Accurate evaluation of dislocation tractions

Daniel Celis-Garza, Edmund Tarleton

E-mail: `edmund.tarleton@materials.ox.ac.uk`

Department of Materials, University of Oxford, Parks Road, OX1 3PH, UK

Abstract. Dislocations generate tractions on the surface of a finite volume. Traditionally the dislocation tractions are integrated numerically over a finite element face to obtain the nodal forces; as required when using superposition to couple discrete dislocation plasticity and the finite element method. Here we implement analytic finite element nodal forces obtained using closed form expressions for the traction integrals. We compare the behaviour of the errors which arise when using numerical integration, provide insight into how and why they occur and give recommendations on avoiding numerical issues when implementing the analytic solution. Evaluating the tractions analytically in a large scale discrete dislocation plasticity simulations of a microcantilever showed a clear difference in the simulated load-displacement curve while the dislocation structure was unchanged.

1. Introduction

Coupling discrete dislocation dynamics (DDD) to the finite element method (FEM) allows micromechanical tests to be simulated [1]. Virtual experiments are essential for interpreting experimental data and relating the measured mechanical response to dislocation mechanisms [2–5]. One of the most popular methods for coupling DDD and FEM is the superposition method [6–8]. The superposition scheme works by decomposing the problem into separate DDD and FE problems as in figure 1. A linear-elastic solid V bounded by a surface S is subjected to traction boundary conditions, $\boldsymbol{T}$, on S_T and displacement boundary conditions, $\boldsymbol{U}$, on S_U. The (˜) fields are those generates by the dislocations in an infinite solid and are obtained by evaluating analytic solutions in a DDD simulation [9]. Formally the dislocation field satisfies

$$\left.\begin{aligned} \nabla \cdot \tilde{\boldsymbol{\sigma}} &= 0 \\ \tilde{\boldsymbol{\sigma}} &= \boldsymbol{C} : \tilde{\boldsymbol{\varepsilon}} \\ \tilde{\boldsymbol{\varepsilon}} &= \tfrac{1}{2}\left(\nabla \tilde{\boldsymbol{u}} + (\nabla \tilde{\boldsymbol{u}})^T\right) \end{aligned}\right\} \text{in V} \tag{1}$$

$$\tilde{\boldsymbol{\sigma}} \cdot \boldsymbol{n} = \tilde{\boldsymbol{T}} \text{ on } S_T \tag{2}$$

$$\left.\begin{aligned} \tilde{\boldsymbol{u}} &= \tilde{\boldsymbol{U}} & t > 0 \\ \tilde{\boldsymbol{u}} &= \boldsymbol{0} & t = 0 \end{aligned}\right\} \text{ on } S_U. \tag{3}$$

As the dislocation fields do not vanish on S the dislocations load the volume by generating tractions, $\tilde{\boldsymbol{T}}$, on S_T and displacements, $\tilde{\boldsymbol{U}}$, on S_U this additional loading

Figure 1: The superposition used to couple DDD and FEM. The volume V is bounded by a surface $S = S_T \cup S_U$ and contains a dislocation ensemble and is subjected to tractions $\boldsymbol{T}$ on S_T and $\boldsymbol{u}$ on S_u. First, the traction, $\tilde{\boldsymbol{T}}$, and displacement, $\tilde{\boldsymbol{U}}$, fields due to the dislocations in the infinite domain (DDD) are evaluated on the boundaries S_T and S_U respectively. Then an elastic boundary value problem can be solved with FEM to calculate the corrective elastic fields required to satisfy the boundary conditions $\hat{\boldsymbol{T}} = \boldsymbol{T} - \tilde{\boldsymbol{T}}$ and $\hat{\boldsymbol{u}} = \boldsymbol{U} - \tilde{\boldsymbol{U}}$.

deforms V generating an additional "image" stress which the dislocations then feel. Therefore corrective (^) fields must be superimposed to satisfy the desired boundary conditions. The corrective field which accounts for both the applied and image stress is obtained numerically by solving the elastic boundary value problem

$$\left.\begin{aligned}
\nabla \cdot \hat{\boldsymbol{\sigma}} &= 0 \\
\hat{\boldsymbol{\sigma}} &= \boldsymbol{C} : \hat{\boldsymbol{\varepsilon}} \\
\hat{\boldsymbol{\varepsilon}} &= \tfrac{1}{2}\left(\nabla\hat{\boldsymbol{u}} + (\nabla\hat{\boldsymbol{u}})^T\right)
\end{aligned}\right\} \text{ in V} \tag{4}$$

$$\hat{\boldsymbol{\sigma}} \cdot \boldsymbol{n} = \boldsymbol{T} - \tilde{\boldsymbol{T}} \text{ on } S_T \tag{5}$$

$$\hat{\boldsymbol{u}} = \boldsymbol{U} - \tilde{\boldsymbol{U}} \text{ on } S_U \tag{6}$$

once the solution to both problems are known, then their superposition solves the desired mixed boundary value problem:

$$\left.\begin{aligned}
\nabla \cdot \boldsymbol{\sigma} &= \nabla \cdot (\hat{\boldsymbol{\sigma}} + \tilde{\boldsymbol{\sigma}}) = 0 \\
\boldsymbol{\sigma} &= \hat{\boldsymbol{\sigma}} + \tilde{\boldsymbol{\sigma}} = \boldsymbol{C} : (\hat{\boldsymbol{\varepsilon}} + \tilde{\boldsymbol{\varepsilon}}) = \boldsymbol{C} : \boldsymbol{\varepsilon} \\
\boldsymbol{\varepsilon} &= \hat{\boldsymbol{\varepsilon}} + \tilde{\boldsymbol{\varepsilon}} = \tfrac{1}{2}\left(\nabla(\hat{\boldsymbol{u}} + \tilde{\boldsymbol{u}}) + (\nabla(\hat{\boldsymbol{u}} + \tilde{\boldsymbol{u}}))^T\right) = \tfrac{1}{2}\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right)
\end{aligned}\right\} \text{ in V} \tag{7}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n} = \boldsymbol{T} \text{ on } S_T \tag{8}$$

$$\left.\begin{aligned}
\boldsymbol{u} &= \boldsymbol{U} & t > 0 \\
\boldsymbol{u} &= \boldsymbol{0} & t = 0
\end{aligned}\right\} \text{ on } S_U \tag{9}$$

The method is not without problems however. As a dislocation segment moves closer to the surface, its (˜) field diverges and starts causing numerical problems [10]. This can be partially solved by using a non-singular formulation such as that proposed by Cai et al. [9] however the gradients in the dislocation field are difficult to accurately

capture when a dislocation approaches S. Another problem with this method is the computational cost when simulating a heterogeneous solid as it requires calculating polarisation stresses due to the difference in the elastic constants in a secondary phase [6, 10, 11]. A modified superposition scheme [12] can overcome this by dividing the problem into separate DDD/FEM problems coupled through 1 elastic FE problem. This requires accurate evaluation of $\tilde{T}$ on the domain boundaries which can only be captured with a fine FE mesh. This in turn will increase the computational cost. In order to simulate polycrystalline or composite materials using superposition requires methods to accurately evaluate both $\tilde{U}$ and $\tilde{T}$. The displacements can be evaluated analytically [13] and this paper investigates the evaluation of the dislocation tractions analytically.

The relative simplicity of the superposition method has made it a popular choice for coupling DDD and FEM since all it requires is the evaluation of FE nodal forces and displacements on the boundary. Furthermore, analytic expressions for the stress field produced by a finite straight dislocation line segment, has allowed Queyreau et al. [14] to use a non-singular formulation to obtain a closed-form solution for the tractions generated by a segment on the surface of a finite element. They used integration by parts giving rise to a number of recursion relations that can be used to construct the full solution from a small set of seed functions.

