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. Comparisons to infinite-domain solutions and a very simple unloaded simulation shows the drastic differences between the two methods.

1. Introduction

Coupling discrete dislocation dynamics (DDD) to the finite element method (FEM) allows micromechanical tests to be simulated with explicit dislocation interactions [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 and simplest methods for coupling DDD and FEM is the superposition method [6–8], which works by decomposing the problem into separate DDD and FE problems as illustrated in figure 1.

Specifically, 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 generated by the dislocations in an infinite solid and in our case are obtained by evaluating analytic solutions in a DDD simulation [9]. Formally, the dislocation field satisfies,

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

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

$$\left. \begin{aligned} \tilde{\boldsymbol{u}} &= U, & t > 0 \\ \tilde{\boldsymbol{u}} &= \boldsymbol{0}, & t = 0 \end{aligned} \right\} \quad \text{on } S_U . \tag{3}$$
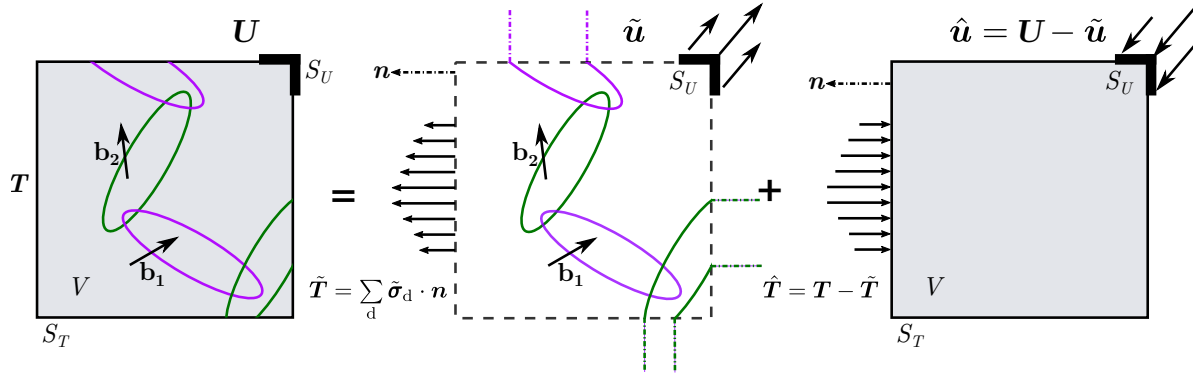
Figure 1: The superposition used to couple DDD and FEM. The volume V is bounded by a surface $S = S_T \cup S_U$ and contains a dislocation ensemble and is subjected to tractions $\boldsymbol{T}$ on S_T and $\boldsymbol{u}$ on S_u. First, the traction, $\tilde{\boldsymbol{T}}$, and displacement, $\tilde{\boldsymbol{u}}$, 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 $\hat{\boldsymbol{u}} = \boldsymbol{U} - \tilde{\boldsymbol{U}}$.

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 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,

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

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

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

Once the solutions to both problems are known, their superposition solves the desired mixed boundary value problem,

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

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

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

For all its simplicity and elegance, the method is not without issue. As the distance between dislocation and surface decreases, $\tilde{\boldsymbol{\sigma}}$ diverges [10]. This can be

partially solved by using a non-singular formulation for $\tilde{\boldsymbol{\sigma}}$, such as that proposed by Cai et al. [9]. Regardless, the steep gradients in the dislocation field are difficult to accurately capture as the dislocation approaches S. Another problem with this method is the large computational cost when simulating a heterogeneous solid, as this requires calculating polarisation stresses due to the difference in the elastic constants between phases [6, 10, 11].

A modified superposition scheme [12] can overcome this by dividing the problem into separate DDD-FEM problems coupled through a elastic FE problem. This requires accurate evaluation of $\tilde{\boldsymbol{T}}$ on the domain boundaries—which can only be captured with a fine FE mesh—increasing the computational cost. Therefore, simulating polycrystalline or composite materials using superposition necesitates methods to accurately evaluate both $\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{T}}$. The displacements can be evaluated analytically as shown by [13] and this paper investigates the evaluation of the dislocation tractions analytically.

As previously mentioned, the relative simplicity of the superposition method has made it a popular choice for coupling DDD and FEM as all it needs is the evaluation of FE nodal forces and displacements on the boundary. Furthermore, the analytic expression for the stress field produced by a finite, straight dislocation line segment has allowed Queyreau et al. [14] to use the non-singular formulation [9] to obtain closed-form solutions for the tractions generated by the segment on the surface of a finite element.

2. Theory

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

$$\boldsymbol{F}_a = \int_{S_e} \left[\tilde{\boldsymbol{\sigma}}(\boldsymbol{x}) \cdot \boldsymbol{n} \right] N_a(\boldsymbol{x}) \, \mathrm{d}S_e \,. \tag{10}$$

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. The solutions are for linear rectangular surface elements and as such the shape functions are,

$$N_1 = \frac{1}{4}(1 - s_1)(1 - s_2) \tag{11}$$

$$N_2 = \frac{1}{4}(1 + s_1)(1 - s_2)$$

$$N_3 = \frac{1}{4}(1 + s_1)(1 + s_2)$$

$$N_4 = \frac{1}{4}(1 - s_1)(1 + s_2) \,.$$

Where s_1 and s_2 are the orthogonal coordinates local to the element.

Gauss quadrature is usually used to numerically evaluate the surface integral in equation (10). In 1D this is,

$$\int_{-1}^{1} f(s) \, \mathrm{d}s \approx \sum_{i=1}^{n} w_i f(s_i) \quad \text{where} \quad w_i = \frac{2}{(1 - s_i^2) \left[P_n'(s_i) \right]^2} \,, \tag{12}$$
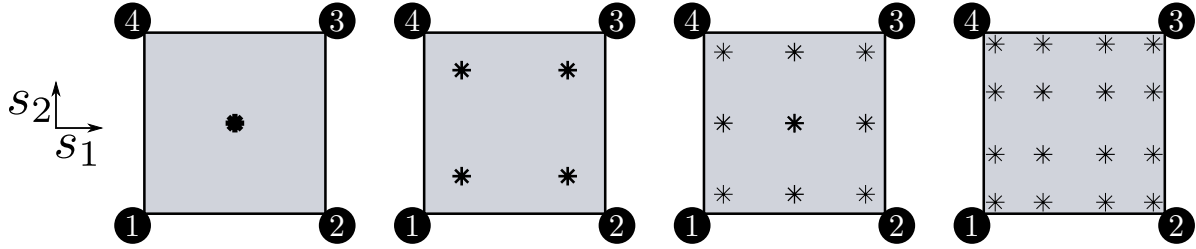
Figure 2: Examples of 2D Gauss-Legendre quadrature of the parent element with $Q = 1, 2, 3, 4$. The point size represents the weight w of the integration point. The parent elements are centred at the origin and s_1, $s_2 \in [-1, 1]$. We use an anticlockwise node numbering scheme.

is the weighting of the Gauss point, s_i, which is the ith root of the nth normalised Legendre polynomial, $P_n(1) = 1$. P_n' is the first order derivative of P_n. This method is very accurate for functions that can be accurately approximated by polynomials. In fact, for a polynomial of degree n, one needs $n - 1$ points to obtain an exact numerical solution. However, this quadrature is well-known for being unsuitable for integrating functions with poles or near-poles [15, 16].

We avoid the strict pole in $\tilde{\boldsymbol{\sigma}}$ by using the non-singular formulation described in [9], where the true singularity is avoided by adding a small cut-off radius to account for the dislocation core. However, if an integration point falls close to, or within the dislocation core, Gauss quadrature can still produce 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). Evaluating equation (10) in the parent element and mapping to the real element gives the force on node a,

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

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

Explicitly, equation (10) is actually a triple vector integral as shown in figure 3. This is because given the isotropic Burgers vector distribution proposed in [9], the dyadic form of the stress tensor produced by a straight, finite dislocation segment bounded by

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

nodes at $\mathbf{x_1}$ and $\mathbf{x_2}$ [14] is,

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

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

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

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

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 $i = j = k = 0\,.$ (20)

Using partial differentiation and integration by parts, they found a series of recurrence relations that lead to double and single integrals of similar form to equation (19). All of which are used to construct a full, exact solution. The recurrence relations stop working when $i = j = k = 0$ and $m = 1, 3$. At which point, direct integration of the remaining 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). The problem is particularly egregious when using general purpose compilers instead of high-performance or scientific computing compilers where mathematical functions are implemented more precisely. Such issues must be taken into account when using analytic tractions, which can be done by using numerical tolerances as described in section 3.

In simulations, tractions are manifested as image stresses calculated by the FE solver at FE nodes. In order to validate and compare the practical differences between analytic and numeric methods, we compare the resulting image stresses from both methods to the analytic expressions for infinite dislocations in inhomogenous media for edge dislocations [17], as well as those for screw dislocations [18, p. 59, 64]. We keep the same nomenclature and coordinate system as both infinite-domain solutions. Where the traction surface is the line $x = 0$, the dislocation line direction is the positive z-direction (pointing out of the page), the dislocation coordinates are represented by $(a,\ c)$, and points in the xy-plane described by their $(x,\ y)$ coordinates.

