

# Accurate evaluation of dislocation tractions

Daniel Celis-Garza, Edmund Tarleton

E-mail: [edmund.tarleton@materials.ox.ac.uk](mailto: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,  $\mathbf{T}$ , on  $S_T$  and displacement boundary conditions,  $\mathbf{U}$ , on  $S_U$ . The  $(\tilde{\cdot})$  fields are those generated by the dislocations in an infinite solid and are obtained by evaluating analytic solutions in a DDD simulation [9]. Formally the dislocation field satisfies

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

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

$$\left. \begin{aligned} \tilde{\mathbf{u}} &= \tilde{\mathbf{U}} & t > 0 \\ \tilde{\mathbf{u}} &= \mathbf{0} & t = 0 \end{aligned} \right\} \text{on } S_U. \quad (3)$$

As the dislocation fields do not vanish on  $S$  the dislocations load the volume by generating tractions,  $\tilde{\mathbf{T}}$ , on  $S_T$  and displacements,  $\tilde{\mathbf{U}}$ , 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  $\mathbf{T}$  on  $S_T$  and  $\mathbf{u}$  on  $S_u$ . First, the traction,  $\tilde{\mathbf{T}}$ , and displacement,  $\tilde{\mathbf{U}}$ , 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  $\hat{\mathbf{T}} = \mathbf{T} - \tilde{\mathbf{T}}$  and  $\hat{\mathbf{u}} = \mathbf{u} - \tilde{\mathbf{u}}$ .

deforms  $V$  generating an additional “image” stress which the dislocations then feel. Therefore corrective ( $\hat{\cdot}$ ) 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

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

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

$$\hat{\mathbf{u}} = \mathbf{u} - \tilde{\mathbf{u}} \text{ on } S_U \quad (6)$$

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

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

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \mathbf{T} \text{ on } S_T \quad (8)$$

$$\left. \begin{aligned} \mathbf{u} &= \mathbf{U} \quad t > 0 \\ \mathbf{u} &= \mathbf{0} \quad t = 0 \end{aligned} \right\} \text{on } S_U \quad (9)$$

The method is not without problems however. As a dislocation segment moves closer to the surface, its ( $\sim$ ) 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,

$$\mathbf{F}_a = \int_{S_e} [\tilde{\boldsymbol{\sigma}}(\mathbf{x}) \cdot \mathbf{n}] N_a(\mathbf{x}) \, dS_e, \quad (10)$$

where  $dS_e$  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,

$$\begin{aligned} N_1 &= \frac{1}{4}(1 - s_1)(1 - s_2) \\ N_2 &= \frac{1}{4}(1 + s_1)(1 - s_2) \\ N_3 &= \frac{1}{4}(1 + s_1)(1 + s_2) \\ N_4 &= \frac{1}{4}(1 - s_1)(1 + s_2). \end{aligned} \quad (11)$$

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

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

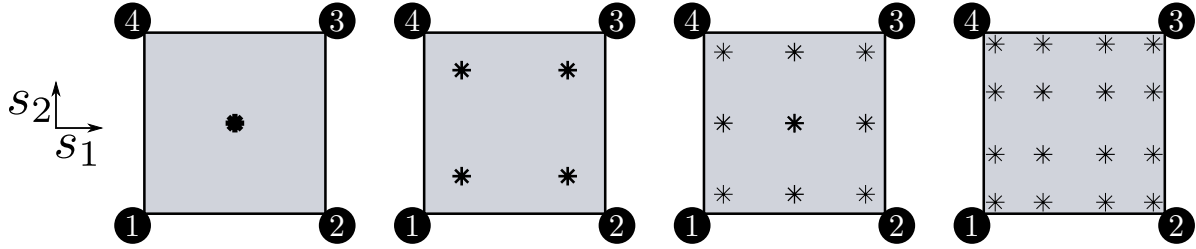


Figure 2: Examples of 2D Gauss-Legendre quadrature of the parent element with  $Q = 1, 2, 3, 4$ . The point size represents the weight  $w$  of the integration point. The parent elements are centred at the origin and  $s_1, s_2 \in [-1, 1]$ . We use an anticlockwise node numbering scheme.

is the weighting of the Gauss point  $s_i$  which is the  $i^{\text{th}}$  root of  $P_n$ ; the  $n^{\text{th}}$  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  $a$ ,

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

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. 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 produced by a straight finite dislocation segment