2. Theory

The force exerted by a dislocation ensemble on a node a belonging to element e is given by,

$$\boldsymbol{F_a} = \int_{S_e} [\tilde{\boldsymbol{\sigma}}(\boldsymbol{x}) \cdot \boldsymbol{n}] \, N_a(\boldsymbol{x}) \, \mathrm{d}S_e \,, \tag{10}$$

where $\mathrm{d}S_e$ is the surface of element e with surface area S_e and N_a is the finite element shape function for node a which is bi-linear in this paper. In the parent element (shown in figure 2) the shape functions are,

$$N_1 = \frac{1}{4}(1 - s_1)(1 - s_2) \tag{11}$$
$$N_2 = \frac{1}{4}(1 + s_1)(1 - s_2)$$
$$N_3 = \frac{1}{4}(1 + s_1)(1 + s_2)$$
$$N_4 = \frac{1}{4}(1 - s_1)(1 + s_2) \,.$$

Usually Gauss quadrature is used to evaluate the surface integral in equation (10) numerically. The 1D Gauss quadrature is,

$$\int_{-1}^{1} f(s) \, \mathrm{d}s \approx \sum_{i=1}^{n} w_i f(s_i) \quad \text{where} \quad w_i = \frac{2}{(1 - s_i^2) \, [P_n'(s_i)]^2}, \tag{12}$$

is the weighting of the Gauss point s_i which is the ith root of P_n; the nth Legendre polynomial normalised so $P_n(1) = 1$. P'_n is the first order derivative of P_n.

The method is very accurate for functions that can be approximated by polynomials, but unsuitable for functions with singularities and near singularities [15, 16]. We use a non-singular formulation for the forces [9], but the singularity is avoided by adding a small cut off radius to account for the dislocation core. However if the integration point falls close to or within the dislocation core, numerical integration can still generate large errors.

For rectangular surface elements we must transform from the parent element in (s_1, s_2) in equation (12) to the real element coordinate system (x, y, z) shown in figure 3. Evaluating equation (10) in the parent element and mapping to the real element gives the force on node a,

$$\boldsymbol{F}_a \approx \sum_{i=1}^{Q} w_i \sum_{j=1}^{Q} w_j [\boldsymbol{\sigma}(r_i, s_j) \cdot \boldsymbol{n}] N_a(r_i, s_j) \det(\mathbf{J}). \tag{13}$$

Where the sum is over the Q quadrature points and $J_{ij} = dx_i/ds_j$ is the element Jacobian defining the transformation from $(s_1, s_2) \mapsto (x, y)$; as the real element is rectangular with surface area S_e, $\det(\mathbf{J}) = S_e/4$. Figure 2 contains examples of how the Gauss points are distributed on the surface.
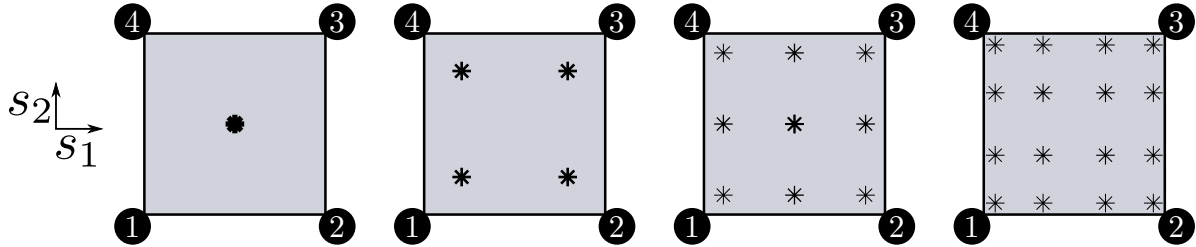

Figure 2: Examples of 2D Gauss-Legendre quadrature of the parent element with $Q = 1, 2, 3, 4$. The point size represents the weight w of the integration point. The parent elements are centred at the origin and $s_1, s_2 \in [-1, 1]$. We use an anticlockwise node numbering scheme.

Equation (10) is actually a triple vector integral. It involves integrating over the surface element and along the dislocation line segment, as shown in figure 3. Using the isotropic Burgers vector distribution proposed in [9], the dyadic form of the stress tensor

Figure 3: Diagram of the parametric line integrals solved by Queyreau et al. [14] to find the forces on linear rectangular surface elements.

produced by a straight finite dislocation segment bounded by nodes at $\mathbf{x_1}$ and $\mathbf{x_2}$ is [14],

$$\boldsymbol{\sigma}(\boldsymbol{x}) = -\frac{\mu}{8\pi} \int\limits_{\mathbf{x_1}}^{\mathbf{x_2}} \left(\frac{2}{R_a^3} + \frac{3a^2}{R_a^5} \right) [(\boldsymbol{R} \times \boldsymbol{b}) \otimes \mathrm{d}\boldsymbol{x}' + \mathrm{d}\boldsymbol{x}' \otimes (\boldsymbol{R} \times \boldsymbol{b})] \tag{14}$$

$$+ \frac{\mu}{4\pi(1-\nu)} \int\limits_{\mathbf{x_1}}^{\mathbf{x_2}} \left(\frac{1}{R_a^3} + \frac{3a^2}{R_a^5} \right) [(\boldsymbol{R} \times \boldsymbol{b}) \cdot \mathrm{d}\boldsymbol{x}'] \, \boldsymbol{I_2}$$

$$- \frac{\mu}{4\pi(1-\nu)} \int\limits_{\mathbf{x_1}}^{\mathbf{x_2}} \frac{1}{R_a^3} [(\boldsymbol{b} \times \mathrm{d}\boldsymbol{x}') \otimes \boldsymbol{R} + \boldsymbol{R} \otimes (\boldsymbol{b} \times \mathrm{d}\boldsymbol{x}')]$$

$$+ \frac{\mu}{4\pi(1-\nu)} \int\limits_{\mathbf{x_1}}^{\mathbf{x_2}} \frac{3}{R_a^5} [(\boldsymbol{R} \times \boldsymbol{b}) \cdot \mathrm{d}\boldsymbol{x}'] \, \boldsymbol{R} \otimes \boldsymbol{R},$$

where

$$\boldsymbol{R} = \boldsymbol{x} - \boldsymbol{x}' = y\boldsymbol{l} + r\boldsymbol{p} + s\boldsymbol{q} \tag{15}$$

$$R_a = \sqrt{\boldsymbol{R} \cdot \boldsymbol{R} + a^2} \tag{16}$$

$$\mathrm{d}\boldsymbol{x}' = -\mathrm{d}y\boldsymbol{l} \tag{17}$$

The vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ are aligned with the edges of the rectangular finite element, $\boldsymbol{n} = \boldsymbol{p} \times \boldsymbol{q}$ is the element surface normal (pointing away from the dislocation), and $\boldsymbol{l}$ is parallel to the dislocation line segment as shown in figure 3. Then (provided $\boldsymbol{l}$ is not parallel to $\boldsymbol{p}$ or $\boldsymbol{q}$) $\boldsymbol{R}$ can be expressed in terms of $(\boldsymbol{l}, \boldsymbol{p}, \boldsymbol{q})$ with coefficients

$$y = \frac{\boldsymbol{R} \cdot \boldsymbol{n}}{\boldsymbol{l} \cdot \boldsymbol{n}}, \quad r = \frac{\boldsymbol{R} \cdot (\boldsymbol{q} \times \boldsymbol{l})}{\boldsymbol{p} \cdot (\boldsymbol{q} \times \boldsymbol{l})}, \quad s = \frac{\boldsymbol{R} \cdot (\boldsymbol{p} \times \boldsymbol{l})}{\boldsymbol{q} \cdot (\boldsymbol{p} \times \boldsymbol{l})} . \tag{18}$$

Substituting equation (14) and equation (11) into equation (10) yields four long and messy equations (one for each FE node) that were elegantly solved by Queyreau et al. [14] by utilising the fact that the triple integrals all had the form,

$$H_{ijkl} = \int\limits_{r_1}^{r_2} \int\limits_{s_1}^{s_2} \int\limits_{y_1}^{y_2} \frac{r^i s^j y^k}{R_a^m} \tag{19}$$

when $m = 5$ then $i, j \in [0, 3], \ k \in [0, 2]$

when $m = 3$ then $i, j \in [0, 2], \ k \in [0, 1]$

when $m = 1$ then $i = j = k = 0$.