The original paper by Head [17] has a few typos that have been replicated in other sources. We therefore include the complete and correct expressions in equations (21) to (23). Head [17] gives two basic cases. Equation (21) corresponds to the case where $\boldsymbol{b}$ is perpendicular to the surface and positive b means it points in the positive x-direction,

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

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

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

Equation (22) corresponds to the case where the $\boldsymbol{b}$ lies parallel to the surface and positive b means it points in the positive y-direction,

$$\sigma_{xx} = D \left\{ (x - a) \frac{(x - a)^2 - (y - c)^2}{[(x - a)^2 + (y - c)^2]^2} - (x + a) \frac{(x + a)^2 - (y - c)^2}{[(x + a)^2 + (y - c)^2]^2} \right. \tag{22a}$$
$$\left. + 2a \frac{(3x + a)(x + a)^3 - 6x(x + a)(y - c)^2 - (y - c)^4}{[(x + a)^2 + (y - c)^2]^3} \right\},$$

$$\sigma_{yy} = D \left\{ (x - a) \frac{(x - a)^2 + 3(y - c)^2}{[(x - a)^2 + (y - c)^2]^2} - (x + a) \frac{(x + a)^2 + 3(y - c)^2}{[(x + a)^2 + (y - c)^2]^2} \right. \tag{22b}$$
$$\left. - 2a \frac{(x - a)(x + a)^3 - 6x(x + a)(y - c)^2 + (y - c)^4}{[(x + a)^2 + (y - c)^2]^3} \right\},$$

$$\sigma_{xy} = D(y - c) \left\{ \frac{(x - a)^2 - (y - c)^2}{[(x - a)^2 + (y - c)^2]^2} - \frac{(x + a)^2 - (y - c)^2}{[(x + a)^2 + (y - c)^2]^2} \right. \tag{22c}$$
$$\left. + 4ax \frac{3(x + a)^2 - (y - c)^2}{[(x + a)^2 + (y - c)^2]^3} \right\}.$$

Equation (23) corresponds to screw dislocations, which are markedly simpler as only the shear components are non-zero. Here $\boldsymbol{b} = \boldsymbol{l}$ so positive b means it points in the positive z-direction,

$$\sigma_{xz} = -D \left(\frac{y - c}{(x - a)^2 + (y - c)^2} - \frac{y - c}{(x + a)^2 + (y - c)^2} \right) \tag{23a}$$

$$\sigma_{yz} = D \left(\frac{x - a}{(x - a)^2 + (y - c)^2} - \frac{x + a}{(x + a)^2 + (y - c)^2} \right). \tag{23b}$$

In every case, the constant D is defined by equation (24),

$$D = \frac{\mu}{2\pi} \cdot \frac{1 + \nu}{1 - \nu^2} \cdot b. \tag{24}$$

Note the first terms of equations (21) to (23) all correspond to the stress field generated by the dislocation itself. 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 independently, which we do in order to only visualise the effect of the different traction calculations. Furthermore, equations (21) to (23) are all singular at the dislocation coordinates. Our simulation code uses the 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 9 and 10).

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, $\boldsymbol{n} = [0\,\overline{1}\,0]$. The dislocation is of pure edge character with $\boldsymbol{b} = [1\,0\,\overline{1}]$, and line direction, $\boldsymbol{l} = [\overline{1}\,2\,\overline{1}]/\sqrt{6}$ 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 these simple test cases benefits the accuracy of the numerical solutions because the stress fields exhibit symmetries about the dislocation line. If under ideal conditions for error cancellation, it can be proven that the numerical method is inferior to the analytic one, it can be more effectively argued that the analytic one should always be used instead.

Queyreau et al. [14] found the analytic solution is approximately 10 times more computationally expensive than its numerical counterpart for 1 quadrature point, our findings agree with this result but the implications for a whole simulation are favourable (see section 4).

One serious disadvantage of the analytic tractions is that the implementation of the analytic solution is also much more involved and full of snags. 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. We do this about its midpoint and around the $\boldsymbol{l} \times \boldsymbol{n}$ axis in both clockwise and anticlockwise directions. We use the mean of the values as the answer for $\theta = 0$. An example of what this rotation looks like in terms of the forces on a surface element can be seen in figure 6 (avoiding the singularity at $\theta = 0$). 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 use a total of 8 perturbations of 1 deg each (4 clockwise and 4 anti-clockwise). We found this worked well, but can be changed if desired.

However, under finite-precision arithmetic, the check for orthogonality is dependent

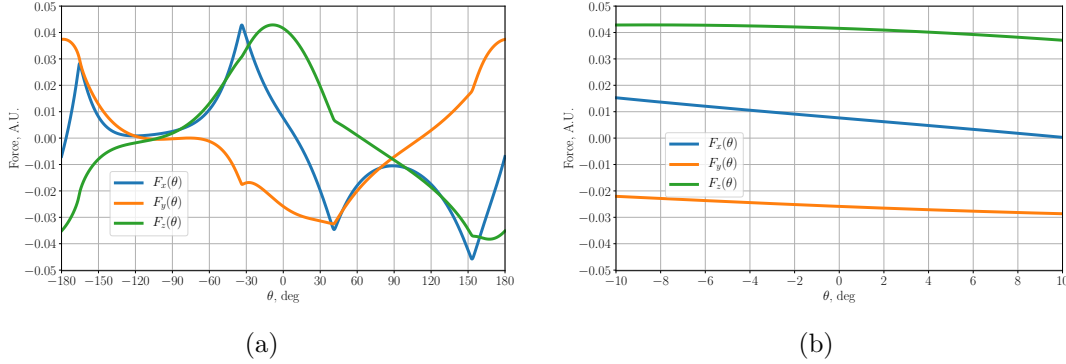(a)   (b)

Figure 6: (a) 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 $\boldsymbol{l} \times \boldsymbol{n}$. (b) is zomms into $\pm 10\,\text{deg}$, the force is smooth but not necessarily antisymmetric about the neighbourhood of $\theta = 0$.

on the length scales involved in the simulation. This is particularly important considering the aforementioned use of arctangents, hyperbolic arctangents, logarithms and the large number of recurrence relations. Causing unexpected and rampant error propagation and numerical blow up is not difficult to achieve. When it happens, finding the root cause can lead one on a wild goose chase that is best avoided and often leads to a single segment in a single step of a long simulation that is slightly too small compared to its distance to a surface element to which it is slightly too parallel. To avoid these rare but high impact scenarios, we created a heuristic that dictates how strict the tolerance should be in order for the code to consider a segment to be parallel,

$$|\boldsymbol{l} \cdot \boldsymbol{n}| \lesssim \frac{\max\left(|\boldsymbol{R} \cdot \boldsymbol{n}|\right)}{10^8}. \tag{25}$$

In our case the numerator on the RHS is simply the FEM domain'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 if the value of $\boldsymbol{l} \cdot \boldsymbol{n}$ is too close to zero. Ironically, tolerances which are too large can cause the perturbations to rotate the dislocation segment closer to the singularity, 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. This heuristic is a good general purpose rule that keeps the tolerance in a goldilocks zone.

Another issue with the rotation is that 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,

$$\theta_{\max} = \arctan\left(\frac{2d}{|\mathbf{x_2} - \mathbf{x_1}|}\right). \tag{26}$$
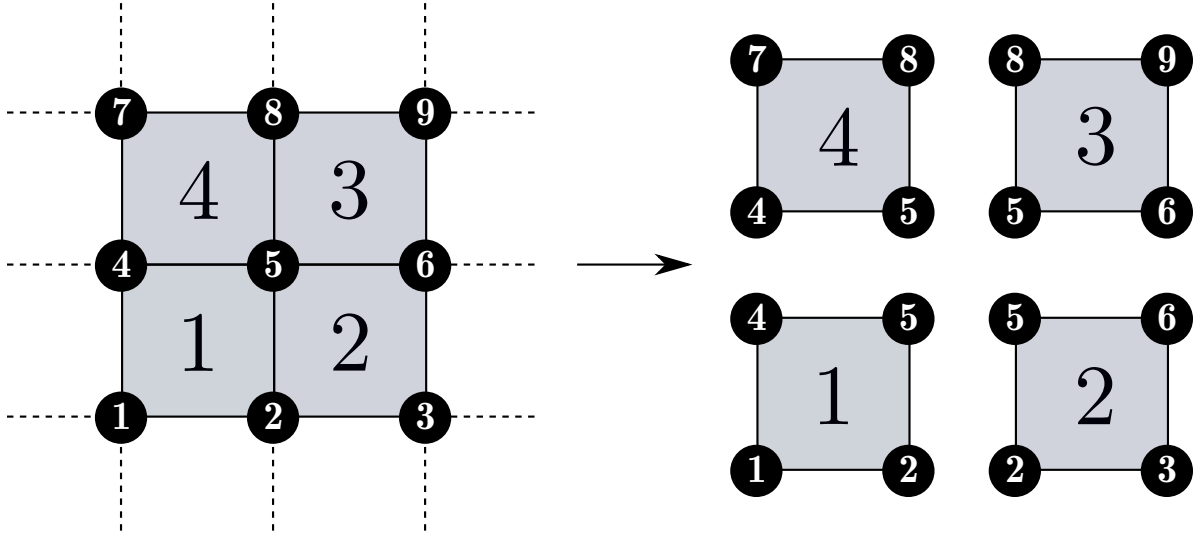
Figure 7: FE nodes are shared by either 4 element faces or 3 if it is a corner node. The total force on a given node is the summation of the force contributions from each element it belongs to.

