lattice-based DD, the angles between neighboring segments remain unchanged regardless of how much the lattice or segment length is refined.

microMegas [24, 65] and TRIDIS [99, 103] are two examples of lattice-based DD codes dedicated to the 3D DD simulations of crystalline solids. microMegas, an open source code written mainly in Fortran, utilizes a base of eight line vectors per slip system, for describing dislocation lines in FCC, BCC, and HCP crystals, in addition to a few mineral materials. TRIDIS, suitable for the study of the mechanical response of FCC and BCC metals and alloys, is a parallel code that uses four line vectors per slip system and has been coupled to the finite element code CAST3M.

There are many node-based codes available and we briefly discuss a few. Parametric dislocation dynamics (PDD) is the only code which uses curved (cubic) dislocation segments [35, 77]; it was recently made open-source and renamed mechanics of defect evolution library (MODEL) [67, 77]. Multiscale dislocation dynamics plasticity (MDDP) [64] is a hybrid code coupling dislocation dynamics, continuum finite elements, and heat transfer models. Its DD code was originally named micro3d and was later implemented in MDDP. PARANOID [89] is a DD code suitable for DD simulations of thin films, strained layers, and bulk metals and semiconductors. Parallel dislocation simulator (ParaDiS) [5, 76] is an open-source, massively parallel DD code that has mobility laws implemented for FCC and BCC crystals incorporating glide and climb. NUMODIS [73] is a recently developed open-source, parallelized code, with features for simulations of polycrystals and polyphases.

2.3.3 Input Specification

Usually, DD simulations are controlled through two (or more) different input files. The *control file* specifies the parameters of the simulation. These include the material properties (elastic constants, drag coefficients, etc.), the loading conditions (strain rate, stress state, etc.), the numerical parameters (time step size, remeshing parameters, etc.), and output controls (e.g., what output to generate and how frequently). The *structure file* specifies the initial dislocation configuration and the geometry of the simulation cell or boundaries. This generally requires specifying where nodes are located, how segments connect the nodes, and what their Burgers vectors are. In the next section we will discuss how to select the necessary parameters and design a DD simulation.

2.3.4 Designing a Simulation

2.3.4.1 Initial Configuration

The initial dislocation configuration will be dependent upon the type of simulation. In small-scale settings, generally a few initially straight lines are used, and the dislocation character angle is often varied to see the different effects. The specific goals of the simulation will decide the initial geometry.

In large-scale simulations, initial configuration selection is more complex [71]. Usually, the initial configuration is intended to emulate a specific material state, for example, an annealed or cold-worked metal. The DD simulation would then predict the response of a material in such a state to the chosen loading. However, the full three-dimensional detail of dislocation structures in materials is generally not known; this means the initial configuration will have to be approximated somehow. Often, the following procedure is used. First, a simulation cell is populated with a chosen initial density of straight dislocation lines, usually randomly oriented and positioned. Then, the simulation cell is allowed to *relax*—equilibrate under zero imposed stress—until the dislocation structure reaches a meta-stable configuration. Once relaxed, the configuration may be used for further simulations.

2.3.4.2 Loads and Boundary Conditions

As with most solid mechanics simulations (and experiments), there are two common types of loadings in DD: *stress-controlled* and *strain-controlled*. Under stress-control, often referred to as creep loading, the applied stress is specified and the dislocation lines simply respond to the Peach-Kochler forces resulting from the applied stress and the stress fields of other dislocations. The stress state may be constant or vary in time.

Under strain-control, usually a strain rate tensor, $\dot{\epsilon}_{ij}$ is specified and the resulting stress state must be calculated as follows. The total strain at any time t is

$$\epsilon_{ij}^{\text{tot}}(t) = \int_0^t \dot{\epsilon}_{ij}(\lambda) d\lambda \,. \tag{2.19}$$

When $\dot{\epsilon}_{ij}$ is a constant, the result is simply $\epsilon_{ij}^{\text{tot}}(t) = t\dot{\epsilon}_{ij}$. Using the procedure discussed in Sect. 2.3.4.3, the plastic strain due to the motion of the dislocation lines at time t, $\epsilon_{ij}^{\text{p}}(t)$, can be determined. The elastic strain is then $\epsilon_{ij}^{\text{el}}(t) = \epsilon_{ij}^{\text{tot}}(t) - \epsilon_{ij}^{\text{p}}(t)$ (assuming infinitesimal deformations), which is related to the stress through Hooke's law. For an isotropic linear elastic material with Lamé constants λ and μ (the shear modulus), they are related by

$$\sigma_{ij} = \lambda \bar{\epsilon}^{\text{el}} \delta_{ij} + 2\mu \epsilon_{ij}^{\text{el}} \tag{2.20}$$

where δ_{ij} is the Kronecker delta and $\bar{\epsilon}^{\text{el}} = \frac{1}{3}(\epsilon^{\text{el}}_{xx} + \epsilon^{\text{el}}_{yy} + \epsilon^{\text{el}}_{zz})$ is the hydrostatic elastic strain. At each time step, the increments of total strain and plastic strain are computed, and then the stress state is updated according to Eq. (2.20).

The two loading conditions can also be combined. For instance, in the commonly used uniaxial tension loading condition, a normal strain rate is imposed along the loading direction while all other stress components are set to zero. In this case, assuming the imposed uniaxial strain rate is $\dot{\epsilon}_{xx}$, the externally applied stress state at any point in time is simply $\sigma_{xx} = E(t\dot{\epsilon}_{xx} - \epsilon^{\text{p}}_{xx}(t))$, where E is the Young's modulus.

As discussed in Sect. 2.2.3.2, the boundary conditions will depend on the problem of interest. Periodic BCs are used to simulate bulk material response. Often infinite BCs are used when we are interested in the behavior of a few isolated dislocation lines. When running simulations under periodic boundary conditions, the size of the simulation cell is an important feature of the simulation; any dislocation structure whose length scale is larger than the simulation cell width cannot be accurately represented. Furthermore, if the cell is too small the interaction between a dislocation and its own periodic image can yield artificial behaviors.

2.3.4.3 Outputs

With DD, the positions of all the dislocation lines are known at each time step. This means that specific features of the dislocation structure can be extracted directly. For instance, we can determine how common a particular type of junction is or how predominant different line orientations are (e.g., edge versus screw). Often, it is useful to express features of the dislocation structure in terms of their density, ρ, the dislocation line length per unit volume (in units of [length]$^{-2}$). For example, a dislocation structure could be characterized in terms of the densities of the different slip systems. The density of a dislocation population can be computed by simply summing the length of all relevant segments and dividing by the simulation volume.

An important output for DD simulations is the plastic strain; it is needed for computing the stress state under strain-control (discussed in Sect. 2.3.4.2). In DD simulations, plastic deformation is produced by the motion of the dislocation lines. The area swept out by a dislocation segment in a time step is proportional to the plastic strain produced in the crystal according to the relation [4]

$$\delta\epsilon^{\text{p}}_{ij} = \frac{b_i n_j + b_j n_i}{2\Omega}\delta A \tag{2.21}$$

where δA is the area swept out by the dislocation segment during its motion, Ω is the simulation volume, **b** is the burgers vector, and **n** is the slip plane normal. The total plastic strain produced in a time step is the summation of Eq. (2.21) over all dislocation segments.