Using partial differentiation and integration by parts, they found a series of recurrence relations that lead to double and single integrals with a similar form to equation (equation (19)). Together they are used to construct a full solution. The recurrence relations stop working when $i = j = k = 0$ and $m = 1, 3$. At which point, direct integration of the resulting single and double integrals (the last triple integrals all cancel out in the global calculation) yields six seed functions that are used as the starting point for the recurrence relations. Three of them are logarithms and three either arctangents or—if a discriminant is negative—hyperbolic arctangents. The details of the procedure can be found in [14].

Although exact, the use of arctangents/hyperbolic arctangents and logarithmic functions, compounded by the large number of recurrence relations is prime territory for error propagation and numerical problems (see section 3). This is particularly egregious when using general purpose compilers instead of high-performance or scientific computing compilers where mathematical functions are implemented more precisely. These issues must be taken into account when using analytic tractions in the form of numeric tolerances.

In simulations, the tractions are manifested in the image stresses calculated by the FE solver at FE nodes. In order to validate and compare the practical differences between analytic and gauss quadrature methods, we use them both to calculate the reaction stresses produced by the tractions with the same FE solver and compare the results with the analytic expressions for infinite dislocations in inhomogenous media in [17] for edge dislocations as well as those for screw dislocations found in [18, p. 59, 64], where the traction surface is the line $x = 0$, looking at the xy-plane in the positive z-direction, where the dislocation coordinates are represented by (a, c) and points in the plane described by their (x, y) coordinates.

The original paper by Head [17] has a few typos that have been replicated in other sources, so we include the complete and correct expressions. Head [17] gives two basic cases. The first case in equation (20a) corresponds to the case where b is perpendicular

to the surface and positive b means it points in the positive x direction,

$$\sigma_{xx} = D(y-c)\left\{-\frac{3(x-a)^2+(y-c)^2}{[(x-a)^2+(y-c)^2]^2}+\frac{3(x+a)^2+(y-c)^2}{[(x+a)^2+(y-c)^2]^2}\right. \tag{20a}$$
$$\left.+4ax\frac{3(x+a)^2-(y-c)^2}{[(x+a)^2+(y-c)^2]^3}\right\},$$

$$\sigma_{yy} = D(y-c)\left\{\frac{(x-a)^2-(y-c)^2}{[(x-a)^2+(y-c)^2]^2}-\frac{(x+a)^2-(y-c)^2}{[(x+a)^2+(y-c)^2]^2}\right. \tag{20b}$$
$$\left.+4a(2a-x)\frac{(x+a)^2+(3x+2a)(y-c)^2}{[(x+a)^2+(y-c)^2]^3}\right\},$$

$$\sigma_{xy} = D\left\{(x-a)\frac{(x-a)^2-(y-c)^2}{[(x-a)^2+(y-c)^2]^2}-(x+a)\frac{(x+a)^2-(y-c)^2}{[(x+a)^2+(y-c)^2]^2}\right. \tag{20c}$$
$$\left.+2a\frac{6x(x+a)(y-c)^2-(x-a)(x+a)^3-(y-c)^4}{[(x+a)^2+(y-c)^2]^3}\right\}.$$

The case where the $\boldsymbol{b}$ lies parallel to the surface and positive b means it points in the positive y direction, is found in equation (21a),

$$\sigma_{xx} = D\left\{(x-a)\frac{(x-a)^2-(y-c)^2}{[(x-a)^2+(y-c)^2]^2}-(x+a)\frac{(x+a)^2-(y-c)^2}{[(x+a)^2+(y-c)^2]^2}\right. \tag{21a}$$
$$\left.+2a\frac{(3x+a)(x+a)^3-6x(x+a)(y-c)^2-(y-c)^4}{[(x+a)^2+(y-c)^2]^3}\right\},$$

$$\sigma_{yy} = D\left\{(x-a)\frac{(x-a)^2+3(y-c)^2}{[(x-a)^2+(y-c)^2]^2}-(x+a)\frac{(x+a)^2+3(y-c)^2}{[(x+a)^2+(y-c)^2]^2}\right. \tag{21b}$$
$$\left.-2a\frac{(x-a)(x+a)^3-6x(x+a)(y-c)^2+(y-c)^4}{[(x+a)^2+(y-c)^2]^3}\right\},$$

$$\sigma_{xy} = D(y-c)\left\{\frac{(x-a)^2-(y-c)^2}{[(x-a)^2+(y-c)^2]^2}-\frac{(x+a)^2-(y-c)^2}{[(x+a)^2+(y-c)^2]^2}\right. \tag{21c}$$
$$\left.+4ax\frac{3(x+a)^2-(y-c)^2}{[(x+a)^2+(y-c)^2]^3}\right\}.$$

Screw dislocations are markedly simpler, as only the shear components are non-zero. Here $\boldsymbol{b}=\boldsymbol{l}$ and therefore positive b means it points in the positive z direction (away from the observer) equation (22a),

$$\sigma_{xz} = -D\left(\frac{y-c}{(x-a)^2+(y-c)^2}-\frac{y-c}{(x+a)^2+(y-c)^2}\right) \tag{22a}$$

$$\sigma_{yz} = D\left(\frac{x-a}{(x-a)^2+(y-c)^2}-\frac{x+a}{(x+a)^2+(y-c)^2}\right). \tag{22b}$$

In every case, the constant D is defined by equation (23)

$$D = \frac{\mu}{2\pi}\cdot\frac{1+\nu}{1-\nu^2}\cdot b. \tag{23}$$

Note the first terms of equations (20a), (21a) and (22a) all correspond to the stress field generated by the dislocation. The following terms are the corrective terms required to make the boundary conditions on the surface equal to zero. Therefore, we can split these equations and only look at the real or corrective terms, which we do in order to only visualise the effect of the different traction calculations. Furthermore equations (20a), (21a) and (22a) are all singular at the dislocation coordinates, our simulation code uses non singular expressions found by Cai et al. [9], which smooth out drastic increases in stresses and avoid numerical blow up as we near the dislocation core.

3. Methodology

Numerical integration of tractions can produces unexpected behaviour such as force hot spots and sign inversions as a dislocation approaches a surface. During a large simulation, these effects are hard to spot. Figure 4 has a quick example of the relative errors for an idealised system not disimilar to what can be found with in a simple cantilever bending simulation with a single dislocation loop. As expected, the errors decrease as the mesh gets finer.

Queyreau et al. [14] identified that for a given number of quadrature points, the error is dependent on the dislocation character but always increases rapidly as the distance between the segment and element surface decrease (see figures 10 and 11).

Expanding the test cases reveals just how problematic numerical integration of the

(a) 60×16 elements $(x \times y)$. (b) 120×36 elements $(x \times y)$.

Figure 4: Relative error $(\boldsymbol{F}^\eta)$ in the nodal force obtained using numerical integration with one quadrature point $Q = 1$, and the analytic solution. The xz-face of a rectangular cantilever with plane normal, $\boldsymbol{n} = [0\,\bar{1}\,0]$. The dislocation is of pure edge character with $\boldsymbol{b} = [1\,0\,\bar{1}]$, and line direction, $\boldsymbol{l} = [\bar{1}\,2\,\bar{1}]/\sqrt{6}$ which pierces both xz-faces. The dislocation has its centre at the centroid of the cantilever.

Figure 5: Simple test cases for an edge segment and surface element perpendicular (a) and parallel (b). The perpendicular dislocation is centered at the midpoint of the surface element, node $\mathbf{x_1}$ is separated by a perpendicular distance $\mathbf{x_1}^z$ to prevent the dislocation from intersecting the surface. On the right, the parallel dislocation runs along the x-axis at half the height of the surface element. The nodes of each dislocation line segment are kept at a perpendicular distance of at least one core radius away from the surface element.

tractions can be when a dislocation approaches a surface. The two basic test cases, an orthogonal and parallel edge segment, are shown in figure 5.

The symmetry of the simple test cases can also influence how solution is as the stress fields exhibit symmetries about the dislocation line. If the dislocation is centered at the center of the element and orthogonal to it then the numerical solution becomes more accurate. However isolated test cases are far from typical simulation scenarios. A single quadrature point is often used to evaluate tractions due to the low computational cost and works well for segments which are not close to the surface. This is therefore the benchmark against which we measure the cost-benefit of implementing and utilising the analytic solutions. It should also be noted that by using a single Gauss point, one can use the same FE mesh as for the Gauss points, simplifying the implementation.