Where d is the minimum orthogonal distance from the dislocation to the surface element i.e. the collision distance—which in our case is a function of the dislocation core radius—and $\mathbf{x_1}$, $\mathbf{x_2}$ are the dislocation segment node coordinates—whose maximum and minimum lengths are also functions of the dislocation core radius. However, θ_{max} might be too small in cases where 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 $1\,\mathrm{deg}$ perturbation such that we safely avoid this problem while keeping within the bounds of the non-singular model.

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. The node ordering for the various surfaces is that for which the calculated normals point out from the the domain, this is dependant on the specific FE mesh implementation.

The total force on a given node must include the force contributions from every element in which said node appears, see figure 7. The specifics of the mapping depend on the global FE node numbering. Using figure 7 as our reference labels for elements and nodes, e and n respectively, we can give a concrete example of how this is done by

defining,

$$\boldsymbol{x_{e,n}} \equiv \begin{bmatrix} x_{e,n} & y_{e,n} & z_{e,n} \end{bmatrix}^{\mathsf{T}}, \quad \boldsymbol{x_n} \equiv \begin{bmatrix} x_n & y_n & z_n \end{bmatrix}^{\mathsf{T}} \tag{27a}$$

$$\mathbf{N_L} = \begin{bmatrix} l_{1,1} & l_{1,2} & l_{1,4} & l_{1,5} \\ l_{2,2} & l_{2,3} & l_{2,5} & l_{2,6} \\ l_{3,5} & l_{3,6} & l_{3,8} & l_{3,9} \\ l_{4,4} & l_{4,5} & l_{4,7} & l_{4,8} \end{bmatrix}, \quad \boldsymbol{\gamma} = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_9 \end{bmatrix} \tag{27b}$$

$$\mathbf{F_e} = \begin{bmatrix} \boldsymbol{x_{1,1}} & \boldsymbol{x_{1,2}} & \boldsymbol{x_{1,4}} & \boldsymbol{x_{1,5}} \\ \boldsymbol{x_{2,2}} & \boldsymbol{x_{2,3}} & \boldsymbol{x_{2,5}} & \boldsymbol{x_{2,6}} \\ \boldsymbol{x_{3,5}} & \boldsymbol{x_{3,6}} & \boldsymbol{x_{3,8}} & \boldsymbol{x_{3,9}} \\ \boldsymbol{x_{4,4}} & \boldsymbol{x_{4,5}} & \boldsymbol{x_{4,7}} & \boldsymbol{x_{4,8}} \end{bmatrix}, \quad \tilde{\boldsymbol{F}} = \begin{bmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \vdots \\ \boldsymbol{x_9} \end{bmatrix}. \tag{27c}$$

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 7 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 (27c) 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.

Algorithm 1 illustrates how the total force on each node is obtained. However, our implementation does not strictly follow it 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.

The resulting force vector is then used in equation (5) to calculate $\hat{\boldsymbol{\sigma}}$, since $\tilde{\boldsymbol{T}} \equiv \tilde{\boldsymbol{F}}$. Figure 8 shows the simple system we used to compare the image stresses calculated by our FE solver using numeric tractions v.s. analytic tractions v.s. infinite-domain,

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**

singular image stresses in equations (21) to (23). The three test cases are: two edge dislocations and one screw, all of which have line direction $\boldsymbol{l} = [0\,0\,1]$. The Burgers vectors for the different scenarios are $\boldsymbol{b}_{\text{e1}} \equiv \boldsymbol{b} = [1\,0\,0]$, $\boldsymbol{b}_{\text{e2}} \equiv \boldsymbol{b} = [0\,1\,0]$, as defined in [17]; and $\boldsymbol{b}_{\text{s}} \equiv \boldsymbol{b} = \boldsymbol{l} = [0\,0\,1]$, as defined in [18, p. 59, 64].

The units in all our examples are normalised to lattice parameter, a is the dislocation core radius for the non-singular formulation discussed in [9], and $b \equiv \|\boldsymbol{b}\|$. The slices we took for our contour plots in section 4 are on the middle plane of the domain at $z = 1000$.

We also ran a very simple simulation comparing the results between using analytic tractions as opposed to numeric ones. Finding a simple yet clear case of tractions producing catastrophically wrong behaviours can be difficult. Large differences in image forces are relatively rare and other factors often dominate. Such factors include the core radius—which affects the dislocation line tension and therefore works against the deformation of a dislocation line—mobility law and mobility parameters used, external loading, etc.

However, we found a simple and realistic configuration that is close to the setup in the infinite domain examples. The differences between those comparisons and the simulation are described in table 1. We use a mobility law developed in-house by B. Bromage [13], that corrects common issues found in other laws. We also rotate the domain such that the $[1\,1\,1]$ and $[1\,\bar{1}\,0]$ crystallographic directions respectively correspond to the simulation's x and y-directions, for this we use the same rotation technique as [5]. The dislocation was allowed to move under no external loads or
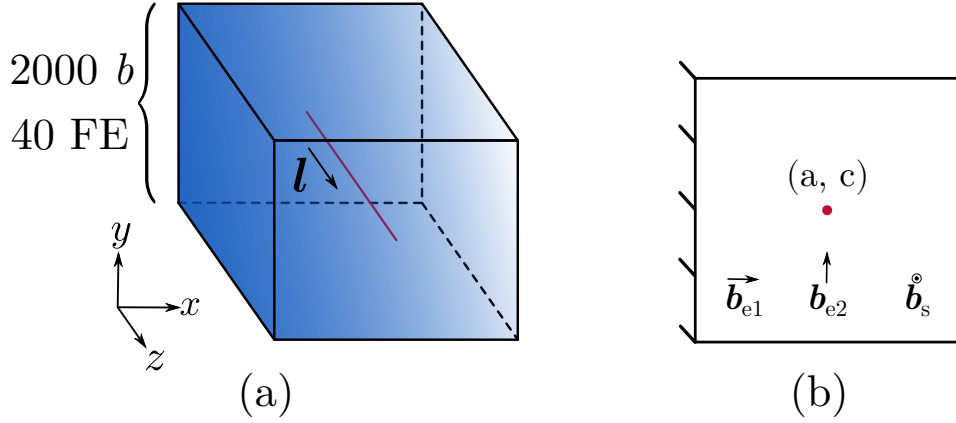
Figure 8: Dislocation parallel to a surface described by the line $x = 0$, where the (x, y) dislocation coordinates are (a, c) and the dislocation line going in the positive z-direction. (a) describes the box we used for our comparison, a $40 \times 40 \times 40$ element cubic box with side lengths equal to 2000 with $\boldsymbol{T} = \boldsymbol{0}$, traction boundary conditions only on the nodes along $x = 0$, and $\boldsymbol{U} = \boldsymbol{0}$, displacement conditions everywhere else. (b) is the 2D view with the dislocation coordinates (a, c), as well as the two edge Burgers vectors in [17], $\boldsymbol{b}_{e1} \equiv \boldsymbol{b} = [1\,0\,0]$, $\boldsymbol{b}_{e2} \equiv \boldsymbol{b} = [0\,1\,0]$ and the screw Burgers vector $\boldsymbol{b}_s \equiv \boldsymbol{b} = \boldsymbol{l} = [0\,0\,1]$ in [18, p. 59, 64].

displacements, with $\boldsymbol{T} = \boldsymbol{0}$ boundary conditions only on the yz-plane and $\boldsymbol{U} = \boldsymbol{0}$ everywhere else.

Table 1: Parameters for our simulation. Where $\mathbf{R}$ is a rotation matrix such that the $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ basis vectors of our simulation correspond to the $\boldsymbol{b}$, $\boldsymbol{n}$, $\boldsymbol{l}$ crystallographic directions, i.e. $\mathbf{R}\boldsymbol{x} = \boldsymbol{b}$. Everything else was kept just as the previous comparisons. All values are in units of lattice parameters.