2.3.4.4 Solution Convergence

As with any numerical simulation technique, it is important to ensure that the errors introduced by our discretizations in space and time are sufficiently small so that the solution converges. In DD, this means ensuring the time step and dislocation segments are small enough.

Two approaches have been used in DD to confirm that the time step size is adequately small. The first was discussed in Sect. 2.2.2.4, and involves approximating the truncation error of the time integrator, and selecting the time step size so that it falls below a user-specified tolerance. Another approach is to limit the time step size so that the dislocation structure does not change too much from step to step. This usually involves specifying a maximum displacement and/or rotation allowed for any dislocation segment during a time step, and limiting the time step so they are obeyed. While this approach does not directly control the error of the solution, it is commonly used and generally accepted.

Spatial discretization error is dictated by how well the discretized structure approximates the actual smooth structure of interest. The goal of the remeshing algorithms discussed in Sect. 2.2.2.3 is to provide a means for controlling the quality of the discretization. The remeshing algorithm operates according to the chosen remeshing parameters—the maximum and minimum segment lengths and areas. As these parameters are reduced, the discretization becomes more and more refined, and the discretization error is reduced. A refined structure is more accurate, but is also more computationally expensive. This is also true when choosing the shape of the dislocation segments. The cubic segments used in PDD better reproduce smooth dislocation structures, but at the cost of increased computational complexity. The user must decide where his or her simulation falls in the trade-off between speed and accuracy.

2.3.5 Example Simulations

Here we present three case studies showing the basics of running a DD simulation. First, we determine the activation stress of a Frank–Read source using the lattice-based code microMegas and the node-based code ParaDiS. Second, we examine the activation of a single-arm source in a micropillar using ParaDiS. Finally, we show results from a few simple work hardening simulations using ParaDiS. All simulations use the material properties for nickel at $T = 300\,\mathrm{K}$, which are given in Table 2.1, and the FCC mobility law presented in Sect. 2.2.2.2.

2.3.5.1 Case Study 1: Activation Stress of a Frank–Read Source

The Frank–Read source is a canonical case study in dislocation theory, showing how a single dislocation can multiply indefinitely by simply gliding in its slip plane under an applied shear stress [45]. A Frank–Read source can be modeled

Table 2.1 Parameters for nickel at $T = 300$ K used in DD simulations in all case studies

	microMegas			ParaDiS		
Property	Name	Value	Unit	Name	Value	Unit
Shear modulus (μ)	`ModuleG0`	76.0	GPa	`shearModulus`	76e9	Pa
Poisson's ratio (ν)	`DPOISS`	0.31	–	`pois`	0.31	–
Burgers vector (b)	`VecBurgers`	2.49	Å	`burgMag`	2.49e-10	m
Core energy parameter[a]		–		`Ecore`	6.05e9	Pa
Drag coefficient (B)	`Coef_visqueux`	1.61e-5	Pa s	`MobEdge`	62,112.0	$(\text{Pa s})^{-1}$
or Mobility (M)				`MobScrew`	62,112.0	$(\text{Pa s})^{-1}$
Core radius (r_c)		–		`rc`	1.0	b
Error tolerance		–		`rTol`	2.0	b
Reference scale	`Echelle`	6.75	–		–	
Time step size	`deltat0`	1e-12	s	Variable based on rTol		
Line tension type	`LINTEN`	4 (Mohles)			–	

Unspecified parameters were set to their default values. The mobility or drag coefficients are obtained from atomistic simulations [75]

[a]In ParaDiS, the core energy parameter `Ecore` controls the scaling of the core energy in the same way the $\mathscr{E}$ parameter does in Eq. (2.6) (hence it has units of Pa). The `Ecore` value used here leads to a core energy per unit length which scales as $\frac{\mu b_i^2}{4\pi}$, where b_i is the magnitude of the Burgers vector of segment i.

by considering a straight dislocation line of length L lying in its slip plane that is pinned at both ends. These pinning points could represent intersections with forest dislocations, impurities, obstacles, or any number of other pinning sites that occur in real metals. A force per unit length of τb (b is the magnitude of the Burgers vector) will be experienced by the dislocation line when a shear stress with magnitude τ is applied, which in turn causes the line to bow out. As τ increases, the radius of curvature decreases until the shear stress reaches τ_{act}, the activation stress. Figure 2.7a and e shows the configuration when $\tau = \tau_{\text{act}}$ from simulations in microMegas and ParaDiS, respectively. At stresses above τ_{act}, the line is able to bow around completely and partially annihilate with itself, as shown in (b,c) and (f,g). This process produces a new dislocation loop, shown in (d) and (h), which is free to continue expanding. The objective of this case study is to determine the activation stress for the Frank–Read source.

For these simulations, we choose a line direction of $[1\,\bar{1}\,2]$ and a Burgers vector of $\frac{b}{\sqrt{2}}[1\,1\,0]$, corresponding to an edge dislocation in an FCC metal with glide plane normal $(\bar{1}\,1\,1)$. The x-, y-, and z-axes of the coordinate system are along the $[1\,0\,0]$, $[0\,1\,0]$, and $[0\,0\,1]$ directions, respectively. The end nodes are flagged as immobile (velocities set to zero). Simulations were run under stress-control, with σ_{yy} applied to produce a resolved shear stress on the glide plane of τ, and all other stress components were set to zero. By slowly increasing the applied stress with increments of $\Delta\tau = 0.5$ MPa and monitoring for activation, we can determine the activation stress to within $\pm\Delta\tau/2$. To detect activation, we can simply watch for

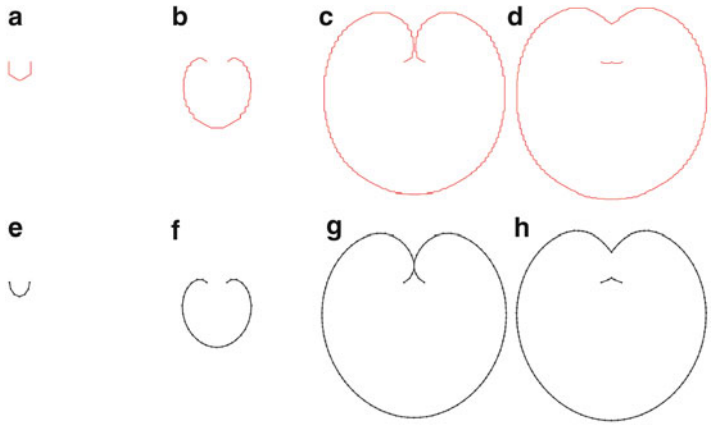

Fig. 2.7 Snapshots from Frank–Read source simulations using (**a**–**d**) microMegas with $L/\bar{l} = 2.4$ and (**e**–**h**) ParaDiS with $L/l_{\max} = 2.4$, where $L = 0.596\,\mu$m ($2400b$) is the distance between the pinning points. ParaDiS graphics made with AtomEye [59]

whether activation occurs, or examine the plastic strain as a function of time—the plastic strain will plateau if the source is not activated. This is by no means the only possible approach for computing the activation stress, and strain-control or a combination of stress- and strain-control (like with microMegas deformation mode 6) could also be used.