Queyreau et al. [14] found the analytic solution is approximately 10 times more computationally expensive than its numerical counterpart for 1 quadrature point. The implementation of the analytic solution is also much more involved. One issue is the calculation of the y-coordinate in the local coordinate frame as shown in figure 3 and equation (18), if $\boldsymbol{l} \perp \boldsymbol{n}$, we get a singularity. As mentioned in [14], an easy fix is to rotate the line segment about its midpoint around the $\boldsymbol{l} \times \boldsymbol{n}$ axis and use the mean values either side as the true result. An example of what this looks like in terms of forces can be seen in figure 6 (we removed the singularity to keep the graph smooth). The specific shape of the curves will vary depending on the element-segment configuration, but is smooth and well-behaved about singularity. For our purposes, we used 4 perturbations each of 1 deg, in the clockwise and anti-clockwise directions (8 rotations in total) w.r.t. the parallel position.

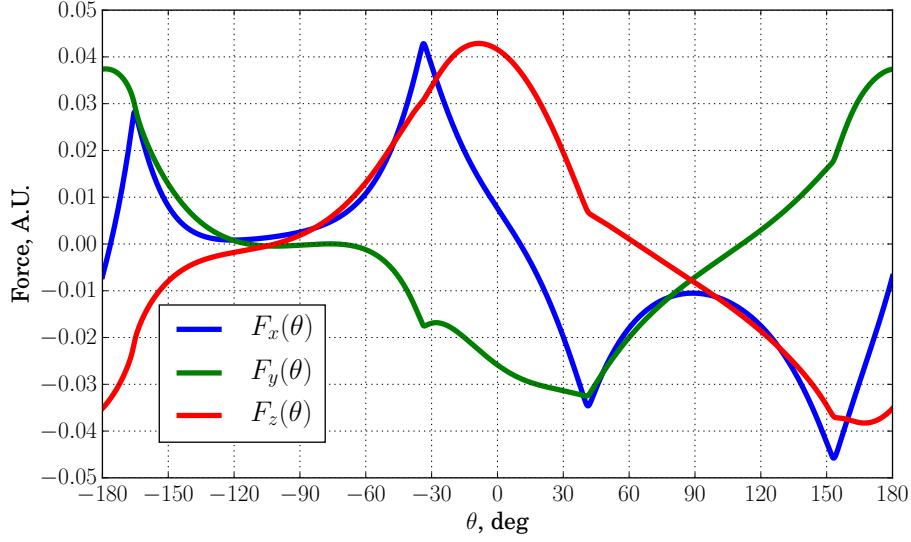However, the check for orthogonality is dependent on the length scales involved in

Figure 6: Example of the components of the total force on a surface element (the force summed over all four FE nodes) as a dislocation segment parallel to the surface element ($\theta = 0$) is rotated about its midpoint around the axis defined by $\boldsymbol{l} \times \boldsymbol{n}$. The force is smooth but not necessarily antisymmetric about the neighbourhood of $\theta = 0$.

the simulation, we found that machine precision plays an important part in deciding how strict the tolerance should be. If,

$$|\boldsymbol{l} \cdot \boldsymbol{n}| \lesssim \frac{\max\left(|\boldsymbol{R} \cdot \boldsymbol{n}|\right)}{10^8}, \tag{24}$$

it is advisable to treat the segment as parallel to the surface element. In our case the numerator on the RHS is simply the cantilever's largest dimension. 10^8 is used instead of actual machine precision $\sim 10^{15}$ because the seed functions and large number of recurrence relations of the solution propagate errors. Smaller tolerances lead to these errors growing so much that the simulation can be impacted; this is a problem when such dislocations move slightly during a simulation and become nearly parallel with the surface, which are detected if the tolerance is too small, yet the limited numerical precision leads to round off and truncation errors during the calculation. Ironically tolerances which are too large can cause the perturbations to rotate the dislocation segment closer to the singularity also producing erroneous results. Larger than necessary tolerances can also slow down the calculation by detecting dislocations that are far enough from the special case that they can be treated like non-parallel segments.

In general one does not want a dislocation segment to intersect the surface when it is being rotated. Naively one would calculate the maximum rotational angle, $\theta_{\max}$, to be,

$$\theta_{\max} = \arctan\left(\frac{2d}{|\mathbf{x_2} - \mathbf{x_1}|}\right), \tag{25}$$

(a) Node numbering for our FE voxels.

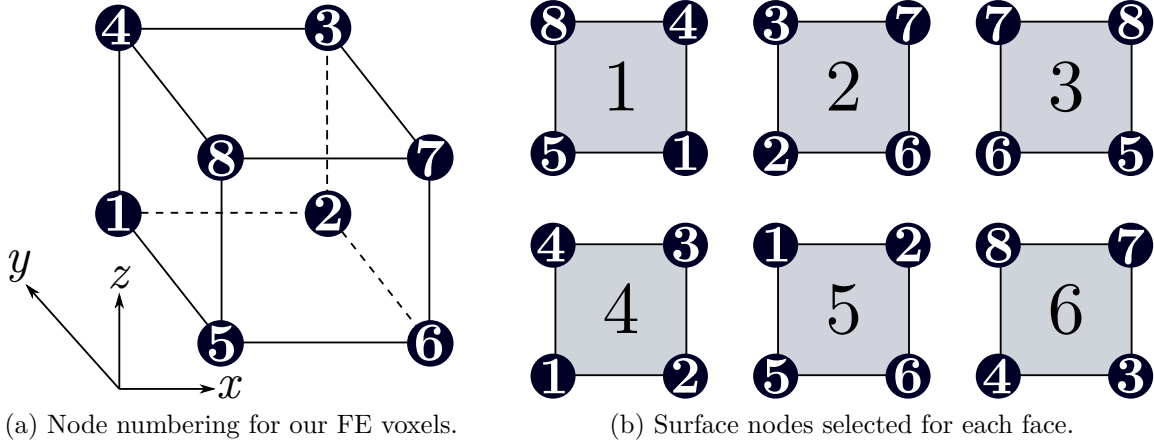(b) Surface nodes selected for each face.

Figure 7: Surface element nodes must be self-consistent so the forces have the correct sign.

where d is the minimum orthogonal distance from the dislocation to the surface element (collision distance), and $\mathbf{x_1}$, $\mathbf{x_2}$ are the dislocation segment node coordinates. However, θ_{max} might be too small (due to numerical precision) in cases when the segment length is too small compared to the distance to a surface element or when the segment length is much greater than d. Fine tuning the angle is a task that involves knowing the minimum collision distance, minimum segment length, dislocation core radius, and the compiler's implementation of mathematical functions. Given the rarity of such cases and their comparatively low impact, we chose our perturbation such that we safely avoid this problem.