Parameter	Value
Crystal Structure	BCC
$\boldsymbol{b}$	$[1\,1\,1]$
$\boldsymbol{n}$	$[1\,\bar{1}\,0]$
$\boldsymbol{l}$	$[1\,1\,\bar{2}]$
b	$\sqrt{3}/2$
a	5
Grid Size	(20, 20, 20)
Domain Size	(2000, 2000, 2000)
Lattice size	$3.18 \times 10^{-4}\ \mu\text{m}$
(x_0, y_0)	(62.5, 1000)
Min segment length	50
Max segment length	125
$\mathbf{R}$	$[\hat{\boldsymbol{b}}\ \hat{\boldsymbol{n}}\ \hat{\boldsymbol{l}}]$

4. Results and Discussion

Figure 9 shows that even for dislocations only one dislocation core radius (5) away from the surface element, the force can be obtained, up to numerical precision, with 1000 Gauss quadrature points Q for all segment lengths tested. It also shows a very peculiar issue Gauss quadrature has when computing integrals of rational functions when the Gauss nodes are close poles/maximal values. This undesirable behaviour is observed in the case where $Q = 11$. Where the highest weighted Gauss node is closest to the point where $1/R_a$ is maximal, resulting in lower accuracy when compared to $Q = 2, 10$ in figure 9 (a) and (b). It is worth noting however that the relative errors for small numbers of quadrature points don't really start decreasing until relatively large distances. And when close to a surface, these can be quite large even under hihgly symmetric circumstances.

   The limitations become even more evident when the dislocation line segment is parallel to a surface element. In figure 10, we observe the relative errors are quite large when close to the surface. At distances larger than 10^4, loss of significance causes the relative errors to converge at ~ 1 which is expected as two finite precision floating point numbers get closer to zero. Of particular note is how large the relative errors are even at 100 lattice units away from the surface, even for large numbers of quadrature points. This figure backs up the earlier point regarding Gauss nodes close to maximal values of rational functions. It can be shocking to see that as $Q = 2$ performs significantly

(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.

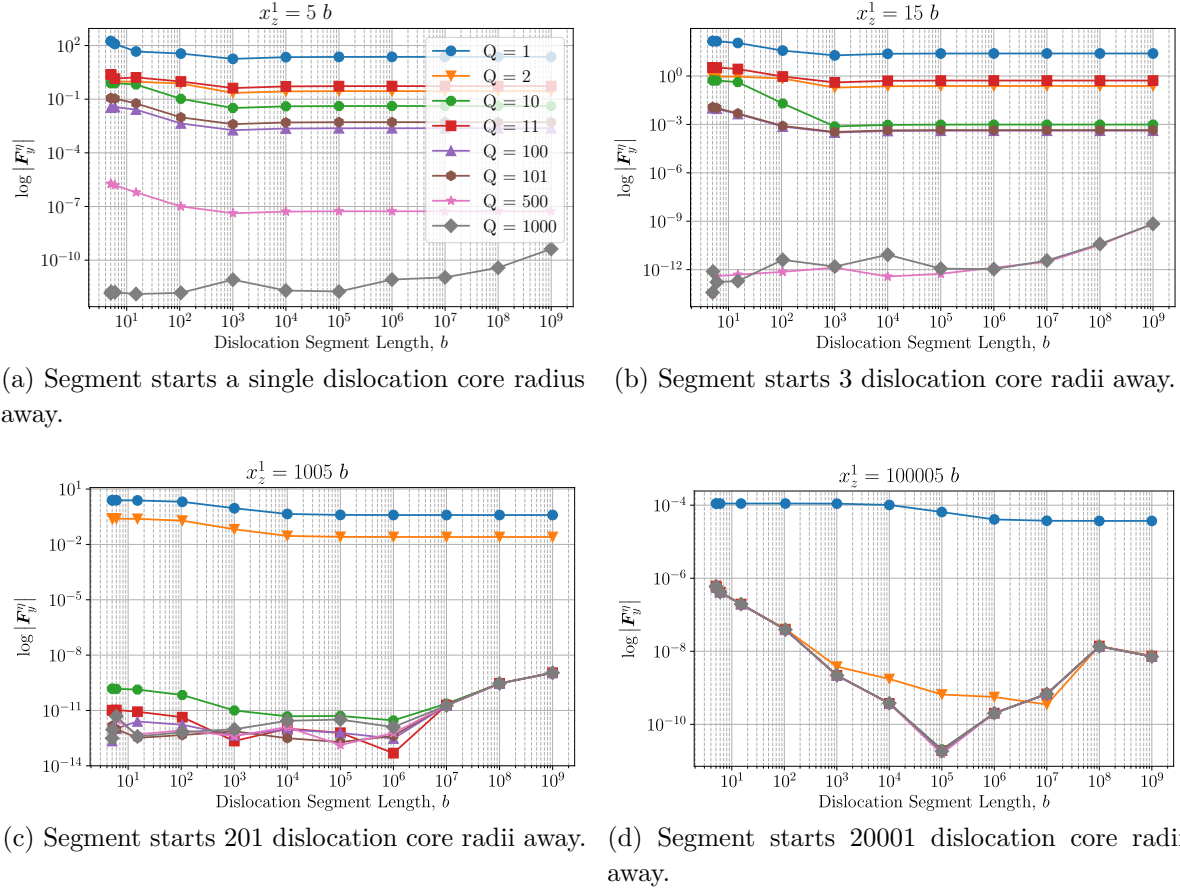(d) Segment starts 20001 dislocation core radii away.

Figure 9: 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 $\mathbf{x_1}$. $\boldsymbol{b} = [1\,0\,0]$, line direction, $\boldsymbol{l} = [0\,0\,1]$, and dislocation core radius, $a = 5$, the surface element's normal and size are, $\boldsymbol{n} = [0\,0\,1]$, $L = 1000$, respectively. Q is the number of quadrature points per dimension.

better than $Q = 10, 11, 100, 100$ at distances from the surface as having nodes near the maximal values is a large source of error. Consequently, different configurations have different optimal numbers of points. The numerical instability of this method, which when coupled to the chaotic nature of dislocation dynamics and stiffness of the equations of motion, can lead to large deviations between simulations.

From figures 9 and 10 one might be tempted to say that for a segment parallel to a surface, Gauss quadrature performs far worse than for a perpendicular one. However there is a further wrinkle in this problem: symmetry. To exemplify this we plot the relevant components of the stress tensor for the arrangement found in figure 5(a) in figures 11a to 11c. The σ_{xz} and σ_{zz} components are antisymmetric about the centre of the element. If we use Gauss quadrature on them, we sample equivalent but oppositely valued points that are equally weighted, thus the sum vanishes and therefore do not contribute to figure 9. However, σ_{yz} does not vanish, but can be accurately integrated
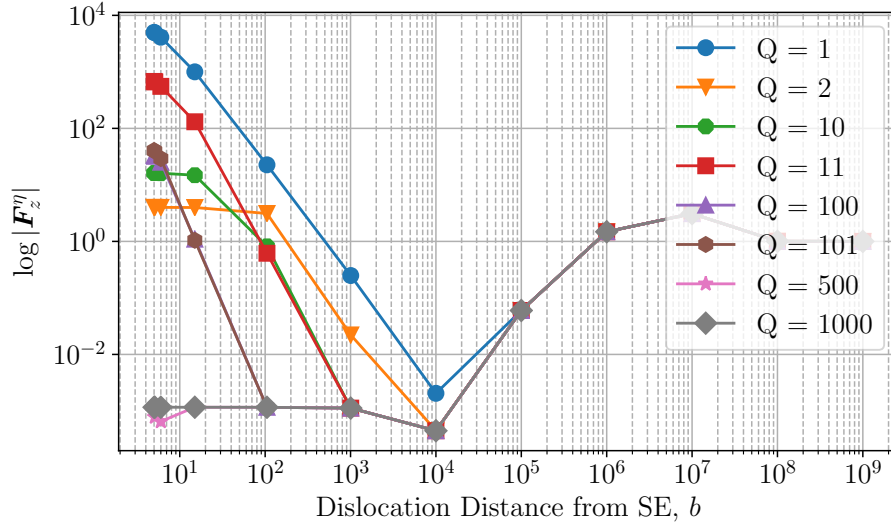
Figure 10: Log-log plot of the relative error as a function of distance from the surface element for a parallel edge dislocation (figure 5b). Burgers vector, $\boldsymbol{b} = [0\,1\,0]$, line direction, $\boldsymbol{l} = [1\,0\,0]$, dislocation core and surface element parameters are the same as figure 9. The dislocation length is fixed to 10^6, $x \in [-0.5 \times 10^6, 0.5 \times 10^6]$ and bisects the surface element along the $[1\,0\,0]$ direction. The whole dislocation was segmented into 10^4 pieces of length 100 to prevent the dislocation from intersecting the surface element when they were rotated to avoid the singularity. The relative error at large distances ($> 10^6$) converges to one due to loss of significance.

with sufficiently large Q. If we were to move the dislocation off-centre such that these symmetries are broken, the errors would increase.