The coarseness of the representation of the Frank–Read source will affect the activation stress. As the segment length is reduced, the activation stress should reach a converged value. Figure 2.8a shows the effect of the segment length on the activation stress with source length $L = 0.596\,\mu$m ($2400b$) in both codes. It is clear from the figure that the solution converges as smaller segments are used in both codes, and large segment lengths overestimate (in ParaDiS) or underestimate (in microMegas) the activation stress by up to 10 % in the parameter space considered here.

We can also study the effect of the source length, L, on the activation stress. Figure 2.8b shows the results for L ranging from 0.298 to 1.79 μm ($1200b$ to $7200b$) when $L/\bar{l} = L/l_{\max} = 10$. The activation stresses estimated by the two codes differ by at most 2.2 % for all source lengths tested here. Also provided is the fit utilized by Foreman [31] of the form $\tau_{\text{act}} = [(A\mu b)/(4\pi L)]\ln(L/r_c)$ where r_c is the core radius, with $A = 1.2$, close to unity as Foreman found for an edge source.

2.3.5.2 Case Study 2: Spiral-Arm Source Activation in a Cylinder

In nanomaterials, the small specimen size allows spiral-arm or single-arm sources to operate. This type of source is similar to a Frank–Read source, except that only one end of the source is pinned in place; the rest of the source is free to rotate about

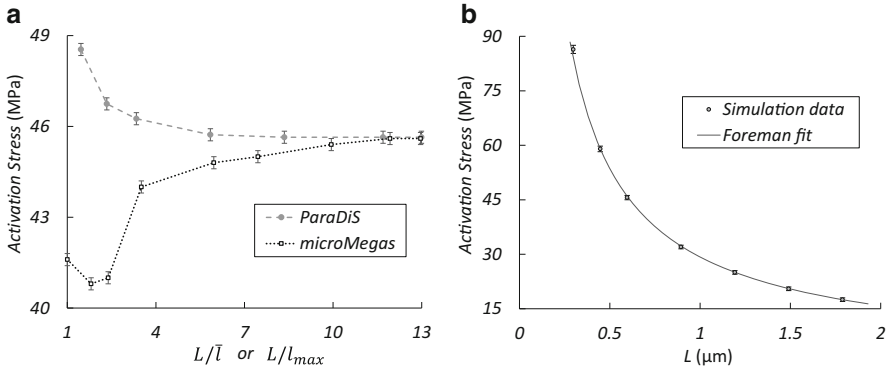

Fig. 2.8 (**a**) Activation stress τ_{act} of an edge Frank–Read source with $L = 0.596\,\mu\text{m}$ ($2400b$) as a function of the inverse of the segment length. The maximum segment length in ParaDiS and microMegas is, respectively, controlled by `maxSeg` ($l_{\max}$) and `Ldis` ($\bar{l}$). Error bars show the inaccuracy caused by $\Delta\tau$, the stress increment used. (**b**) Activation stress τ_{act} as a function of the source length with $L/\bar{l} = L/l_{\max} = 10$. Error bars include the differences between microMegas and ParaDiS as well. See text for explanation of Foreman fit

the pinning point under an applied stress. The result is a spiral-shaped dislocation line generating plastic strain with each revolution.

We here study the behavior of a single-arm source in a cylindrical specimen of radius R oriented along the [0 0 1] direction, imitating a source in a micropillar. For our source geometry, we choose a screw dislocation in an FCC metal with Burgers vector and line direction $[0\,1\,\bar{1}]$ and glide plane normal (1 1 1) that has a Lomer jog at its mid point. The Lomer jog is a section of dislocation with line direction $[0\,\bar{1}\,\bar{1}]$ which is out of the glide plane and treated as sessile in our simulation; in this way, the jog provides the pinning points for the source. Lomer jogs can form during plastic deformation when dislocations react and are often thought to act as immobile locks. We choose a jog height of $0.141\,\mu\text{m}$ ($566b$) for all simulations.

The simulation geometry is shown in Fig. 2.9. Initially, the arms are straight (Fig. 2.9a). Under an applied compressive stress σ_{zz}, the source will begin to rotate. Once again, the application of a stress greater than the activation stress σ_{act} is necessary for the source to activate and rotate freely about the jog. Figure 2.9b shows the configuration at activation when $R = 0.37\,\mu\text{m}$ ($1500b$). For this case study, we will again examine the activation stress, but now focusing on the effects of the free surface. The same procedure with stress-control taking steps of $\Delta\sigma = 0.5\,\text{MPa}$ is employed. We use ParaDiS to simulate the activation process, with a Fourier-based image stress solver [105]. A fast Fourier transform is used over a uniform grid on the surface of the cylinder to determine the image stress field. As with the discretization length, this grid spacing must be small enough to achieve a convergent solution. Periodic boundary conditions are used at either end of the cylinder, with a cell height of $6R$, which gives us an approximately square $n \times n$ grid. The maximum segment length was set to $l_{\max} = 2R/15$.

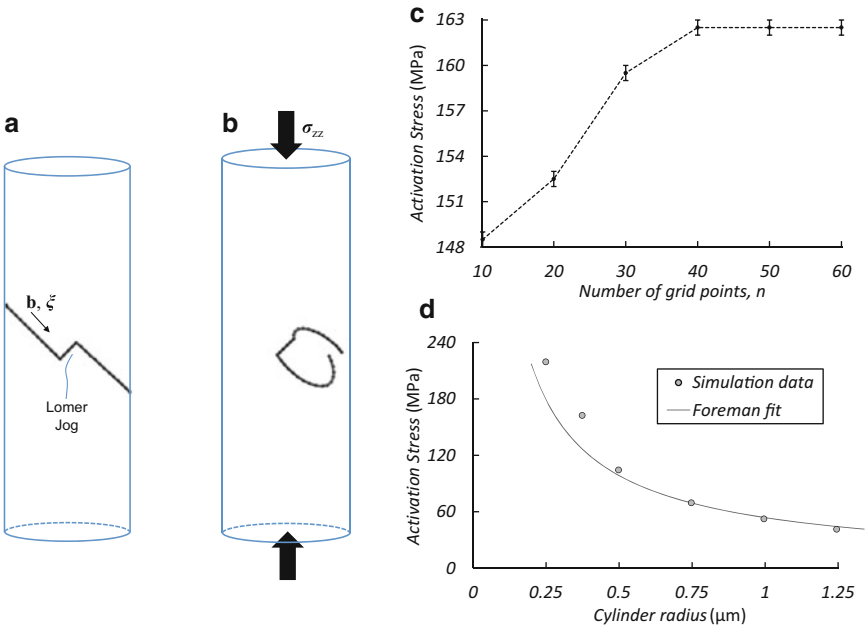

Fig. 2.9 Snapshots and results from single-arm source simulations using ParaDiS. (**a**) Initial configuration. (**b**) Configuration slightly below the activation stress when $R = 0.37\,\mu\text{m}$ ($1500b$). (**c**) Convergence of the activation stress as the number of grid points used for image stress calculation, n, is increased, with $R = 0.37\,\mu\text{m}$ ($1500b$). (**d**) Activation stress as a function of the cylinder radius. See text for the definition of Foreman fit. Graphics made with AtomEye [59]

First we examine the convergence behavior of the image stress solver. Figure 2.9c shows how the activation stress varies with the number of grid points. We see that about a 40×40 grid is required to achieve a converged result. Figure 2.9d demonstrates the dependence of the activation stress on the cylinder radius in the range $R = 0.124$–$1.24\,\mu\text{m}$ ($500b$–$5000b$) using $n = 50$ for the image stress calculation. At the larger radii, the activation stress again follows the Foreman behavior with $A = 2.15$ (using R in place of L), however, the smaller cylinders yield slightly higher values than the Foreman estimate.