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

bounded by nodes at  $\mathbf{x}_1$  and  $\mathbf{x}_2$  is [14],

$$\begin{aligned}
 \boldsymbol{\sigma}(\mathbf{x}) = & -\frac{\mu}{8\pi} \int_{\mathbf{x}_1}^{\mathbf{x}_2} \left( \frac{2}{R_a^3} + \frac{3a^2}{R_a^5} \right) [(\mathbf{R} \times \mathbf{b}) \otimes d\mathbf{x}' + d\mathbf{x}' \otimes (\mathbf{R} \times \mathbf{b})] \\
 & + \frac{\mu}{4\pi(1-\nu)} \int_{\mathbf{x}_1}^{\mathbf{x}_2} \left( \frac{1}{R_a^3} + \frac{3a^2}{R_a^5} \right) [(\mathbf{R} \times \mathbf{b}) \cdot d\mathbf{x}'] \mathbf{I}_2 \\
 & - \frac{\mu}{4\pi(1-\nu)} \int_{\mathbf{x}_1}^{\mathbf{x}_2} \frac{1}{R_a^3} [(\mathbf{b} \times d\mathbf{x}') \otimes \mathbf{R} + \mathbf{R} \otimes (\mathbf{b} \times d\mathbf{x}')] \\
 & + \frac{\mu}{4\pi(1-\nu)} \int_{\mathbf{x}_1}^{\mathbf{x}_2} \frac{3}{R_a^5} [(\mathbf{R} \times \mathbf{b}) \cdot d\mathbf{x}'] \mathbf{R} \otimes \mathbf{R},
 \end{aligned} \tag{14}$$

where,

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

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

$$d\mathbf{x}' = -dy\mathbf{l} \tag{17}$$

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

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

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,

$$H_{ijkl} = \int_{r_1}^{r_2} \int_{s_1}^{s_2} \int_{y_1}^{y_2} \frac{r^i s^j y^k}{R_a^m} \quad (19)$$

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  $\mathbf{b}$  is perpendicular

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

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

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

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

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

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

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

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

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

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

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

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

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

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 produce 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 dissimilar 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



Figure 4: Relative error ( $\mathbf{F}^\eta$ ) 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,  $\mathbf{n} = [0 \ 1 \ 0]$ . The dislocation is of pure edge character with  $\mathbf{b} = [1 \ 0 \ 1]$ , and line direction,  $\mathbf{l} = [1 \ 2 \ 1]/\sqrt{6}$  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  $\mathbf{l} \perp \mathbf{n}$ , we get a singularity. As mentioned in [14], an easy fix is to rotate the line segment about its midpoint around the  $\mathbf{l} \times \mathbf{n}$  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

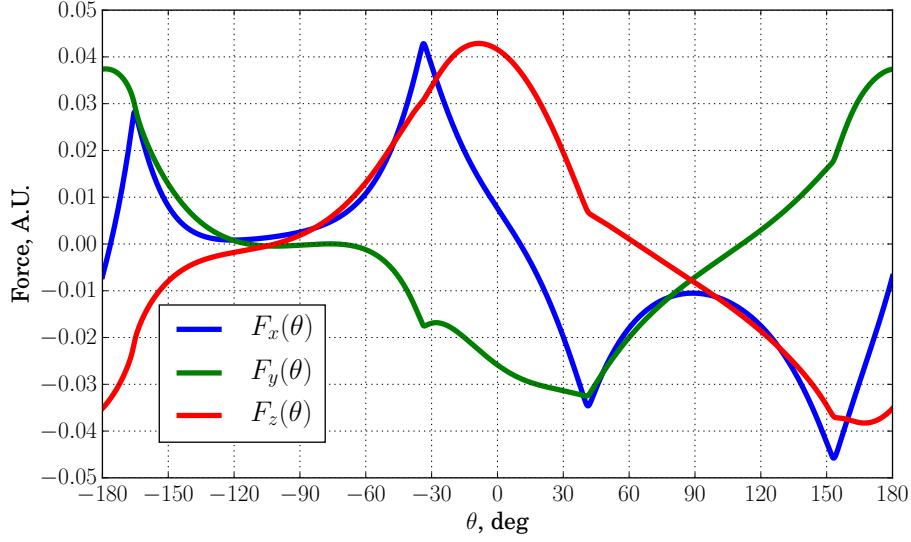


Figure 6: Example of the components of the total force on a surface element (the force summed over all four FE nodes) as a dislocation segment parallel to the surface element ( $\theta = 0$ ) is rotated about its midpoint around the axis defined by  $\mathbf{l} \times \mathbf{n}$ . The force is smooth 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,

$$|\mathbf{l} \cdot \mathbf{n}| \lesssim \frac{\max(|\mathbf{R} \cdot \mathbf{n}|)}{10^8}, \quad (24)$$