We also plot the relevant stresses for the arrangement described by figure 5(b) in figures 11d to 11f. From figure 11d and figure 11e, it is immediately apparent why gauss quadrature fails so spectacularly in figure 10. At low numbers of quadrature points, it fails to appropriately sample the rapidly changing value of $\tilde{\sigma}_{xz}$ at both ends of the dislocation, as well those in the neighbourhood of the dislocation in $\tilde{\sigma}_{yz}$. Moreover, the further away the quadrature points are from the midpoint of the domain, the lower their relative weighting. So even with a relatively large number of them, there can still be large errors. Which is a particularly egregious problem in figure 11e, and explains why such a large number of quadrature points is required to accurately compute the integral.

Despite these being artificially idealised examples that illustrate the failings of Gauss quadrature, other problematic scenarios commonly show up in simulations. These tend to worsen with smaller core radii a, fewer Gauss nodes, higher dislocation densities near surfaces, more permissive mobility functions, and coarser FE meshes. The $\mathcal{O}(1/R)$ decay rate of $\tilde{\boldsymbol{\sigma}}$ and chaotic nature of dislocation dynamics, means these errors may result in unwarranted topological changes that cascade as the simulation advances. This is
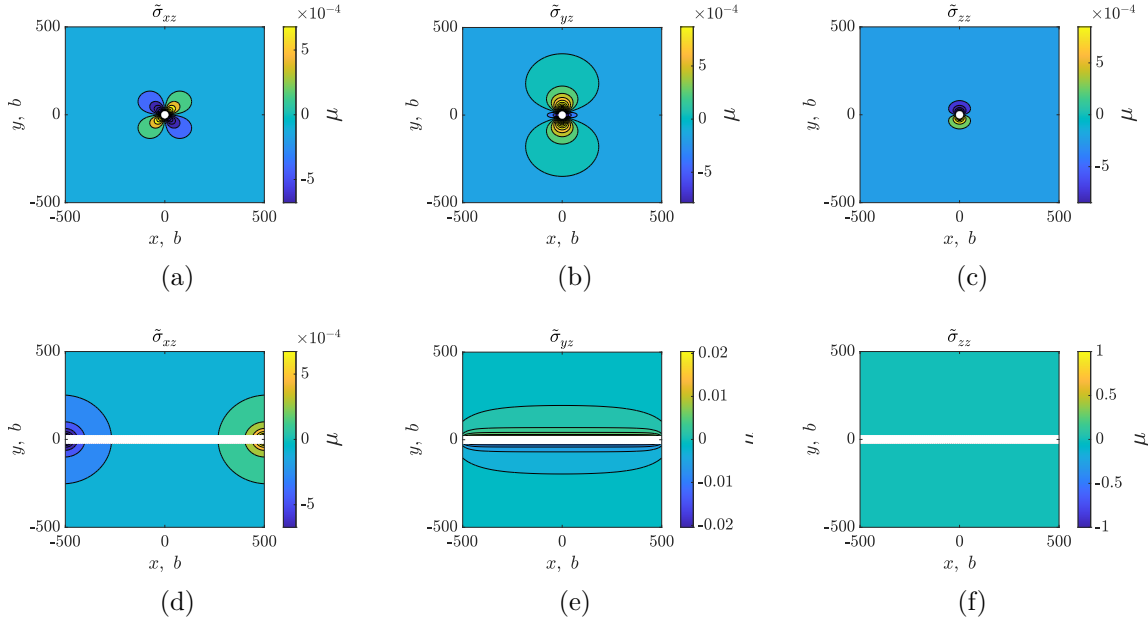
Figure 11: (a) to (c) show the real stress fields from a dislocation in the same configuration that yields the plots found in figure 9 as shown in figure 5(a), where the closest node is a dislocation core radius away from the surface, $a = 5$. (e) to (f) does the same for the configuration that yields figure 10 as shown in figure 5(b), where the whole dislocation is a core radius away from the surface, $a = 5$. The white line or dot is the dislocation. Units are in terms of lattice parameters.

particularly deleterious when doing simulations with higher dislocation densities and/or where a large number of dislocations are close to the surface, such as nanoindentation simulations.

As stated in section 3, tractions are used to calculate the image stresses resulting from the boundary conditions. We therefore compare the differences in image stresses resulting from numeric ($Q = 1$) and analytic traction calculations of both our implementations and how they compare to the infinite-domain, singular expressions in equations (21) to (23)‡. The stress field comparisons for all three cases are found in figures 12 to 14, where the dislocation is denoted by a white dot.

Figure 12 shows the stress fields corresponding to analytic expressions for image stress components, $\hat{\sigma}_{ij}$ in equation (21) where no superscript denotes the infinite-domain singular expressions, the A superscript are the stresses calculated from analytic tractions and the N superscript are those coming from numeric tractions where $Q = 1$ (same nomenclature in figures 13 and 14). The setup corresponds to the one described in figure 8 where $\boldsymbol{b} = \boldsymbol{b}_{\mathrm{e1}} = [1\,0\,0]$ and the dislocation is found at $(26, 1000)$.

It is clear edge effects play a role in the generated stresses. All three components

‡ Since the first term in each equation corresponds to the real stresses, we omit them to view the image stresses.
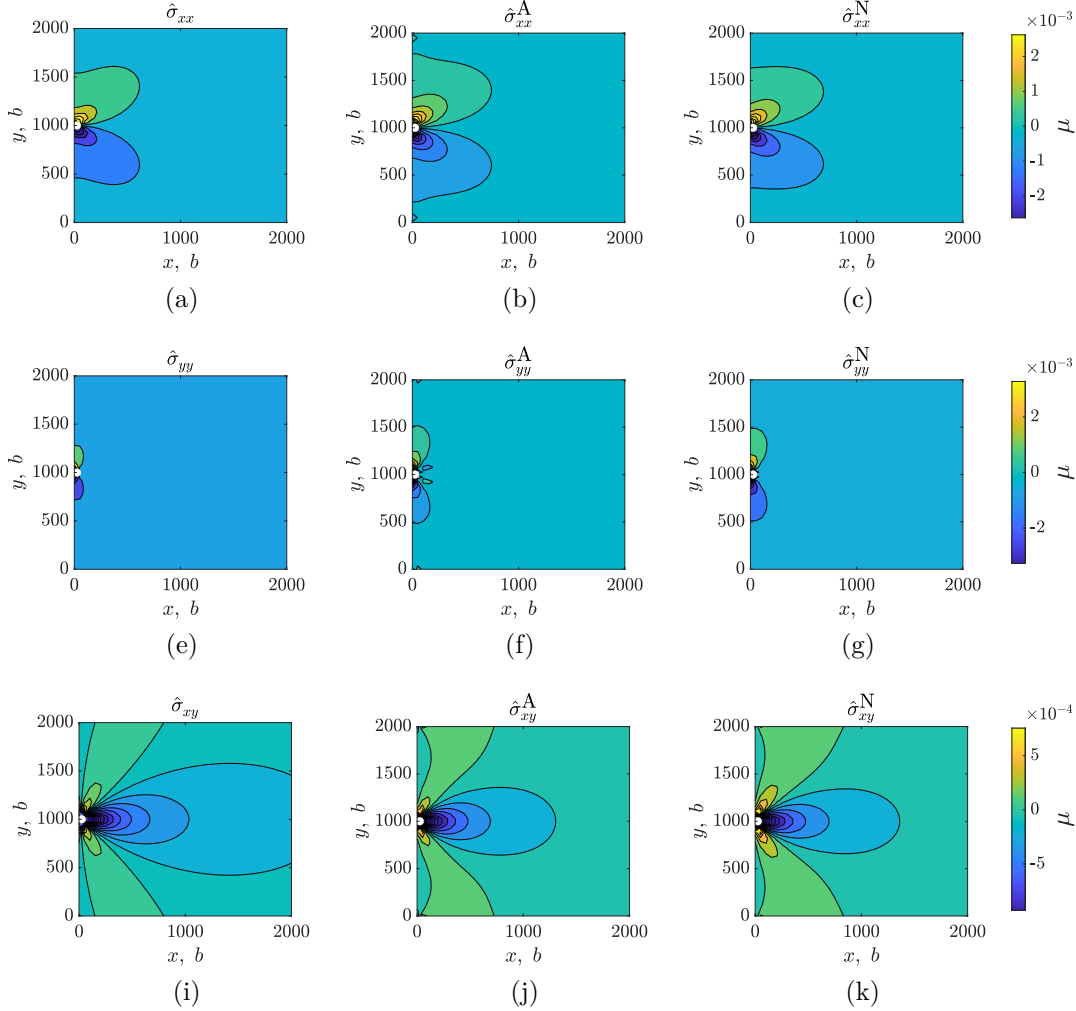
Figure 12: Image stresses for an edge dislocation with $\boldsymbol{l} = [001]$, $\boldsymbol{b} = [100]$, where $a = 5$, with coordinates $(26, 1000)$, i.e. the centre of the first FE from the surface at $x = 0$, and in the centre of the simulation box along the y-direction. (a), (e) and (i) are the stress fields for the infinite-domain solution, $\hat{\boldsymbol{\sigma}}$; (b), (f) and (j) are those obtained from analytic tractions + FEM, $\hat{\boldsymbol{\sigma}}^{\mathrm{A}}$; (c), (g) and (k) are those obtained using numeric tractions $(Q = 1)$ + FEM, $\hat{\boldsymbol{\sigma}}^{\mathrm{N}}$. (a) to (c) represent the xx; (e) to (g) the yy; and (i) to (k) the xy components of the stress tensor.

have notable differences from the infinite-domain solutions resulting from the finite constraints, but the overall agreement between them all is quite good.