2.3.5.3 Case Study 3: Bulk Plasticity Simulation

During plastic deformation the dislocation density tends to increase, causing the material to strengthen. This behavior is called *work-hardening* or *strain-hardening*. The study of work hardening is a key research area ripe for DD simulations. We close out this section with a few work hardening simulations.

For our work hardening simulations we use ParaDiS with a $10 \times 10 \times 10\,\mu\text{m}$ simulation cell, imposing periodic boundary conditions in all directions. No cross-

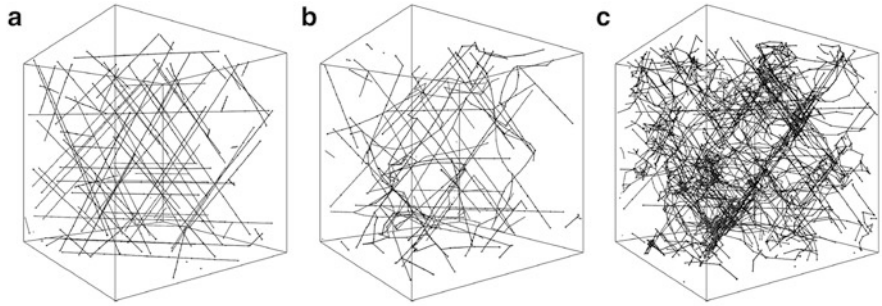

Fig. 2.10 Snapshots of work-hardening simulation performed on nickel at 300 K. (**a**) Initial configuration, (**b**) after relaxation, (**c**) dislocation microstructure at 0.5 % strain with [0 0 1] uniaxial loading. Graphics made with AtomEye [59]

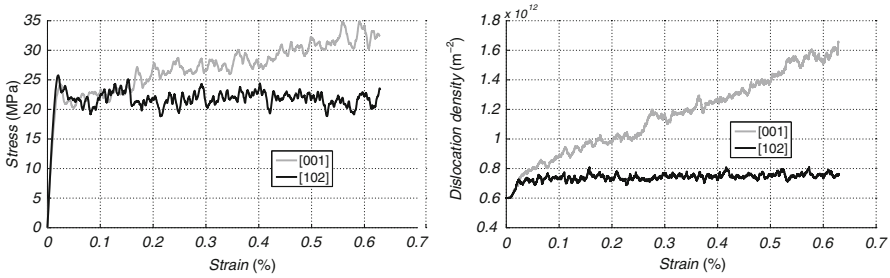

Fig. 2.11 Stress–strain and dislocation density-strain curves for the different loading directions

slip was allowed. The remesh parameter $l_{\max}$ (`maxSeg`) was set to 1.25 μm (5000b) (other remesh parameters were set to defaults). We start with 50 straight dislocation lines with a 60° character angle, random {1 1 1}-type glide plane, and random ⟨1 1 0⟩-type Burgers vector, as depicted in Fig. 2.10a, and then allow the system to relax under zero applied stress until it has reached the equilibrium configuration shown in Fig. 2.10b.

We study the response of the system under uniaxial tension with a constant strain rate of $10^3\ \mathrm{s}^{-1}$ applied in the [0 0 1] and [1 0 2] directions. A recently developed subcycling-based time integrator was used [96], with simulations run for 40 and 7.2 h on a single CPU for [0 0 1] and [1 0 2] loading, respectively. The resulting dislocation configuration after a total strain of 0.5 % in the [0 0 1] direction is shown in Fig. 2.10c. Figure 2.11 shows the evolution of stress and dislocation density with respect to total strain. The initial yield strengths are similar for both loading directions. However for the [0 0 1] loading the crystal hardens with plastic strain as the dislocation density increases. In comparison, the flow stress and dislocation density remain relatively unchanged for the [1 0 2] loading.

2.4 Relation to Models at Other Length/Time Scales

Dislocation dynamics is just one of many tools that can be used to study the deformation behavior of materials. As this book demonstrates, these various models can be organized into a hierarchy that spans many orders of magnitude in both length and time scale. It is important to understand where a given model falls in this hierarchy, so that its connections to other models can be assessed. As discussed in the introduction, DD simulations are generally run at the length scale of about 0.1–10 μm and at time scales in the range of 1 μs to 1 ms, depending on the material. In this section, we will briefly discuss how DD relates to other material models, and examine a few examples of information propagation from one length/time scale to another.

2.4.1 Lower Scale Models

By its nature as a mesoscale modeling approach, DD requires numerous inputs that describe the physical behavior of dislocation lines. Elasticity and dislocation theory provide much of the information needed to define these models (e.g., Peach–Koehler forces, stress fields of dislocations, etc.). However, certain basic features of the behavior of dislocations are simply out of reach of these types of continuum models. A common example is the dislocation core. Many aspects of a dislocation's behavior are controlled by the structure at the dislocation core. Because the core is composed of a small number of atoms that are displaced far from their equilibrium positions, continuum models are often highly inaccurate. Where these models fail, experiments can be used to inform dislocation physics. However, it is usually challenging to extract information on individual dislocations from experiments.

Atomistic simulations, on the other hand, are well suited to informing DD models. Because the atomistic approach is closer to a "first principles" model, it can be used to study the fundamentals of dislocation physics. Atomistic simulations of one or a few dislocations can be conducted to study basic behaviors with different geometries, loading conditions, and temperature regimes, and this information can be included in the DD framework. Thus, we can think of DD as a model occupying the next larger length/time scale tier above atomistics. Common examples of the transfer of information from atomistic to DD include:

- Dislocation mobilities—This can be in the form of drag coefficients [36, 75, 83] or energy barriers [36, 38, 74, 83].
- Core energies—The core energy affects a number of features, including the core force (as discussed in Sect. 2.2.2.1). Core energy calculations have been carried out for a number of materials [10, 104, 114].
- Strength of junctions—In addition to using the scaling law discussed in Sect. 2.2.3.1, junction strengths can be calculated directly [10, 40].

- Cross-slip rate—Usually, this is calculated in the form of an energy barrier (as discussed in Sect. 2.2.3.3). Examples include the effects of different stress components [47, 56], intersection with forest dislocations [85], the presence of jogs [87, 102], and nucleation at the surface [86]. Many of these results have been incorporated in DD simulations [46].

2.4.2 Higher Scale Models

In the same way that MD can provide inputs for DD simulations, many researchers hope to use DD as a tool for informing higher length/time scale models. For example, a model residing at a larger length/time scale than DD is crystal plasticity (CP). In CP's continuum approach, constitutive laws are defined in terms of phenomenological models based on densities of different dislocation populations (e.g., forest and mobile dislocations). These dislocation densities are tracked at the continuum scale and dictate the loading response of each material element. DD can be used to develop the models which describe the relationship between dislocation densities, stress, and strain, thereby informing CP models.