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,

$$\theta_{\max} = \arctan \left( \frac{2d}{|\mathbf{x}_2 - \mathbf{x}_1|} \right), \quad (25)$$

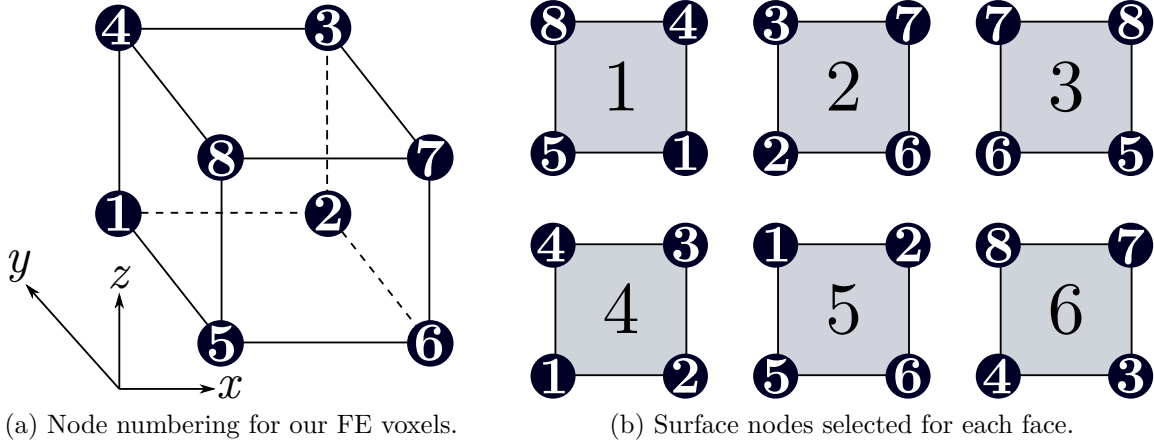


Figure 7: Surface element nodes must be self-consistent so the forces have the correct sign.

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,  $\theta_{\max}$  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

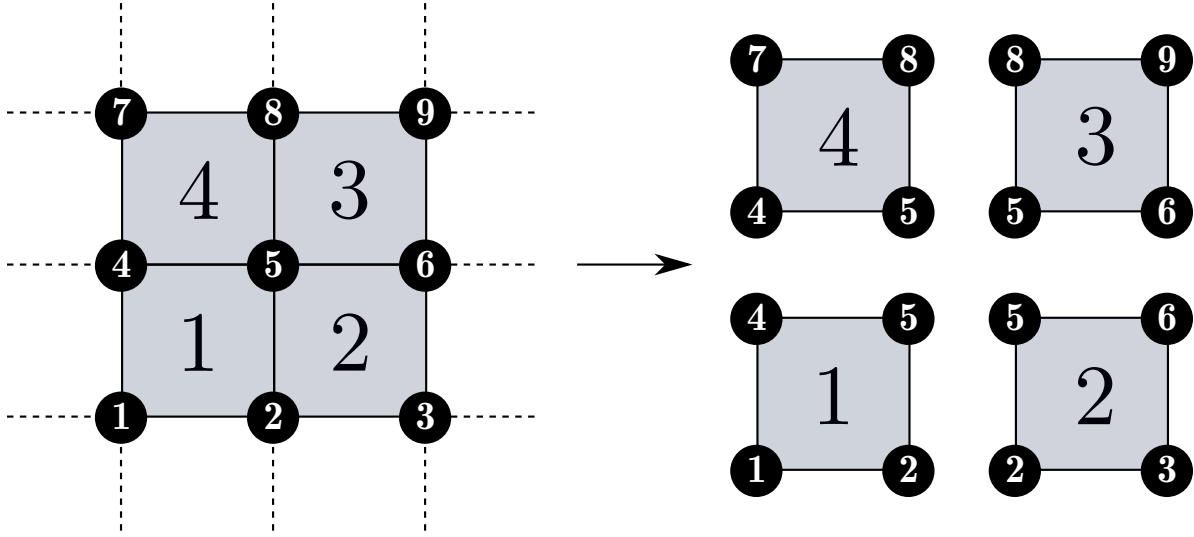


Figure 8: FE nodes are shared by either 4 element faces or 3 if it is a corner node. The total force on a given node is the summation of the force contributions from each element it belongs to.