Things get markedly more interesting when looking at $\boldsymbol{b} = \boldsymbol{b}_{\mathrm{e2}} = [0\,1\,0]$ in figure 13 for a dislocation in the same place, $(26, 1000)$. Of particular note is $\hat{\sigma}_{xx}$, where a comparison between figures 13a and 13b and figure 13c reveals one of the major issues with numeric tractions. If we look at the neighbourhood of the dislocation (just to the right), we will find a sign inversion i.e. yellow and green contours as opposed to purple and blue. As well as a drastically different isosurface shape. Image stresses like
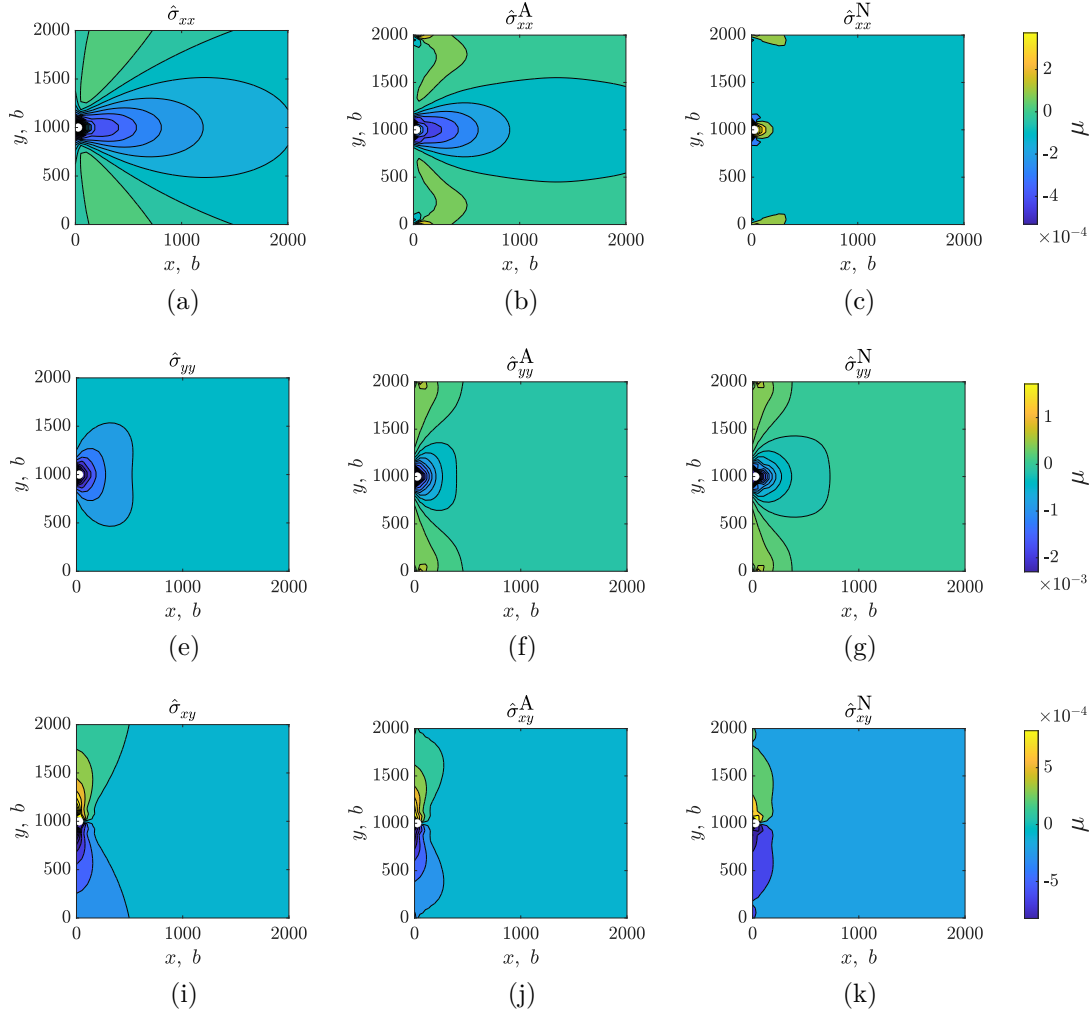
Figure 13: Image stresses for an edge dislocation with $\boldsymbol{l} = [0\,0\,1]$, $\boldsymbol{b} = [0\,1\,0]$, where $a = 5$, with coordinates $(26, 1000)$, i.e. the centre of the first FE from the surface at $x = 0$, and in the centre of the simulation box from top to bottom. (a), (e) and (i) are the stress fields for the infinite-domain solution, $\hat{\boldsymbol{\sigma}}$; (b), (f) and (j) are those obtaine from analytic tractions + FEM, $\hat{\boldsymbol{\sigma}}^{\mathrm{A}}$; (c), (g) and (k) are those obtained using numeric tractions $(Q = 1)$ + FEM, $\hat{\boldsymbol{\sigma}}^{\mathrm{N}}$. (a) to (c) represent the xx; (e) to (g) the yy; and (i) to (k) the xy components of the stress tensor.

those can lead dislocations to behave quite differently than they should, particularly if the sign inversion also has a significantly different magnitude than the correct solution. Specifically, there is a region in the positive x-direction away from the dislocation where $\hat{\sigma}_{xx}$ is tensile rather than compressive. It is as absurd as a ship that floats by lowering the water level.

Perhaps the most evident deformation resulting from the displacement boundaries can be seen when $\boldsymbol{b} = \boldsymbol{b}_{\mathrm{s}}$ in figure 14, where the lobes of the isolines are highly deformed when compared to the infinite-domain solutions. Though deformed, their familiar shape
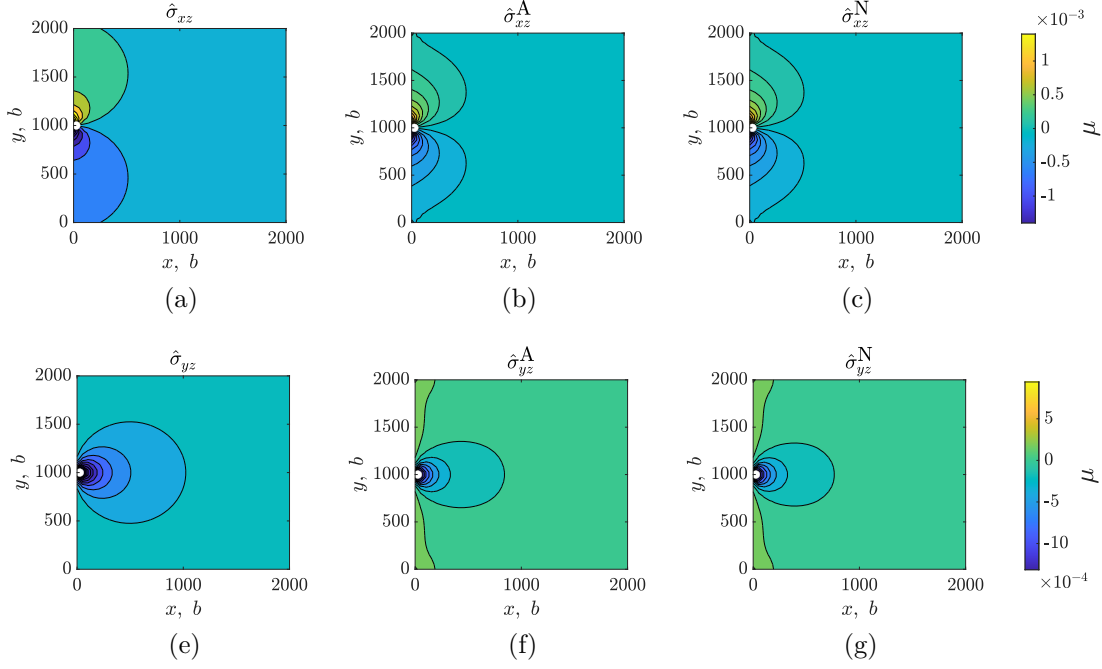
Figure 14: Image stresses for a screw dislocation with $\boldsymbol{l} = [0\,0\,1]$, $\boldsymbol{b} = [0\,0\,1]$, where $a = 5$, with coordinates $(26, 1000)$, i.e. the centre of the first FE from the surface at $x = 0$, and in the centre of the simulation box from top to bottom. (a) and (e) are the stress fields for the infinite-domain solution, $\hat{\boldsymbol{\sigma}}$; (b) and (f) are those obtaine from analytic tractions + FEM, $\hat{\boldsymbol{\sigma}}^{\mathrm{A}}$; (c) and (g) are those obtained using numeric tractions $(Q = 1)$ + FEM, $\hat{\boldsymbol{\sigma}}^{\mathrm{N}}$. (a) to (c) represent the xz; (e) to (g) the yz.

is still recognisable and both analytic and numeric tractions yield fairly similar fields.