An example of this transfer of information is the calculation of interaction coefficients in the Taylor hardening model. The generalized Taylor hardening law is commonly used in CP simulations, and states that the flow stress on slip system i is

$$\tau_i = \mu b \sqrt{\sum_j a_{ij} \rho_j} \tag{2.22}$$

where the summation is over all slip systems j, ρ_j is the dislocation density of slip system j, and a_{ij} is a matrix of *interaction coefficients* between the slip systems. The interaction coefficients can be determined using specialized DD simulations that target a specific pair of slip systems. These calculations have been performed for FCC metals [23, 61] and α-iron [81], and have been used to inform CP models [91].

2.4.3 Concurrently Modeling Across Scales

The approaches we have discussed so far involve passing information between modeling approaches using independently conducted simulations. However, it is also possible to transfer information between simulations as they both run concurrently. This approach may be useful in a number of settings. One example is if we are only interested in atomistic resolution over a small part of the domain, such as at the tip of a crack or beneath an indenter. Since atomistic resolution is not needed far from these regions where events such as dislocation nucleation are not occurring, we wish to represent the rest of the domain with a less expensive, higher scale model like DD. The atomistic and DD simulations would then be coupled at their mutual boundaries.

Such an approach has been implemented by Shilkrot et al. [92] in two-dimensions with the coupled atomistic and discrete dislocation (CADD) method for solving plasticity problems. In the CADD approach, the computational domain is divided up into atomistic and continuum regions; molecular dynamics is used in the atomistic region and 2D dislocation dynamics in the continuum domain [20]. For any concurrent modeling approach, the most challenging aspect is coupling the models at their shared domain boundaries. For instance, with CADD the code must detect when dislocations transmit between the domains. CADD has been used to study nanoindentation [66, 92, 93] as well as fracture and void growth [93].

2.5 Challenges and Current Research Topics

Here we will briefly list and introduce a few active research topics in the DD community. Some of the issues driving this research are purely mathematical or numerical in nature—for example, the fact that dislocation interactions cannot be calculated analytically in anisotropic elasticity. Other issues stem from the difficulty of accurately representing atomic-scale phenomena in a mesoscopic framework—for example, accounting for effects of the dislocation core structure. The following list is by no means comprehensive:

- *Time integration*—Efficiently time integrating the equations of motion in DD, i.e., taking a large time step with minimal computational expense, is a challenging but necessary task. Recent work examined implicit time integration methods [33, 43, 94] and time step subcycling [94, 96]. While larger time steps can be achieved with implicit methods, the additional computational cost makes performance gains less significant [33]. With subcycling, it has been shown that 100-fold speed-ups can be achieved [96].
- *Elastic anisotropy*—Most single crystals exhibit anisotropy in their elastic behavior, and yet most DD codes use isotropic elasticity to calculate the interactions between dislocation segments. This is because the analytic expressions for the stress fields of dislocations in anisotropic media are not known, and their numerical calculation is very expensive [112]. An approximate method was recently developed that utilizes spherical harmonics to estimate the interaction forces between dislocations [6]. With this approach, the computational cost can be adjusted according to the desired accuracy of the approximation.
- *Kinematics*—DD simulations are usually run under the assumption of infinitesimal deformations, so that the displacement field surrounding each dislocation is ignored. There are, however, instances where these displacements are known to be important. For instance, a symmetric tilt boundary can be thought of as a vertical array of edge dislocations; however, if the displacement fields of the dislocations are ignored then there is no tilt across the boundary. In addition to this effect, as dislocations move through a crystal, they alter the alignment of the crystallographic planes, i.e., they shift the connectivity of the planes of

atoms. This means two dislocations which are not coplanar initially may have their planes intersected by a series of dislocations which shift them onto the same plane [57]. This effect is related to the fact that when dislocations cut each other, jogs and/or kinks are produced. Incorporation of these effects in DD is challenging.

- *Core effects*—Some features of dislocation behavior are very sensitive to the nature of the core structure. These behaviors are challenging to capture in a framework that smears out all of these details into a simple line object. In some instances, certain features of the core can be included in the formulation presented above, for example, when constructing the mobility law or determining the stress dependence of the cross-slip rate. Sometimes explicit treatment of the core structure is important. For instance, FCC metals with low stacking fault energies have dislocations which are disassociated into Shockley partial dislocations that can be separated by 10s of nm. This can significantly influence the dislocation structures that develop. An approach for incorporating these effects in ParaDiS has been developed [63] .
- *Point defects*—Dislocations interact in a number of ways with point defects such as vacancies and solute atoms. These defects arise quite readily through material processing, alloying, and contamination, and give rise to many phenomena in dislocation physics. For example, solute atoms can accumulate on dislocations, forming so-called Cottrell atmospheres, which can slow down dislocation motion. Additionally, at high temperatures, dislocations are known to move out of their glide planes (climb) by consuming or producing vacancies. Some models have been developed to account for solutes [17, 69] and vacancy-driven climb[7, 70], however only a limited set of geometries have been considered.
- *Inclusions and precipitates*—The interactions of dislocations with inclusions and precipitates give rise to important phenomena such as precipitation hardening and kinematic hardening (Orowan looping). A number of researchers have conducted simulations examining the interaction of dislocations with a few precipitates in simplified settings [50, 68, 80, 84, 97], in addition to a few examples of large-scale simulations [44, 82, 101]. DD models describing the behavior of a dislocation as it cuts through a precipitate are still lacking.
- *Grain boundaries*—Most DD codes are only capable of simulating single crystals, whereas most structural materials are polycrystalline. The grain boundaries separating the individual grains of polycrystals can interact with dislocations in complex ways. Grain boundaries can both absorb and emit dislocations. A grain often experiences "misfit" stresses imposed by the surrounding grains during deformation, which can exert forces on dislocations. Dislocations can also transmit across grain boundaries, from one grain to another. As discussed in Chap. 11, DD simulations have been run with simplified grain and twin boundary models [27, 113], but a robust DD model for polycrystals still requires further development.

Acknowledgements We wish to thank Dr. Benoit Devincre for useful discussions. This work was supported by the US Department of Energy, Office of Basic Energy Sciences, Division of Materials Sciences and Engineering under Award No. DE-SC0010412. Sandia National Laboratories is

a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the US Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