done by defining,

$$\mathbf{x}_{e,n} \equiv [x_{e,n} \quad y_{e,n} \quad z_{e,n}]^T, \quad \mathbf{x}_n \equiv [x_n \quad y_n \quad z_n]^T \quad (26a)$$

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

$$\mathbf{F}_e = \begin{bmatrix} \mathbf{x}_{1,1} & \mathbf{x}_{1,2} & \mathbf{x}_{1,4} & \mathbf{x}_{1,5} \\ \mathbf{x}_{2,2} & \mathbf{x}_{2,3} & \mathbf{x}_{2,5} & \mathbf{x}_{2,6} \\ \mathbf{x}_{3,5} & \mathbf{x}_{3,6} & \mathbf{x}_{3,8} & \mathbf{x}_{3,9} \\ \mathbf{x}_{4,4} & \mathbf{x}_{4,5} & \mathbf{x}_{4,7} & \mathbf{x}_{4,8} \end{bmatrix}, \quad \tilde{\mathbf{F}} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_9 \end{bmatrix}. \quad (26c)$$

Where  $\mathbf{x}_{e,n}$  is a  $3 \times 1$  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.  $\mathbf{x}_n$  is a  $3 \times 1$  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  $\tilde{\mathbf{F}}$  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  $\mathbf{n}$ ,  $\mathbf{p}$ ,  $\mathbf{q}$  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.  $\tilde{\mathbf{F}}$  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. Our implementation does not strictly follow algorithm 1 because we memoise a