From figures 12 to 14 it seems like both analytic and numeric tractions are appropriate in most cases. At least at these scales, there is only one instance where numeric tractions yield very incorrect results. That said, these can have a significant impact on simulations, particularly those where multiple dislocations interact with surfaces in close proximity with one another.

We also produced line plots to better show how the stress fields deviate from one another. Figure 15 shows line plots through the line $x = 103$ for a dislocation at $(26, 1000)$. Figures 15a to 15c correspond to $\boldsymbol{b} = \boldsymbol{b}_{\mathrm{e1}}$, figures 15d to 15f to $\boldsymbol{b} = \boldsymbol{b}_{\mathrm{e2}}$ and figures 15g and 15h to $\boldsymbol{b} = \boldsymbol{b}_{\mathrm{s}}$. Here, the singular nature of the infinite-domain solutions is evidenced by the sharp spikes in its stresses. In general, the non-singular formulation smoothes out the stress line plots. However, there are a few instances where the numeric tractions lead to larger spikes than even the infinite-domain solutions such as in figure 15d. Again, the general shape of the line plots is the same but the tendancy for numerical tractions to spike under specific circumstances is evident in almost every case.

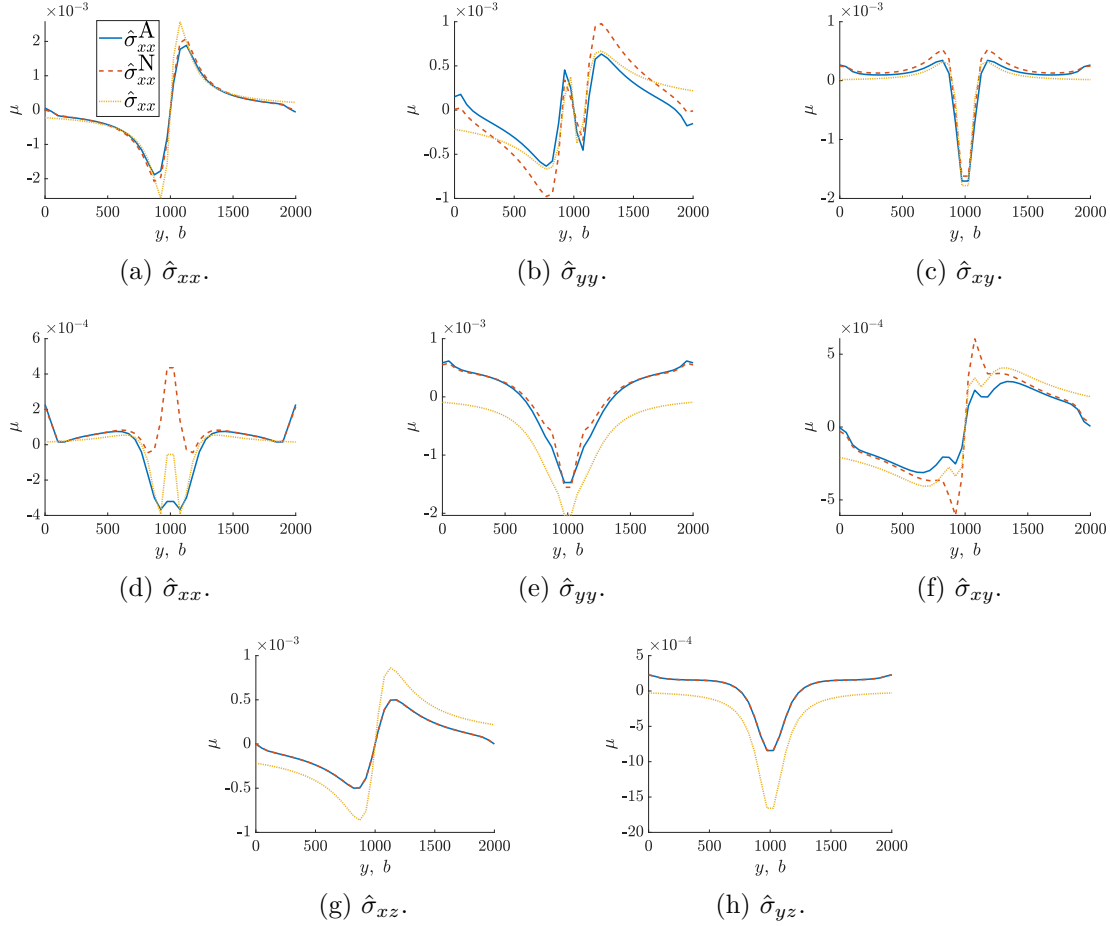To show the convergence in methods we also show stress fields in figure 17 and line

Figure 15: Line plots of the image stresses of a dislocation at $(26, 1000)$ for a line going through $x = 103$ (start of the third element) for the analytic image stresses, as well as those calculated with numeric and analytic tractions. (a) to (c) correspond to $\boldsymbol{b} = \boldsymbol{b}_{\mathrm{e1}}$, (d) to (f) $\boldsymbol{b} = \boldsymbol{b}_{\mathrm{e2}}$ and (g) to (h) to $\boldsymbol{b} = \boldsymbol{b}_{\mathrm{s}}$.

plots figure 17 of the image and total stress fields as an edge dislocation with $\boldsymbol{b} = \boldsymbol{b}_{\mathrm{e2}}$ moves from $(77, 1000)$ to $(487, 1000)$. The line plots are taken from the second closest set of nodes in the positive x-direction i.e. at $x = 103, 513$.

Notice that the image stresses in figures 16a to 16c are all very close to figure 13c, which is the stress field calculated via numerical tractions for a dislocation that is slightly closer to the surface. Essentially, the numerical tractions over or underestimated a set of forces something that become the dominant contributors as the dislocation moves away from the surface.

From figures 16 and 17 it can be observed that both analytic and numeric tractions converge to similar shapes to each other as expected. However it also becomes clear that the infinite-domain solution is not totally accurate for image stresses in finite domains due to edge effects.

Lastly, to show how much numeric tractions can affect a result, figure 18 shows a
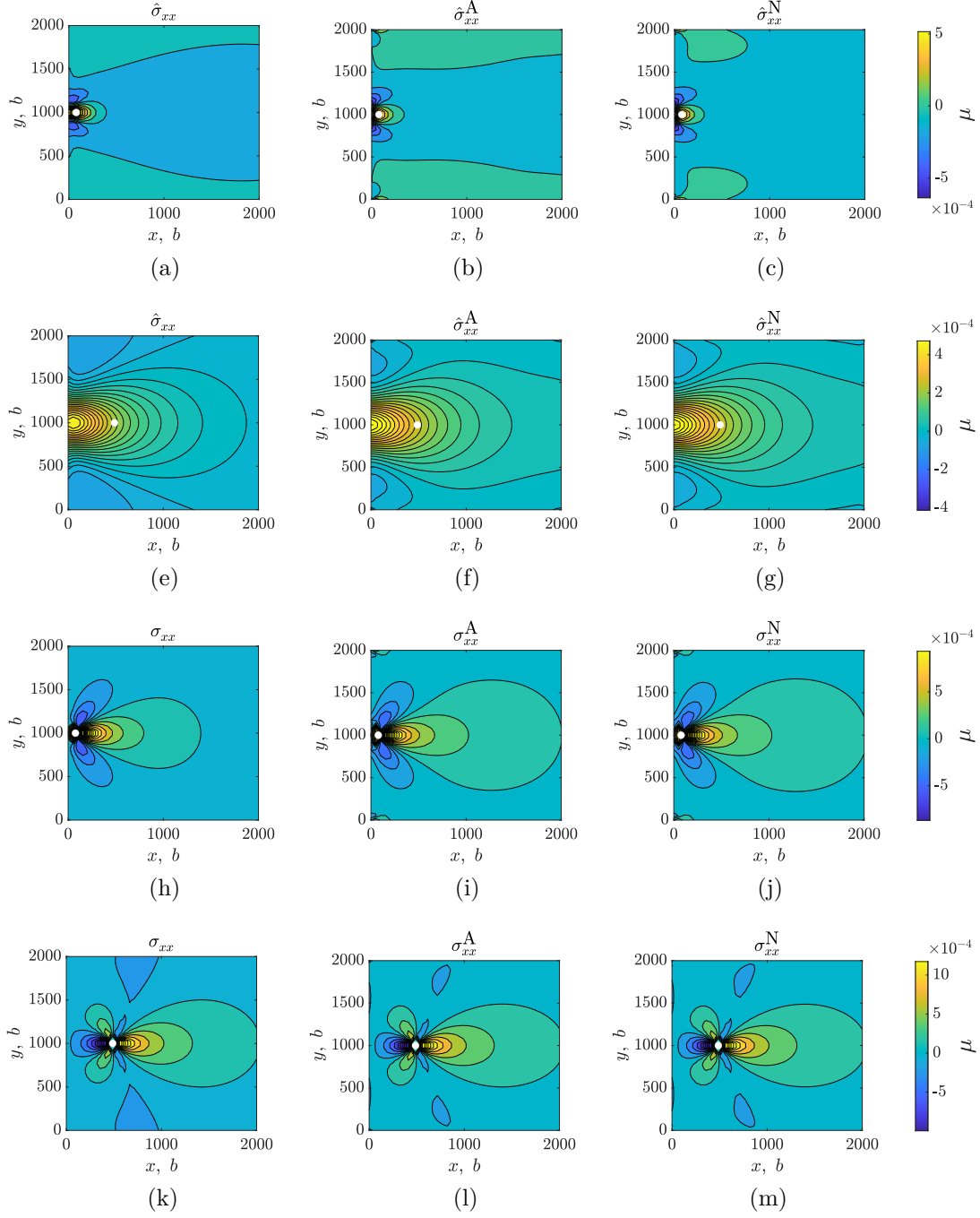
Figure 16: Stresses for an edge dislocation with $\boldsymbol{l} = [0\,0\,1]$, $\boldsymbol{b} = [0\,1\,0]$. Subfigures (a) to (g) show image stresses; (h) to (m) show total stresses. In subfigures (a) to (c) and (h) to (j) the dislocation is found at (77, 1000); in (e) to (g) and (k) to (m) the dislocation is at (487, 1000).