Furthermore, the chirality and self-consistency of the FE nodes must be accounted for such that they are in the proper order regardless of the element face they belong to. Here we use 8 node linear hexahedral (brick) elements with the node numbering shown in figure 7a. The node ordering for the various surfaces are obtained by placing an observer inside the element and noting the nodes for the faces of the element, these are shown in figure 7b. The total force on a given node must include the force contributions from every element in which said node appears, see figure 8. The specifics of the mapping depend on the global FE node numbering. Using figure 8 as our reference labels for elements and nodes, e and n respectively, we can give a concrete example of how this is
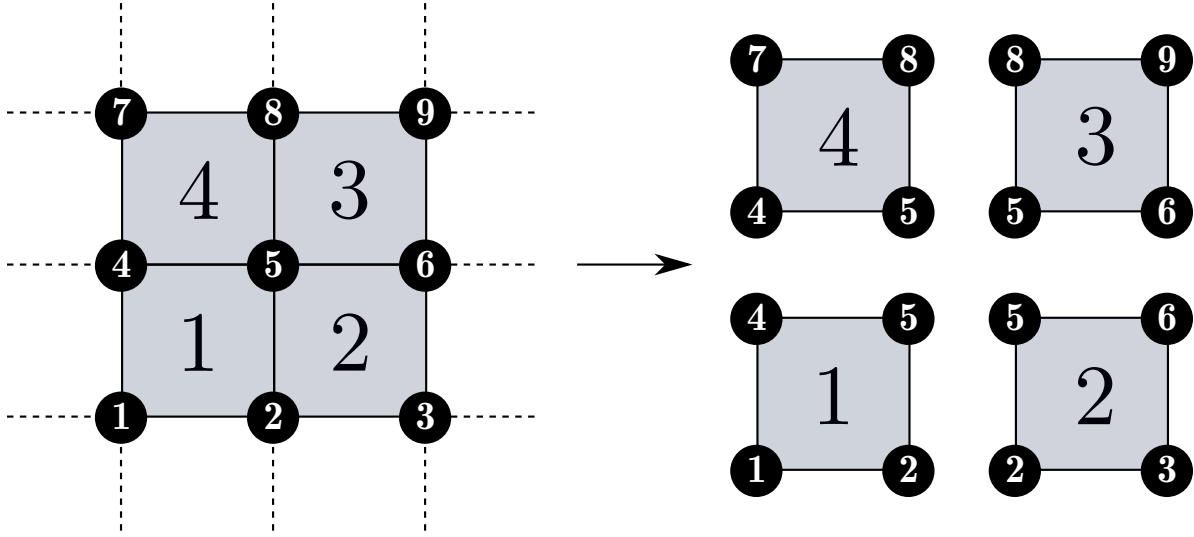
Figure 8: FE nodes are shared by either 4 element faces or 3 if it is a corner node. The total force on a given node is the summation of the force contributions from each element it belongs to.

done by defining,

$$\boldsymbol{x_{e,n}} \equiv \begin{bmatrix} x_{e,n} & y_{e,n} & z_{e,n} \end{bmatrix}^{\mathsf{T}}, \quad \boldsymbol{x_n} \equiv \begin{bmatrix} x_n & y_n & z_n \end{bmatrix}^{\mathsf{T}} \tag{26a}$$

$$\mathbf{N_L} = \begin{bmatrix} l_{1,1} & l_{1,2} & l_{1,4} & l_{1,5} \\ l_{2,2} & l_{2,3} & l_{2,5} & l_{2,6} \\ l_{3,5} & l_{3,6} & l_{3,8} & l_{3,9} \\ l_{4,4} & l_{4,5} & l_{4,7} & l_{4,8} \end{bmatrix}, \quad \boldsymbol{\gamma} = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_9 \end{bmatrix} \tag{26b}$$

$$\mathbf{F_e} = \begin{bmatrix} \boldsymbol{x_{1,1}} & \boldsymbol{x_{1,2}} & \boldsymbol{x_{1,4}} & \boldsymbol{x_{1,5}} \\ \boldsymbol{x_{2,2}} & \boldsymbol{x_{2,3}} & \boldsymbol{x_{2,5}} & \boldsymbol{x_{2,6}} \\ \boldsymbol{x_{3,5}} & \boldsymbol{x_{3,6}} & \boldsymbol{x_{3,8}} & \boldsymbol{x_{3,9}} \\ \boldsymbol{x_{4,4}} & \boldsymbol{x_{4,5}} & \boldsymbol{x_{4,7}} & \boldsymbol{x_{4,8}} \end{bmatrix}, \quad \tilde{\boldsymbol{F}} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \vdots \\ \boldsymbol{x_9} \end{bmatrix}. \tag{26c}$$

Where $\boldsymbol{x_{e,n}}$ is a 3×1 column vector corresponding to the x, y, z dislocation induced forces on node n on the surface element e. There are four of these per rectangular surface element, where a given node, n, can appear in multiple surface elements (e.g. node 5 in figure 8 is shared by all 4 surface elements), all of which independently contribute to the total force on said node. $\boldsymbol{x_n}$ is a 3×1 column vector corresponding to the total x, y, z dislocation induced forces on node n. These are used to shorten the definition of equation (26c) and are not explicitly defined in the implementation, rather they give the force matrices $\mathbf{F_e}$ and $\tilde{\boldsymbol{F}}$ a specific row order. $\mathbf{N_L}$ is crucial for the correct implementation of this analytical solution in traditional FE codes. It is the $E \times 4$ matrix corresponding to the global label of each node in a given surface element. Each row of the matrix represents a surface element and each column represents a node in the surface element. We cannot naïvely add the columns together as that would give the

total force acting on the element as a whole, not each FE node individually. We chose to arrange the columns in accordance to figure 3 as it makes it easier to implement the solution, but the only thing that matters is that the basis vectors $\boldsymbol{n}$, $\boldsymbol{p}$, $\boldsymbol{q}$ are calculated appropriately. $\boldsymbol{\gamma}$ is the vector with the FE node labels, which makes mapping force to node possible. $\mathbf{F_e}$ is the $3E \times 4$ matrix where the forces acting on each of the four nodes (column) in a particular surface element (each element corresponds to three consecutive rows because there are three dimensions) are stored. $\tilde{\boldsymbol{F}}$ is the $3N \times 1$ column vector where the total forces on each node are stored (each node has three rows because there are three dimensions). This is easily generalisable to E elements and N nodes.

The following algorithm illustrates how the total force on each node is obtained. The resulting force vector is later used in equation (5) as $\tilde{\boldsymbol{T}} \equiv \tilde{\boldsymbol{F}}$. Our implementation

---

Algorithm 1 Assuming $\tilde{\boldsymbol{F}}$ is arranged the same way as $\boldsymbol{\gamma}$ and indexing starts at 0.

1: ▷ Loop through the array containing the node labels of the relevant surface nodes.
2: **for** $i = 0$; $i < \text{length}(\boldsymbol{\gamma})$; $i + +$ **do**
3:                       ▷ Save the global node label for the current iteration.
4:     $n \leftarrow \boldsymbol{\gamma}[i]$
5:     ▷ Use the node label to find a vector, $\boldsymbol{L}$, with the linearised indices in $\mathbf{N_L}$ where node n appears as part of a surface element whose tractions we are calculating.
6:     $\boldsymbol{L} \leftarrow \text{find}(\mathbf{N_L} == n)$
7:                                     ▷ Loop over coordinates.
8:     **for** $k = 0$; $k < 3$; $k + +$ **do**
9:                                 ▷ Use global node label vector to index the force array from the analytical force calculation. Multiplied by 3 because there are three coordinates per node. We sum the forces from the analytical calculation because the same global node can be part of multiple surface elements. We add k because the x, y, z coordinates are consecutively stored in $\mathbf{F_e}$.
10:         $\tilde{\boldsymbol{F}}[3n + k] \leftarrow \tilde{\boldsymbol{F}}[3n + k] + \sum \mathbf{F_e}[3\boldsymbol{L} + k]$
11:     **end for**
12: **end for**

---

does not strictly follow algorithm 1 because we memoise a generalised version of $\boldsymbol{L}$ upon simulation initialisation instead of finding one at every iteration, reducing computational time but requiring us to account for nodes without traction boundary conditions. Our indexing also starts at 1, but zero indexing makes the algorithm easier to follow.

After the tractions have been calculated, they are used by the FE solver to generate the corresponding image stresses. Figure 9 shows the system we used to compare the image stresses calculated by our FE solver using numeric tractions v.s. analytic tractions v.s. infinite-domain, singular image stresses found in equations (20a), (21a) and (22a).