1. R. Abbaschian, L. Abbaschian, R.E. Reed-Hill, *Physical Metallurgy Principles* (Cengage Learning, Stamford, CT, 2009)
2. S. Akarapu, H.M. Zbib, D.F. Bahr, Analysis of heterogeneous deformation and dislocation dynamics in single crystal micropillars under compression. Int. J. Plast. **26**, 239–257 (2010)
3. R.J. Amodeo, N.M. Ghoniem, Dislocation dynamics. I. A proposed methodology for deformation micromechanics. Phys. Rev. B **41**(10), 6958–6967 (1990)
4. A.S. Argon, *Strengthening Mechanisms in Crystal Plasticity*. Oxford University Press, Oxford, 2008)
5. A. Arsenlis, W. Cai, M. Tang, M. Rhee, T. Oppelstrup, G. Hommes, T.G. Pierce, V.V. Bulatov, Enabling strain hardening simulations with dislocation dynamics. Model. Simul. Mater. Sci. Eng. **15**, 553 (2007)
6. S. Aubry, A. Arsenlis, Use of spherical harmonics for dislocation dynamics in anisotropic elastic media. Model. Simul. Mater. Sci. Eng. **21**, 065013 (2013)
7. B. Bakó, E. Clouet, L.M. Dupuy, M. Blétry, Dislocation dynamics simulations with climb: kinetics of dislocation loop coarsening controlled by bulk diffusion. Philos. Mag. **91**, 3173–3191 (2011)
8. A.A. Benzerga, Y. Bréchet, A. Needleman, E. van der Giessen, Incorporating three-dimensional mechanisms into two-dimensional dislocation dynamics. Model. Simul. Mater. Sci. Eng. **12**, 159–196 (2004)
9. V.V. Bulatov, W. Cai, *Computer Simulations of Dislocations* (Oxford University Press, Oxford, 2006)
10. V.V. Bulatov, F.F. Abraham, L.P. Kubin, B. Devincre, S. Yip, Connecting atomistic and mesoscale simulations of crystal plasticity. Nature **391**, 669–672 (1998)
11. V.V. Bulatov, L.L. Hsiung, M. Tang, A. Arsenlis, M.C. Bartelt, W. Cai, J.N. Florando, M. Hiratani, M. Rhee, G. Hommes, T.G. Pierce, T.D. de la Rubia, Dislocation multi-junctions and strain hardening. Nature **440**, 1174–1178 (2006)
12. W. Cai, V.V. Bulatov, Mobility laws in dislocation dynamics simulations. Mater. Sci. Eng. A **387**, 277–281 (2004)
13. W. Cai, V.V. Bulatov, J.P. Chang, J. Li, S. Yip, Periodic image effects in dislocation modeling. Philos. Mag. **83**, 539–567 (2003)
14. W. Cai, V.V. Bulatov, J. Chang, J. Li, S. Yip, Dislocation core effects on mobility, in *Dislocations in Solids*, ed. by F.R.N. Nabarro, J.P. Hirth, Vol. 12, Chap. 64 (Elsevier, Amsterdam, 2004), pp. 1–80
15. W. Cai, A. Arsenlis, C.R. Weinberger, V.V. Bulatov, A non-singular continuum theory of dislocations. J. Mech. Phys. Solids **54**, 561–587 (2006)
16. S.S. Chakravarthy, W.A. Curtin, Effect of source and obstacle strengths on yield stress: A discrete dislocation study. J. Mech. Phys. Solids **58**, 625–635 (2010)
17. Q. Chen, X.-Y. Liu, S.B. Biner, Solute and dislocation junction interactions. Acta Mater. **56**, 2937–2947 (2008)
18. H.H.M. Cleveringa, E. van der Giessen, A. Needleman, Comparison of discrete dislocation and continuum plasticity predictions for a composite material. Acta Mater. **45**(8), 3163–3179 (1997)
19. T. Crosby, G. Po, C. Erel, N. Ghoniem, The origin of strain avalanches in sub-micron plasticity of fcc metals. Acta Mater. **89**, 123–132 (2015)

20. W.A. Curtin, R.E. Miller, Atomistic/continuum coupling in computational materials science. Model. Simul. Mater. Sci. Eng. **11**, R33–R68 (2003)
21. V.S. Deshpande, A. Needleman, E. van der Giessen, Plasticity size effects in tension and compression of single crystals. J. Mech. Phys. Solids **53**, 2661–2691 (2005)
22. B. Devincre, L.P. Kubin, C. Lemarchand, R. Madec, Mesoscopic simulations of plastic deformation. Mater. Sci. Eng. A **309**, 211–219 (2001)
23. B. Devincre, L. Kubin, T. Hoc, Physical analyses of crystal plasticity by DD simulations. Scr. Mater. **54**, 741–746 (2006)
24. B. Devincre, R. Madec, G. Monnet, S. Queyreau, R. Gatti, L. Kubin, Modeling crystal plasticity with dislocation dynamics simulations: the 'microMegas' code, in *Mechanics of Nano-Objects*. (Presses de l'Ecole des Mines de Paris, Paris, 2011), pp. 81–100
25. G. deWit, J.S. Koehler, Interaction of dislocations with an applied stress in anisotropic crystals. Phys. Rev. **116**(5), 1113–1120 (1959)
26. J.A. El-Awady, S. Bulent Biner, N.M. Ghoneim, A self-consistent boundary element, parametric dislocation dynamics formulation of plastic flow in finite volumes. J. Mech. Phys. Solids **56**, 2019–2035 (2008)
27. H. Fan, S. Aubry, A. Arsenlis, J.A. El-Awady, The role of twinning deformation on the hardening response of polycrystalline magnesium from discrete dislocation dynamics simulations. Acta Mater. **92**, 126–139 (2015)
28. R.S. Fertig, S.P. Baker, Simulation of dislocations and strength in thin films: a review. Prog. Mater. Sci. **54**, 874–908 (2009)
29. M.C. Fivel, T.J. Gosling, G.R. Canova, Implementing image stresses in a 3D dislocation simulation. Model. Simul. Mater. Sci. Eng. **4**, 581–596 (1996)
30. M.C. Fivel, C.F. Robertson, G.R. Canova, L. Boulanger, Three-dimensional modeling of indent-induced plastic zone at a mesoscale. Acta Mater. **46**, 6183–6194 (1998)
31. A.J.E. Foreman, The bowing of a dislocation segment. Philos. Mag. **15**, 1011–1021 (1967)
32. S. Gao, M. Fivel, A. Ma, A. Hartmaier, Influence of misfit stresses on dislocation glide in single crystal superalloys: a three-dimensional discrete dislocation dynamics study. J. Mech. Phys. Solids **76**, 276–290 (2015)
33. D.J. Gardner, C.S. Woodward, D.R. Reynolds, G. Hommes, S. Aubry, A. Arsenlis, Implicit integration methods for dislocation dynamics. Model. Simul. Mater. Sci. Eng. **23**, 025006 (2015)
34. N.M. Ghoniem, R. Amodeo, Computer simulation of dislocation pattern formation. Solid State Phenom. **3** & **4**, 377 (1988)
35. N.M. Ghoniem, S.H. Tong, L.Z. Sun, Parametric dislocation dynamics: a thermodynamics-based approach to investigations of mesoscopic plastic deformation. Phys. Rev. B **61**, 913 (2000)
36. M.R. Gilbert, S. Queyreau, J. Marian, Stress and temperature dependence of screw dislocation mobility in α-Fe by molecular dynamics. Phys. Rev. B **84**, 174103 (2011)
37. D. Gómez-García, B. Devincre, L.P. Kubin, Dislocation patterns and the similitude principle: 2.5D mesoscale simulations. Phys. Rev. Lett. **96**, 125503 (2006)
38. P.A. Gordon, T. Neeraj, Y. Li, J. Li, Screw dislocation mobility in BCC metals: the role of the compact core on double-kink nucleation. Model. Simul. Mater. Sci. Eng. **18**, 085008 (2010)
39. P.J. Guruprasad, A.A. Benzerga, Size effects under homogeneous deformation of single crystals: A discrete dislocation analysis. J. Mech. Phys. Solids **56**, 132–156 (2008)
40. S.M. Hafez Haghighat, R. Schäublin, D. Raabe, Atomistic simulation of the $a_0 < 100 >$ binary junction formation and its unzipping in body-centered cubic iron. Acta Mater. **64**, 24–32 (2014)
41. A. Hartmaier, M.C. Fivel, G.R. Canova, P. Gumbsch, Image stresses in a free standing thin film. Model. Simul. Mater. Sci. Eng. **7**, 781–793 (1999)
42. J.P. Hirth, J. Lothe, *Theory of Dislocations*, 2nd edn. (Krieger Publishing Company, Malabar, FL, 1992)
43. J. Huang, N.M. Ghoniem, Accuracy and convergence of parametric dislocation dynamics. Model. Simul. Mater. Sci. Eng. **10**, 1–19 (2002)