(a) $\hat{\sigma}_{xx}$.

(b) $\hat{\sigma}_{xx}$.

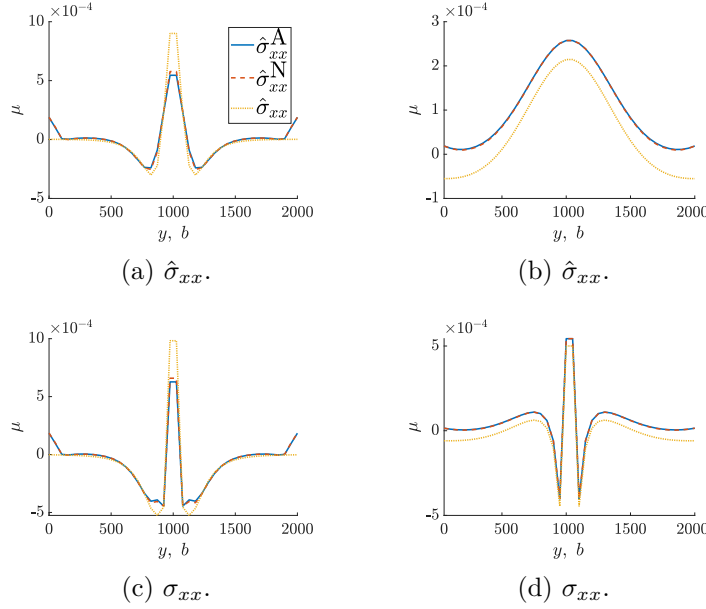(c) σ_{xx}.

(d) σ_{xx}.

Figure 17: Line plots corresponding to figure 16. (a) and (c) are of the image stresses; (b) and (d) are of total stresses. (a) and (b) are of a dislocation at (77, 1000), taken along the line $x = 103$. (b) and (d) are of a dislocation at (487, 1000), taken along the line $x = 513$.

few snapshots of the simulation described in section 3. The figures shown are not at equivalent times because the simulations using numeric tractions ran indefenitely. The line tension equilibrated with the image forces on the traction-free surface, preventing the dislocation from exiting and keeping it oscillating in a local energy minimum. The simulation using analytic tractions ended when the dislocation completely exited the surface as expected. It is feasable that at some point, the simulation using numeric tractions could jump out of the local minimum due to some spike in the tractions, but the last image was taken for a simulation time approximately 20 times greater than it took the one using analytic tractions to end when the dislocation fully exited the box.

(a) Analytic tractions $t = 0$.

(b) Analytic tractions mid simulation.

(c) Analytic tractions end.

(d) Numeric tractions $t = 0$.

(e) Numeric tractions mid simulation.
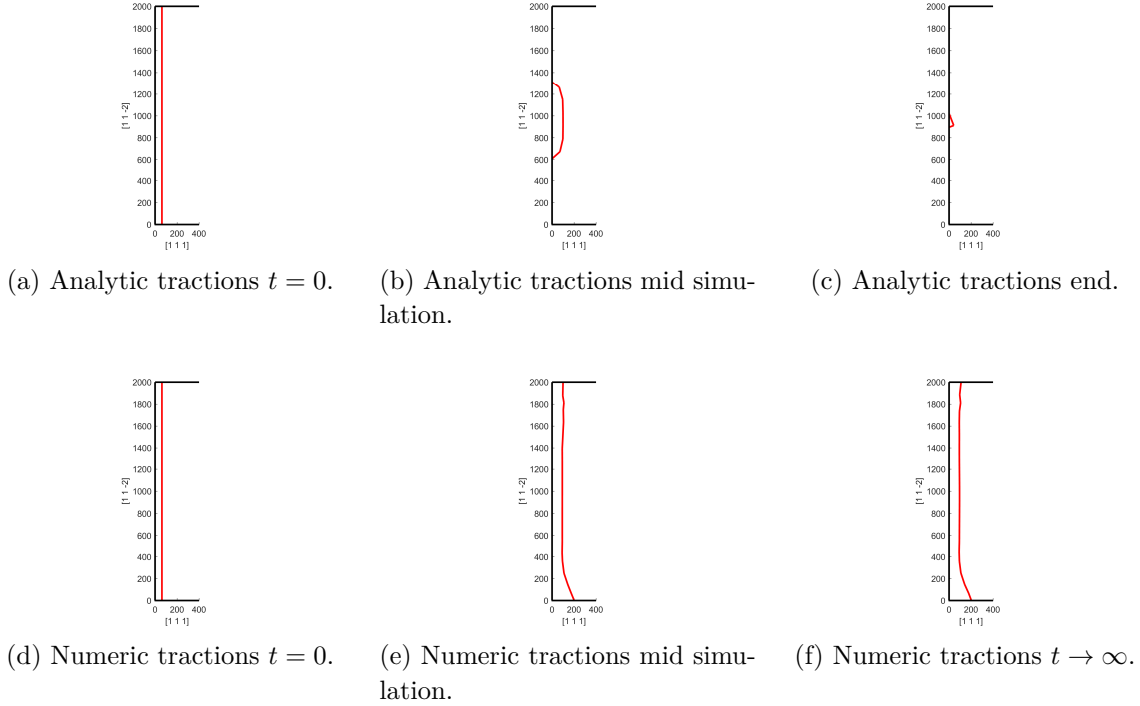
(f) Numeric tractions $t \to \infty$.

Figure 18: Progression of simulations using both traction calculation methods.

5. Conclusion

Often the effects of tractions on a simulation can be quite subtle if things do not go catastrophically wrong, but a quick and easy way of seeing how the numeric tractions have a tendancy to spike and invert sign (as in figure 15d) is to pause a simulation while its running and scatter plot the numeric tractions vs analytic ones. The plot will show a positive correlation—as the sign inversions in numeric tractions are relatively rare events—but whichever axis corresponds to the numeric tractions will have the largest range. Using proportional axes makes the differences quite evident even at a quick glance as the aspect ratio will be far from 1:1.

   In other parts of our work, we have noted that despite the analytic tractions taking approximately 10 times longer to compute than numerically calculated tractions with $Q = 1$ (which agrees with the findings in [14]), yield more stable and ultimately faster simulations. Even simple simulations are typically faster when using analytic tractions over numeric ones. This was not the case with the simulation we show here, but the numeric tractions led to a hung simulation, which we have found is quite common. The margin by which using analytic tractions leads to faster overall simulations grows as simulation complexity increases. Only the simplest simulations initially benefit from numeric tractions but this often decreases and reverses as the simulation advances. Another fortunate side effect of more accurate tractions is the fact that fewer dislocation segments are generated during simulations, which Bromage and Tarleton [13] found

when correctly accounting for the displacements generated by dislocations.

Given our findings, we cannot recommend using numeric tractions when analytic ones are available. The losses in computational speed that result from moving from one to the other are more than made up for by the fewer topological operations, fewer generated segments, and more accurate velocities. All of which result in fewer calculations of segment-segment interactions, fewer collisions, larger timesteps and ultimately cleaner simulations that run into fewer snags along the way.

There is a case however, for combining both approaches in larger scale simulations. Since both numeric and analytic tractions converge to the same value as the dislocation-surface distance increases, a hybrid approach may possibly be undertaken without many negatives. Furthermore, [14] also derived a Taylor series expansion of the solutions which can be used in cases where the dislocation-surface distance is large. These may be worth exploring in larger scale simulatios. However, in the types of systems we model, other parts of our model tend to be much more rate-limiting than tractions. As such, we have all but ceased to use numeric tractions in our work.

6. 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.