Finally, we ran a simple simulation where the differences between methods can be readily observed.
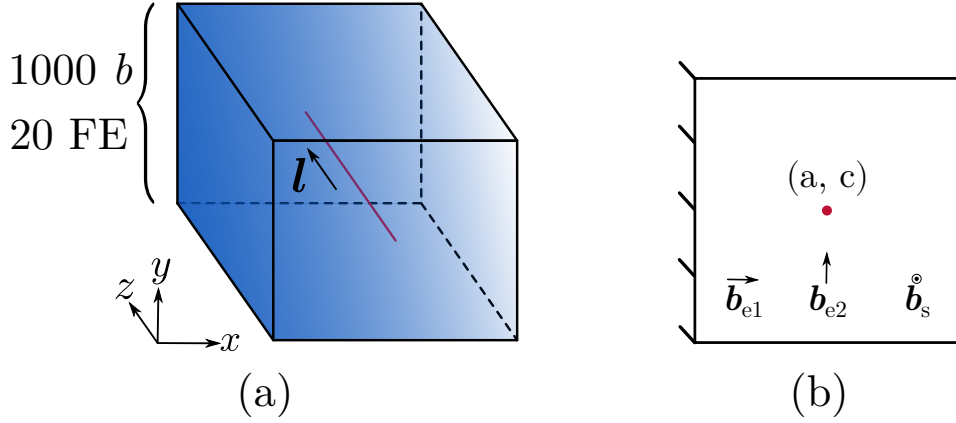
Figure 9: Dislocation parallel to a surface described by the line $x = 0$ (the yz-plane in 3D), where the dislocation coordinates are (a, c), looking down the xy-plane where the x-axis is horizontal and the y-axis vertical. (a) describes the box we used for our comparison, a $20 \times 20 \times 20$ element cubic box with side lengths equal to 1000 b with $\boldsymbol{T} = \boldsymbol{0}$ traction boundary conditions only on the nodes occupying the yz-plane and $\boldsymbol{U} = \boldsymbol{0}$ displacement conditions everywhere else, and a dislocation with line direction in the positive z direction. (b) is the 2D view with the dislocation coordinates (a, c), as well as the two edge Burgers vectors as described by [17], $\boldsymbol{b}_{e1}$ perpendicular to the surface and pointing in the positive x direction, $\boldsymbol{b}_{e2}$ parallel to the surface and pointing in the positive y direction; and the screw Burgers vector parallel to the surface and pointing in the line direction as posited in [18].

4. Results

Isolated test cases were useful in order to gain insight into the differences between the numerical and analytical approaches. When the stress field on the surface element is highly symmetric numerical integration works well; showing good convergence with increasing number of quadrature points. Figure 10 shows that even for dislocations only one dislocation core radius ($5b$) away from the surface element, the force can be obtained, up to numerical precision, with 1000 quadrature points, for all segment lengths tested. However, as stated in section 2, the limitations of Gauss quadrature are also evident. Figures 10 and 11 showcase the problem with numerical integration to compute integrals of rational functions close to a pole. The comparatively large relative errors with $Q = 11$, compared to fewer quadrature points, as the dislocation nears the element face makes it hard to globally minimise the error by picking a specific number of quadrature points while maintaining a feasible computational cost.

Gauss quadrature fails to capture the $1/R_a$ nature of the stress tensor, especially when a segment is close to a heavily weighted quadrature point. When the dislocation node $\mathbf{x_1}$ is close to the surface element (see figure 5), 11 Gauss points ($Q = 11$) performs worse than with $Q = 2$, as shown in figure 10 (a) and (b). This is because When Q is odd a Gauss point is at the element center, as shown in Figure 2, directly in front of

(a) Segment starts a single dislocation core radius away.

(b) Segment starts 3 dislocation core radii away.

(c) Segment starts 201 dislocation core radii away.

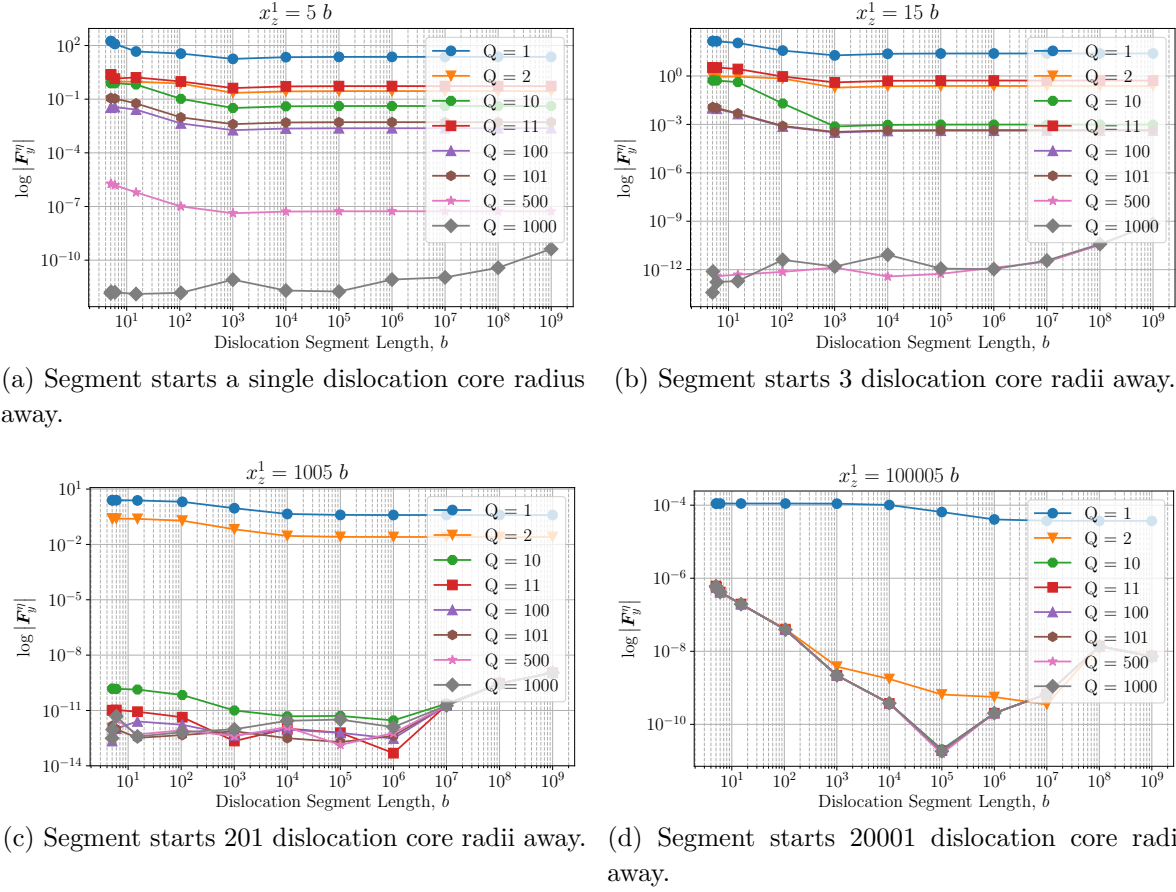(d) Segment starts 20001 dislocation core radii away.

Figure 10: Log-log plot of the relative error as a function of dislocation segment length for a perpendicular edge dislocation (figure 5a). x_z^1 is the z-coordinate of node $\mathbf{x_1}$. $\boldsymbol{b} = [100]$, line direction, $\boldsymbol{l} = [001]$, and dislocation core radius, $a = 5b$, the surface element's normal and size are, $\boldsymbol{n} = [001]$, $L = 1000b$, respectively. Q is the number of quadrature points per dimension.

the dislocation segment.