44. M. Huang, L. Zhao, J. Tong, Discrete dislocation dynamics modelling of mechanical deformation of nickel-based single crystal superalloys. Int. J. Plast. **28**, 141–158 (2010)
45. D. Hull, D.J. Bacon, *Introduction to Dislocations*, 4th edn. (Butterworth Heinemann, Oxford, 2009)
46. A.M. Hussein, S.I. Rao, M.D. Uchic, D.M. Dimiduk, J.A. El-Awady, Microstructurally based cross-slip mechanisms and their effects on dislocation microstructure evolution in fcc crystals. Acta Mater. **85**, 180–190 (2015)
47. K. Kang, J. Yin, W. Cai, Stress dependence of cross slip energy barrier for face-centered cubic nickel. J. Mech. Phys. Solids **62**, 181–193 (2014)
48. Y. Kawasaki, T. Takeuchi, Cell structures in copper single crystals deformed in the [001] and [111] axes. Scr. Met. **14**, 183–188 (1980)
49. T.A. Khraishi, H.M. Zbib, Free-surface effects in 3D dislocation dynamics: formulation and modeling. ASME J. Eng. Mater. Technol. **124**(3), 342–351 (2002)
50. T.A. Khraishi, L. Yan, Y.L. Shen, Dynamic simulations of the interaction between dislocations and dilute particle concentrations in metal–matrix composites (MMCs). Int. J. Plast. **20**, 1039–1057 (2004)
51. U.F. Kocks, A.S. Argon, M.F. Ashby, Thermodynamics and kinetics of slip. Prog. Mater. Sci. **19**, 1–288 (1975)
52. L. Kubin, *Dislocations, Mesoscale Simulations and Plastic Flow*. (Oxford University Press, Oxford, 2013)
53. L.P. Kubin, G. Canova, M. Condat, B. Devincre, V. Pontikis, Y. Bréchet, Dislocation microstructures and plastic flow: a 3D simulation. Solid State Phenom. **23** & **24**, 455–472 (1992)
54. L.P. Kubin, B. Devincre, M. Tang, Mesoscopic modelling and simulation of plasticity in fcc and bcc crystals: dislocation intersections and mobility. J. Comput.-Aided Mater. Des. **5**, 31–54 (1998)
55. W.P. Kuykendall, W. Cai, Conditional convergence in 2-dimensional dislocation dynamics. Model. Simul. Mater. Sci. Eng. **21**, 055003 (2013)
56. W.P. Kuykendall, W. Cai, Effect of multiple stress components on the energy barrier of crossslip in nickel (2016, in preparation)
57. C. Laird, Chapter 27: fatigue, in *Physical Metallurgy*, ed. by R.W. Cahn, P. Haasen, 4th edn. (Elsevier Science, Amsterdam, 1996), pp. 2293–2397
58. S.W. Lee, S. Aubry, W.D. Nix, W. Cai, Dislocation junctions and jogs in a free-standing FCC thin film. Model. Simul. Mater. Sci. Eng. **19**, 025002 (2011)
59. J. Li, AtomEye: an efficient atomistic configuration viewer. Model. Simul. Mater. Sci. Eng. **11**, 173–177 (2003)
60. R. Madec, B. Devincre, L.P. Kubin, From dislocation junctions to forest hardening. Phys. Rev. Lett. **89**, 255508 (2002)
61. R. Madec, B. Devincre, L. Kubin, T. Hoc, D. Rodney, The role of collinear interaction in dislocation-induced hardening. Science **301**, 1879–1882 (2003)
62. J. Marian, A. Caro, Moving dislocations in disordered alloys: connecting continuum and discrete models with atomistic simulations. Phys. Rev. B **74**, 024113 (2006)
63. E. Martínez, J. Marian, A. Arsenlis, M. Victoria, J.M. Perlado, Atomistically informed dislocation dynamics in fcc crystals. J. Mech. Phys. Solids **56**, 869–895 (2008)
64. MDDP: Multiscale Dislocation Dynamics Plasticity code, http://www.cmms.wsu.edu/
65. microMegas (mM) code, http://zig.onera.fr/mm_home_page/
66. R.E. Miller, L.E. Shilkrot, W.A. Curtin, A coupled atomistics and discrete dislocation plasticity simulation of nanoindentation into single crystal thin films. Acta Mater. **52**, 271–284 (2004)
67. MODEL: Mechanics Of Defect Evolution Library, https://bitbucket.org/model/model/wiki/Home
68. V. Mohles, Dislocation dynamics simulations of particle strengthening. in *Continuum Scale Simulation of Engineering Materials: Fundamentals – Microstructures – Process Applications*, ed. by D. Raabe, F. Roters, F. Barlat, L.-Q. Chen (Wiley, Weinheim, 2004)