---

**Algorithm 1** Assuming  $\tilde{\mathbf{F}}$  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,  $\mathbf{L}$ , with the linearised indices in  $\mathbf{N}_L$  where
       node  $n$  appears as part of a surface element whose tractions we are calculating.
6:      $\mathbf{L} \leftarrow \text{find}(\mathbf{N}_L == n)$ 
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:     $\tilde{\mathbf{F}}[3n + k] \leftarrow \tilde{\mathbf{F}}[3n + k] + \sum \mathbf{F}_e[3\mathbf{L} + k]$ 
11:    end for
12: end for
```

---

generalised version of  $\mathbf{L}$  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 later used in equation (5) as  $\tilde{\mathbf{T}} \equiv \tilde{\mathbf{F}}$ . Figure 9 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, singular image stresses found in equations (20a), (21a) and (22a). We use three test cases, two edge dislocations and one screw, all of which have line direction  $\mathbf{l} = [001]$  with Burgers vectors  $\mathbf{b}_{e1} \equiv \mathbf{b} = [100]$ ,  $\mathbf{b}_{e2} \equiv \mathbf{b} = [010]$  [17], and  $\mathbf{b}_s \equiv \mathbf{b} = \mathbf{l} = [001]$  [18]. The slices we took for our contour plots in section 4 taken from the midpoint of the  $z$ -axis

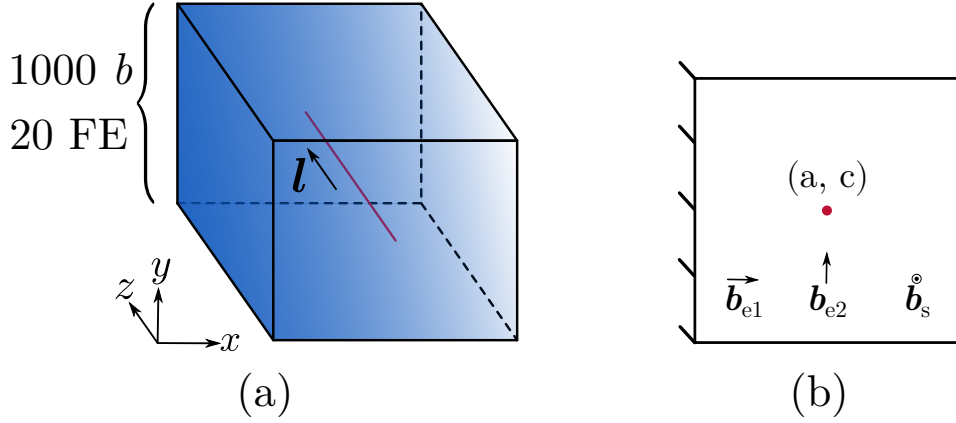


Figure 9: Dislocation parallel to a surface described by the line  $x = 0$  (the  $yz$ -plane in 3D), where the dislocation coordinates are  $(a, c)$ , looking down the  $xy$ -plane where the  $x$ -axis is horizontal and the  $y$ -axis vertical. (a) describes the box we used for our comparison, a  $20 \times 20 \times 20$  element cubic box with side lengths equal to  $1000 \|\mathbf{b}\|$  with  $\mathbf{T} = \mathbf{0}$  traction boundary conditions only on the nodes occupying the  $yz$ -plane and  $\mathbf{U} = \mathbf{0}$  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],  $\mathbf{b}_{e1} \equiv \mathbf{b} = [1\ 0\ 0]$ ,  $\mathbf{b}_{e2} \equiv \mathbf{b} = [0\ 1\ 0]$  and the screw Burgers vector  $\mathbf{b}_s \equiv \mathbf{b} = [0\ 0\ 1]$  as posited in [18].

to ensure minimal interference from the  $xy$ -plane boundaries at  $z = 0, 1000b$ .

Finally, we ran a simple simulation where the differences between methods can be readily observed.

#### 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 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 points 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 10 (a) and (b).

The limitations of Gauss quadrature become even more evident when the dislocation line segment is parallel to a surface element. Figure 11 shows the relative errors for a parallel edge dislocation. In figure 11, we observe the relative errors decrease

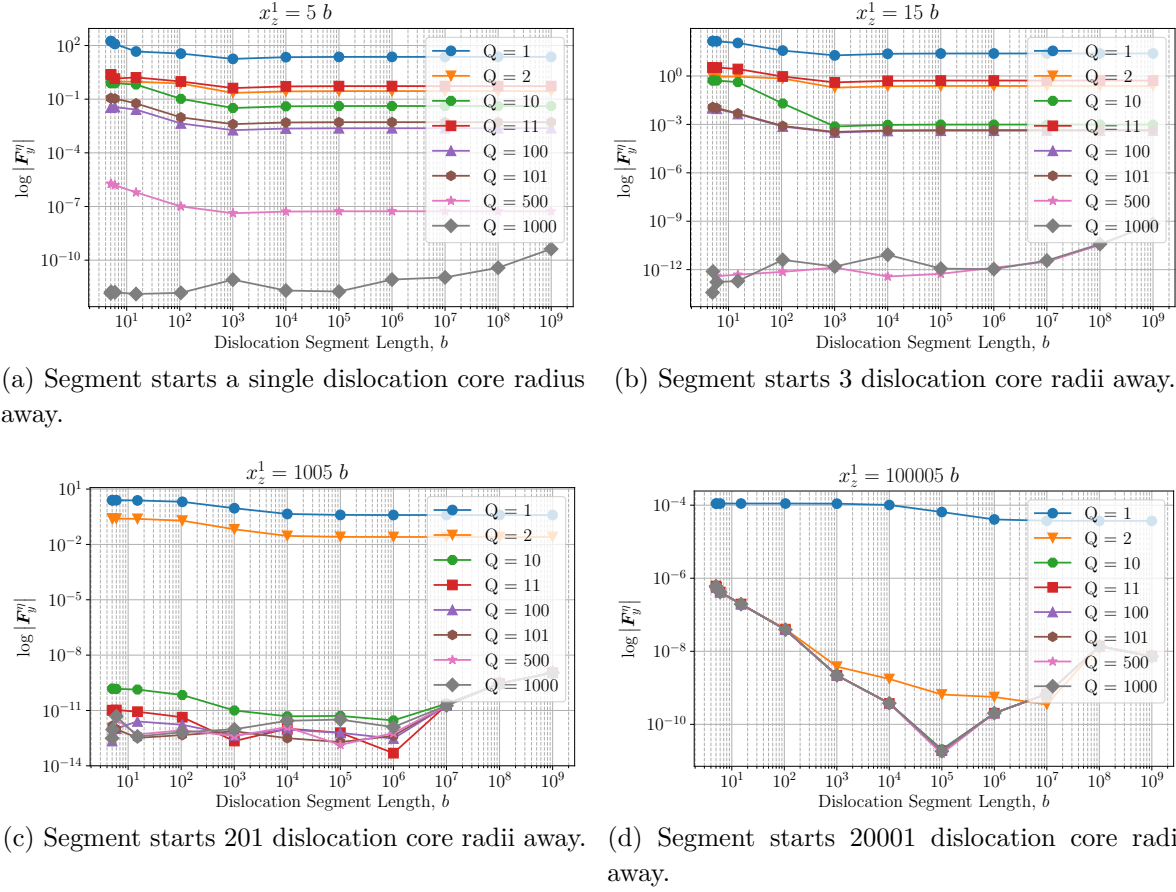