The limitations of Gauss quadrature become evident when the dislocation line segment is parallel to a surface element. Figure 11 shows the relative errors for a parallel edge dislocation. In this configuration the forces due to a screw dislocations vanish as the stress field is antisymmetric. In figure 11, we observe the relative errors decrease and then converge to ~ 1 as the distance grows larger. The convergence to ~ 1 is an artifact arising from the fact that the forces on the surface element get asymptotically closer to zero with increasing distance from the surface element, which results in the division of two increasingly similar, small, non-zero numbers, resulting in ~ 1 when done with limited precision. Given infinite precision, it is not unreasonable to assume the relative error would converge to $\sim 10^{-3}, \ \sim 10^{-4}$. The absolute error *always* decreases as the distance from the surface element increases, but absolute errors are not as useful tools of comparison. However, when the segment is close to the surface, the relative errors are
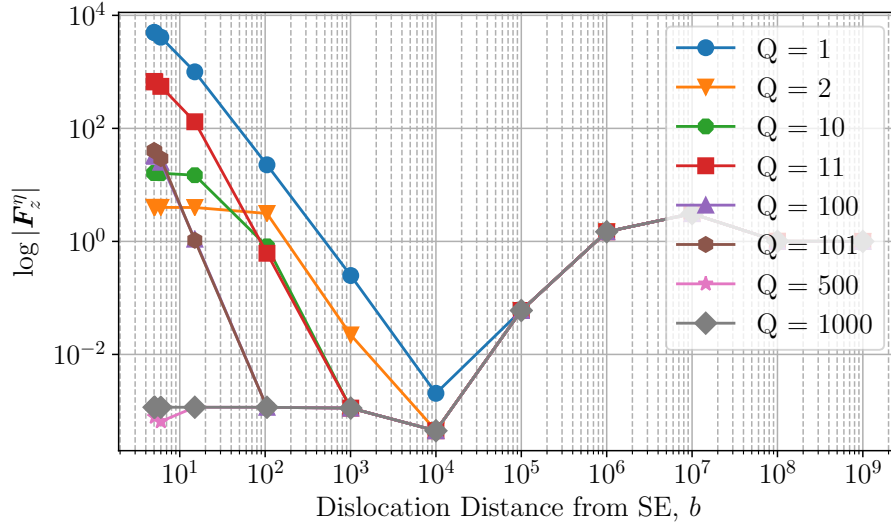
Figure 11: Log-log plot of the relative error as a function of distance from the surface element for a parallel edge dislocation (figure 5b). Burgers vector, $\boldsymbol{b} = [010]$, line direction, $\boldsymbol{l} = [100]$, dislocation core and surface element parameters are the same as figure 10. The dislocation length is fixed to $10^6 \, b$, $x \in [-0.5 \times 10^6, \, 0.5 \times 10^6]$ and bisects the surface element along the $[100]$ direction. The whole dislocation was segmented into 10^4 pieces of length $100 \, b$ to prevent the dislocation from intersecting the surface element when they were rotated to avoid the singularity. The relative error at large distances $(> 10^6 b)$ converges to one due to artifacting when dividing two very small double precision numbers. The absolute error always goes towards zero with increasing distance from the surface.

large $(\gg 1)$ unless using a large number of quadrature points, for example figure 5(a) shows that $Q \geq 100$ are required when the distance $x_z^1 = 15b$.

For a segment parallel to a surface, Gauss quadrature performs far worse than for a perpendicular segment, as shown in figure 11. This is because numerical quadrature is only accurate for integrating functions that can be approximated by polynomials. Whereas the dislocation stress field [9] decays as $\mathcal{O}(1/R)$. This makes Gauss quadrature unsuitable as equation (13) fails to accurately capture the stress, which varies rapidly across the element surface for small R. The relevant components of the stress tensor for a perpendicular edge segment which is not penetrating the surface are shown in figure 12. The σ_{xz} and σ_{zz} components are antisymmetric and the resultant forces vanish which is captured when numerically integrating with small Q, while σ_{yz} does not but can be accurately integrated with sufficiently large Q.

It is evident that symmetry plays a role in the accuracy of numerical integration. If a stress field has the appropriate symmetry, as in figure 12, Gauss quadrature will be accurate given enough quadrature points. However, when the segment is translated off-centre, this breaks the symmetry and the symmetric weighting of the Gauss quadrature

(a) σ_{xz}.
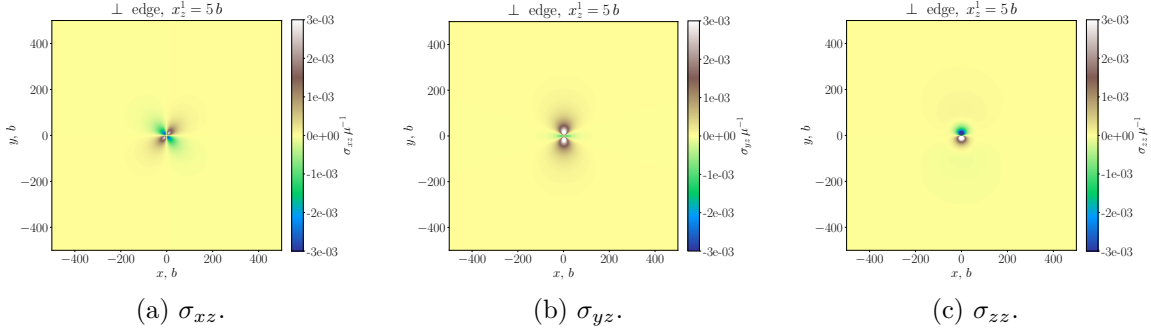
(b) σ_{yz}.

(c) σ_{zz}.

Figure 12: Relevant stress components to figure 10. The antisymmetry of σ_{xz} and σ_{zz} makes them vanish when integrating. σ_{yz} does not vanish, but its symmetry means Gauss quadrature works if sufficient quadrature points are used.

scheme fails to accurately integrate the stress fields when the segment is close to the surface. The latter case is of course much more common in a simulation than the former, so in reality the errors between analytical and numerical solutions are larger than what is led on by these contrived cases. The error convergence with higher numbers of quadrature points is consistent with the findings of [14].

The isolated tractions are only part of the story however, as they are used to calculate the image forces as a result of the boundary. We therefore also compare image forces calculated from numeric ($Q = 1$) tractions and analytic traction calculations affect the image forces to see the extent to which higher accuracy tractions affect the FE part of the simulation. To do so we compare the traction + FEM calculation to analytic solutions by Head [17].

In a DDP simulation it is rare to find a dislocation segment so close to a quadrature point that the errors spike so dramatically. However, numerical integration with small Q produces noticeable errors over a wide range of distances and segment lengths. Simulations usually contain a large number of dislocations which are parallel or near-parallel and close to surface elements. Figure 11 shows it is not possible to accurately compute the tractions and nodal forces using Gauss quadrature even with a large number of quadrature points.

It is worth noting that rotational perturbation required for the analytic solution to be non-singular when a dislocation segment is parallel to a surface element can introduce small erroneous force components if the rotation is too large. If the rotational perturbations were small enough and floating point arithmetic exact, these erroneous forces would cancel when averaged over two opposite rotations. We found that a maximum rotation angle of $\pm 0.5°$ makes these erroneous force components negligible.

The purpose of the cantilever simulation showed herein is not to be have an exactly physical when it comes to dislocation structure. Our initial source sizes and structure are somewhat unphysical, i.e. square sources with segment lengths on the order of the FE mesh courseness. There are also known issues with junction formation when multiple
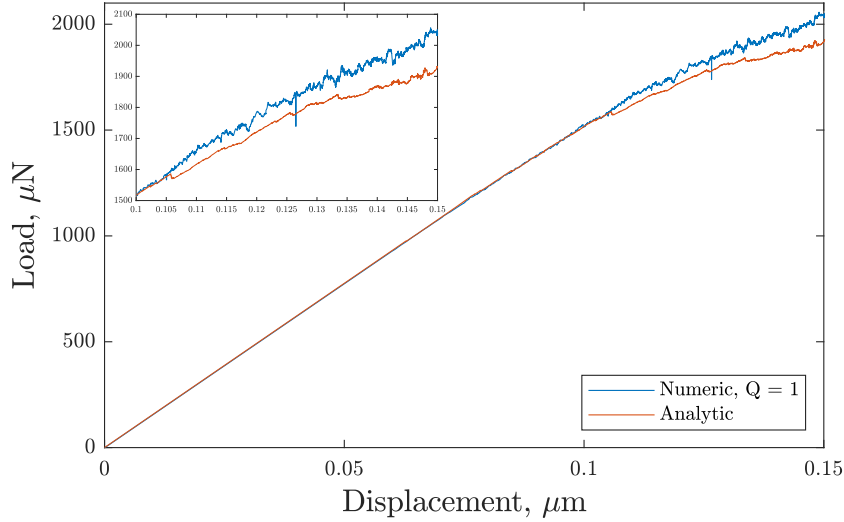
Figure 13: Load-displacement curve. When the tractions are evaluated analytically the curve is less noisy compared to numerical integration. As expected the yield point remains the same in both cases.

active slip systems are used in a single simulation. However, we are investigating not only the results but also the robustness and numerical stability of a fully integrated implementation of analytical solutions to dislocation tractions. By throwing the worst case scenario at both numerical and analytic traction calculations, we ensure both numerical and analytic solutions work even when the multiscale nature of the problem causes serious floating point arithmetic errors.