69. G. Monnet, B. Devincre, Solute friction and forest interaction. Philos. Mag. **86**, 1555–1565 (2006)
70. D. Mordehai, E. Clouet, M. Fivel, M. Verdier, Introducing dislocation climb by bulk diffusion in discrete dislocation dynamics. Philos. Mag. **88**, 899–925 (2008)
71. C. Motz, D. Weygand, J. Senger, P. Gumbsch, Initial dislocation structures in 3-D discrete dislocation dynamics and their influence on microscale plasticity. Acta Mater. **57**, 1744–1754 (2009)
72. A. Needleman, E. van der Giessen, Discrete dislocation and continuum descriptions of plastic flow. Mater. Sci. Eng. A **309–310**, 1–13 (2001)
73. NUMODIS: a Numerical Model for Dislocations, http://www.numodis.com/numodis/index.html
74. R.W. Nunes, J. Bennetto, D. Vanderbilt, Structure, barriers and relaxation mechanisms of kinks in the 90° partial dislocation in silicon. Phys. Rev. Lett. **77**, 1516–1519 (1996)
75. D.L. Olmsted, L.G. Hector Jr., W.A. Curtin, R.J. Clifton, Atomistic simulations of dislocation mobility in Al, Ni and Al/Mg alloys. Model. Simul. Mater. Sci. Eng. **13**, 371–388 (2005)
76. ParaDiS: Parallel Dislocation Simulator code, http://micro.stanford.edu/wiki/ParaDiS_Manuals
77. G. Po, N. Ghoniem, A variational formulation of constrained dislocation dynamics coupled with heat and vacancy diffusion. J. Mech. Phys. Solids **66**, 103–116 (2014)
78. G. Po, M.S. Mohamed, T. Crosby, C. Erel, A. El-Azab, N. Ghoniem, Recent progress in discrete dislocation dynamics and its applications to micro plasticity. J. Mat. **66**, 2108–2120 (2014)
79. W. Püschl, Models for dislocation cross-slip in close-packed crystal structures: a critical review. Prog. Mater. Sci. **47**, 415–461 (2002)
80. S. Queyreau, B. Devincre, Bauschinger effect in precipitation-strengthened materials: A dislocation dynamics investigation. Philos. Mag. Lett. **89**, 419–430 (2009)
81. S. Queyreau, G. Monnet, B. Devincre, Slip systems interactions in α-iron determined by dislocation dynamics simulations. Int. J. Plast. **25**, 361–377 (2009)
82. S. Queyreau, G. Monnet, B. Devincre, Orowan strengthening and forest hardening superposition examined by dislocation dynamics simulations. Acta Mater. **58**, 5586–5595 (2010)
83. S. Queyreau, J. Marian, M.R. Gilbert, B.D. Wirth, Edge dislocation mobilities in bcc Fe obtained by molecular dynamics. Phys. Rev. B **84**, 064106 (2011)
84. S. Rao, T.A. Parthasarathy, D.M. Dimiduk, P.M. Hazzledine, Discrete dislocation simulations of precipitation hardening in superalloys. Philos. Mag. **84**, 30, 3195–3215 (2004)
85. S. Rao, D.M. Dimiduk, J.A. El-Awady, T.A. Parthasarathy, M.D. Uchic, C. Woodward, Activated states for cross-slip at screw dislocation intersections in face-centered cubic nickel and copper via atomistic simulation. Acta Mater. **58**, 5547–5557 (2010)
86. S.I. Rao, D.M. Dimiduk, T.A. Parthasarathy, M.D. Uchic, C. Woodward, Atomistic simulations of surface cross-slip nucleation in face-centered cubic nickel and copper. Acta Mater. **61**, 2500 (2013)
87. S. Rao, D.M. Dimiduk, J.A. El-Awady, T.A. Parthasarathy, M.D. Uchic, C. Woodward, Screw dislocation cross slip at cross-slip plane jogs and screw dipole annihilation in FCC Cu and Ni investigated via atomistic simulations. Acta Mater. **101**, 10–15 (2015)
88. I. Ryu, W.D. Nix, W. Cai, Plasticity of bcc micropillars controlled by competition between dislocation multiplication and depletion. Acta Mater. **61**, 3233–3241 (2013)
89. K.W. Schwarz, Simulation of dislocations on the mesoscopic scale. I. Methods and examples. J. Appl. Phys. **85**, 108–119 (1999)
90. K.W. Schwarz, Local rules for approximating strong dislocation interactions in discrete dislocation dynamics. Model. Simul. Mater. Sci. Eng. **11**, 609–625 (2003)
91. P. Shanthraj, M.A. Zikry, Dislocation density evolution and interactions in crystalline materials. Acta Mater. **59**, 7695–7702 (2011)
92. L.E. Shilkrot, R.E. Miller, W.A. Curtin, Coupled atomistic and discrete dislocation plasticity. Phys. Rev. Lett. **89**(2) 025501 (2002)

93. L.E. Shilkrot, R.E. Miller, W.A. Curtin, Multiscale plasticity modeling: coupled atomistics and discrete dislocation mechanics. J. Mech. Phys. Solids **52**, 755–787 (2004)
94. R.B. Sills, W. Cai, Efficient time integration in dislocation dynamics. Model. Simul. Mater. Sci. Eng. **22**, 025003 (2014)
95. R.B. Sills, W. Cai, Solute drag on perfect and extended dislocations. Philos. Mag. **96**, 895–921 (2016)
96. R.B. Sills, A. Aghaei, W. Cai, Advanced time integration algorithms for dislocation dynamics simulations of work hardening. Model. Simul. Mater. Sci. Eng. **24**, 045019 (2016)
97. A. Takahashi, N.M. Ghoniem, A computational method for dislocation-precipitate interaction. J. Mech. Phys. Solids **56**, 1534–1553 (2008)
98. M. Tang, G. Xu, W. Cai, V.V. Bulatov, A hybrid method for computing forces on curved dislocations intersecting free surfaces in three-dimensional dislocation dynamics. Model. Simul. Mater. Sci. Eng. **14**, 1139–1151 (2006)
99. TRIDIS: the edge-screw 3D Discrete Dislocation Dynamics code, http://www.numodis.com/tridis/index.html
100. E. van der Giessen, A. Needleman, Discrete dislocation plasticity: a simple planar model. Model. Simul. Mater. Sci. Eng. **3**, 689–735 (1995)
101. A. Vattré, B. Devincre, A. Roos, Orientation dependence of plastic deformation in nickel-based single crystal superalloys: discrete-continuous model simulations. Acta Mater. **58**, 1938–1951 (2010)
102. T. Vegge, T. Rasmussen, T. Leffers, O.B. Pedersen, K.W. Jacobsen, Atomistic simulations of cross-slip of jogged screw dislocations in copper. Philos. Mag. Lett. **81**, 137 (2001)
103. M. Verdier, M. Fivel, I. Groma, Mesoscopic scale simulation of dislocation dynamics in fcc metals: Principles and applications. Model. Simul. Mater. Sci. Eng. **6**, 755 (1998)
104. G. Wang, A. Strachan, T. Cagin, W.A. Goddard III, Molecular dynamics simulations of 1/2 $a < 111 >$ screw dislocation in Ta. Mater. Sci. Eng. A **309–310**, 133–137 (2001)
105. C.R. Weinberger, W. Cai, Computing image stress in an elastic cylinder. J. Mech. Phys. Solids **55**, 2027–2054 (2007)
106. C.R. Weinberger, W. Cai, Surface-controlled dislocation multiplication in metal micropillars. Proc. Natl. Acad. Sci. **105**, 38, 14304–14307 (2008)
107. C.R. Weinberger, S. Aubry, S.W. Lee, W.D. Nix, W. Cai, Modelling dislocations in a free-standing thin film. Model. Simul. Mater. Sci. Eng. **17**, 075007 (2009)
108. D. Weygand, L.H. Friedman, E. van der Giessen, A. Needleman, Aspects of boundary-value problem solutions with three-dimensional dislocation dynamics. Model. Simul. Mater. Sci. Eng. **10**, 437–468 (2002)
109. D. Weygand, M. Poignant, P. Gumbsch, O. Kraft, Three-dimensional dislocation dynamics simulation of the influence of sample size on the stress–strain behavior of fcc single-crystalline pillars. Mater. Sci. Eng. A **483**, 188–190 (2008)
110. H. Yasin, H.M. Zbib, M.A. Khaleel, Size and boundary effects in discrete dislocation dynamics: coupling with continuum finite element. Mater. Sci. Eng. A **309–310**, 294–299 (2001)
111. S. Yefimov, I. Groma, E. van der Giessen, A comparison of a statistical-mechanics based plasticity model with discrete dislocation plasticity calculations. J. Mech. Phys. Solids **52**, 279–300 (2004)
112. J. Yin, D.M. Barnett, W. Cai, Efficient computation of forces on dislocation segments in anisotropic elasticity. Model. Simul. Mater. Sci. Eng. **18**, 045013 (2010)
113. C. Zhou, R. LeSar, Dislocation dynamics simulations of plasticity in polycrystalline thin films. Int. J. Plast. **30–31**, 185–201 (2012)
114. X.W. Zhou, R.B. Sills, D.K. Ward, R.A. Karnesky, Atomistic calculation of dislocation core energy in aluminum (2016 in preparation)

http://www.springer.com/978-3-319-33478-3

Multiscale Materials Modeling for Nanomechanics
Weinberger, C.R.; Tucker, G.J. (Eds.)
2016, XV, 547 p. 271 illus., 78 illus. in color., Hardcover
ISBN: 978-3-319-33478-3