A full discrete dislocation plasticity simulation of a microcantilever bend test demonstrates the benefits of using the analytic solution over numerical integration. The FE nodal forces were evaluated using Gauss quadrature with a single integration point for one simulation, and analytical tractions for the other. The simulations were otherwise identical with a finite element size of 0.25μm and sixty initial sources equally distributed on the 12 $\{1\bar{1}0\}\langle111\rangle$ bcc slip systems randomly distributed inside the region occuping the central 70% of the cantilever ensuring the simulation proceeded at an acceptable rate and avoiding dislocations from exiting the fixed end, which can lead to problems due to the necessary FEM boundary conditions. The beam was fixed at $x = 0$ and displaced by $U_z = -0.15$ μm at $x = L = 15$ μm. The material parameters were those of bcc tungsten, $\mu = 145$ GPa and $\nu = 0.28$. The analytic solution significantly reduced the noise in the load displacement curve as shown in figure 13.

The inset in figure 13 shows the yield point remains the same, but the reaction forces diverge from each other as the simulation progresses. It should be noted that the dislocation structure was unchanged and so the change in the mechanical response is due entirely to the accuracy of the force calculation. The fluctuations are due to dislocations approaching a free surface and generating a large traction which is suddenly removed when the segment exits. These events are difficult to capture accurately when using
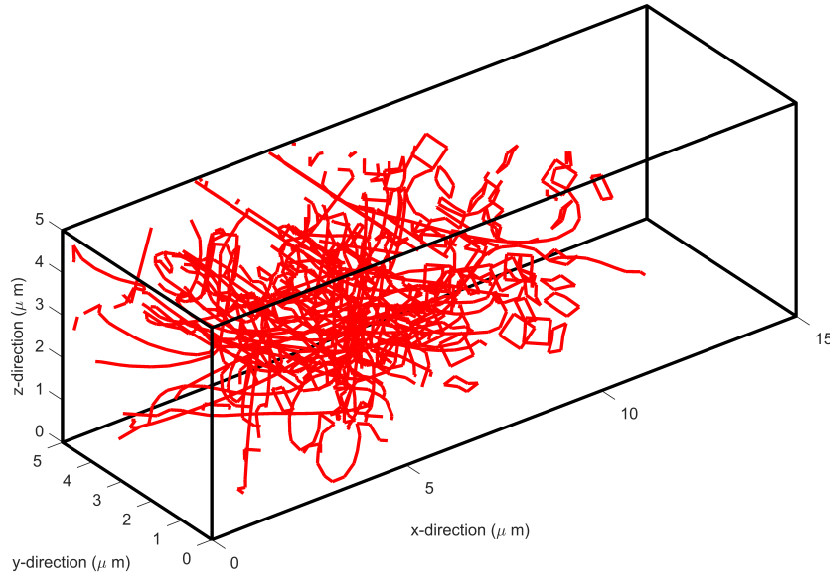
Figure 14: Final dislocation structure using analytic tractions.

numerical integration.

The dislocation structure did not change much in either case. In fact, when looking at it from the zoom level of figure 14, they both appear identical. There are however, many small differences in node position. Surprisingly, depending on the specific simulation, we have found between $10--20\%$ less dislocation segments when the analytic solution is used. Meaning the numerical solution systematically underestimates the average contribution of every single dislocation segment, with occasional massive spikes in force when a dislocation nears a gauss point. We expect these differences to get larger as simulation runtime increases. This is a fundamental and practically unavoidable problem of the numerical solution.

There are a few things that determine whether the numerical solutions are good enough for a given problem. When primarily investigating larger scale features of a dislocation structure—such as larger scale investigation of dislocation forresting around inclusions—Gauss quadrature is quite suitable. However, if one is interested in investigating the finer details of a dislocation structure—such as the nuanced interactions with closly packed inclusions—the analytic method is best. Furthermore, when investigating load-displacement curves, the analytical solutions provide a much better description for a relatively small increase in computational cost [14]. That being said, the cost is nowhere near rate limiting, segment-segment forces and topological changes quickly become much more expensive as they both have worse asymptotic scaling, and topological changes also quickly become memory bound. An accurate description of load-displacement curves should also implement the exact displacement boundary conditions obtained by Bromage and Tarleton [13].

All our subsequent simulations have used analytic tractions as they run faster and yield more reasonable dislocation structures, particularly for more complex simulations.

5. Acknowledgements

References

[1] S. Groh and H. M. Zbib. Advances in Discrete Dislocations Dynamics and Multiscale Modeling. *Journal of Engineering Materials and Technology*, 131(4): 041209, 2009. ISSN 00944289. doi: 10.1115/1.3183783.

[2] M Verdier, M Fivel, and I Groma. Mesoscopic scale simulation of dislocation dynamics in fcc metals: Principles and applications. *Modelling and Simulation in Materials Science and Engineering*, 6(6):755, 1998.

[3] C. Déprés, C. F. Robertson, and M. C. Fivel. Low-strain fatigue in 316l steel surface grains: a three dimension discrete dislocation dynamics modelling of the early cycles. part 2: Persistent slip markings and micro-crack nucleation. *Philosophical Magazine*, 86(1):79–97, 2006. doi: 10.1080/14786430500341250.

[4] E Tarleton, DS Balint, J Gong, and AJ Wilkinson. A discrete dislocation plasticity study of the micro-cantilever size effect. *Acta Materialia*, 88:271–282, 2015.

[5] Haiyang Yu, Alan Cocks, and Edmund Tarleton. Discrete dislocation plasticity helps understand hydrogen effects in bcc materials. *Journal of the Mechanics and Physics of Solids*, 2018. ISSN 0022-5096. doi: https://doi.org/10.1016/j.jmps.2018. 08.020.

[6] Erik Van der Giessen and Alan Needleman. Discrete dislocation plasticity: a simple planar model. *Modelling and Simulation in Materials Science and Engineering*, 3 (5):689, 1995.

[7] MC Fivel and GR Canova. Developing rigorous boundary conditions to simulations of discrete dislocation dynamics. *Modelling and Simulation in Materials Science and Engineering*, 7(5):753, 1999.

[8] Akiyuki Takahashi and Nasr M Ghoniem. A computational method for dislocation–precipitate interaction. *Journal of the Mechanics and Physics of Solids*, 56(4): 1534–1553, 2008.

[9] Wei Cai, Athanasios Arsenlis, Christopher R Weinberger, and Vasily V Bulatov. A non-singular continuum theory of dislocations. *Journal of the Mechanics and Physics of Solids*, 54(3):561–587, 2006.

[10] B Devincre, A Roos, and S Groh. Boundary problems in dd simulations. In *In: Thermodynamics, Microstructures and Plasticity, A. Finel et al., Nato Sciences Series II: Mathematics, Physics and Chemistry, 108, p. 275, Eds (Kluwer, NL-Dordrecht.* Citeseer, 2003.

[11] CS Shin, MC Fivel, and KH Oh. Nucleation and propagation of dislocations near a precipitate using 3d discrete dislocation dynamics simulations. *Le Journal de Physique IV*, 11(PR5):Pr5–27, 2001.

[12] M. P. O'Day and W. a. Curtin. A Superposition Framework for Discrete Dislocation Plasticity. *Journal of Applied Mechanics*, 71(November 2004):805, 2004. ISSN 00218936. doi: 10.1115/1.1794167.

[13] B Bromage and E Tarleton. Calculating dislocation displacements on the surface of a volume. *Modelling and Simulation in Materials Science and Engineering*, 26 (8):085007, 2018.

[14] S Queyreau, J Marian, BD Wirth, and A Arsenlis. Analytical integration of the forces induced by dislocations on a surface element. *Modelling and Simulation in Materials Science and Engineering*, 22(3):035004, 2014.

[15] Gene H Golub and John H Welsch. Calculation of gauss quadrature rules. *Mathematics of computation*, 23(106):221–230, 1969.

[16] Michael A Khayat and Donald R Wilton. Numerical evaluation of singular and near-singular potential integrals. *IEEE Transactions on Antennas and Propagation*, 53 (10):3180–3190, 2005.

[17] AK Head. Edge dislocations in inhomogeneous media. *Proceedings of the Physical Society. Section B*, 66(9):793, 1953.

[18] John Price Hirth, Jens Lothe, and T Mura. Theory of dislocations, 1983.