Figure 10: Log-log plot of the relative error as a function of dislocation segment length for a perpendicular edge dislocation (figure 5a).  $x_z^1$  is the  $z$ -coordinate of node  $\mathbf{x}_1$ .  $\mathbf{b} = [100]$ , line direction,  $\mathbf{l} = [001]$ , and dislocation core radius,  $a = 5b$ , the surface element's normal and size are,  $\mathbf{n} = [001]$ ,  $L = 1000b$ , respectively.  $Q$  is the number of quadrature points per dimension.

and then converge to  $\sim 1$  as the distance grows larger. The convergence to  $\sim 1$  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, resulting in the division of two increasingly similar, small, non-zero numbers, resulting in  $\sim 1$  when done with limited precision. Given infinite precision, it is not unreasonable to assume the relative error would converge to  $\sim 10^{-3}$ ,  $\sim 10^{-4}$ †. However, when the segment is close to the surface, the relative errors are 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  $x_z^1 = 15b$ . This figure backs up the earlier point regarding Gauss nodes close to maximal rational values, as  $Q = 2$  performs significantly better than  $Q = 10, 11, 100, 100$  at

† The absolute error *always* decreases as the distance from the surface element increases, but absolute errors are not as useful tools of comparison as long as we are conscious of small artifacting from limited precision.

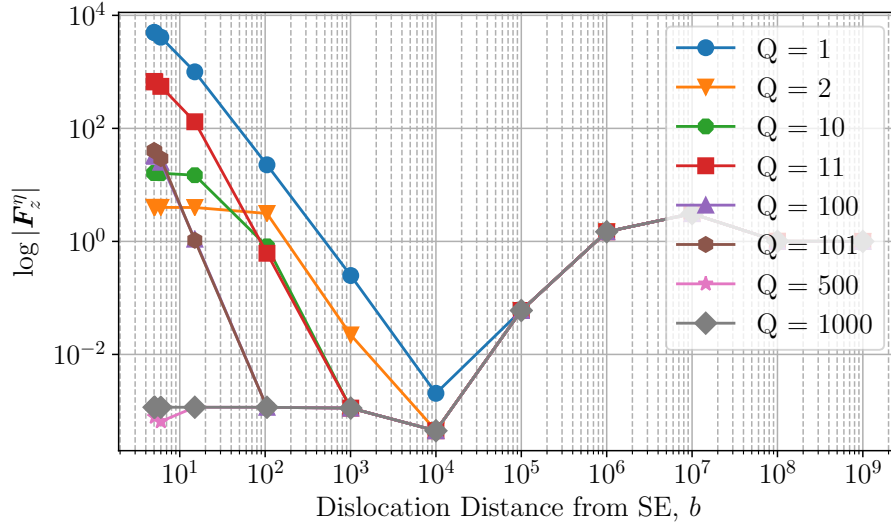


Figure 11: Log-log plot of the relative error as a function of distance from the surface element for a parallel edge dislocation (figure 5b). Burgers vector,  $\mathbf{b} = [010]$ , line direction,  $\mathbf{l} = [100]$ , dislocation core and surface element parameters are the same as figure 10. The 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.

smaller distances from the surface. Generally, there is no best value of  $Q$  other than very large.

From figures 10 and 11 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 in figure 5(a) we find figure 12. The  $\sigma_{xx}$  and  $\sigma_{zz}$  components are antisymmetric about the centre of the element. If we use Gauss quadrature on them, we sample equivalent but oppositely valued points that equally weighted, thus the sum vanishes.  $\sigma_{yz}$  does not, but it can be accurately integrated with sufficiently large  $Q$ . If we were to move the dislocation off-centre such that the rotational anti-symmetry about the origin is broken, the errors would increase.

Despite these being artificially ideal examples that illustrate the failings of Gauss quadrature, other problematic scenarios commonly to show up in simulations. The problems tend to worsen with smaller core radii  $a$ , fewer Gauss nodes, higher dislocation densities near surfaces and more permissive mobility functions. Despite the highly



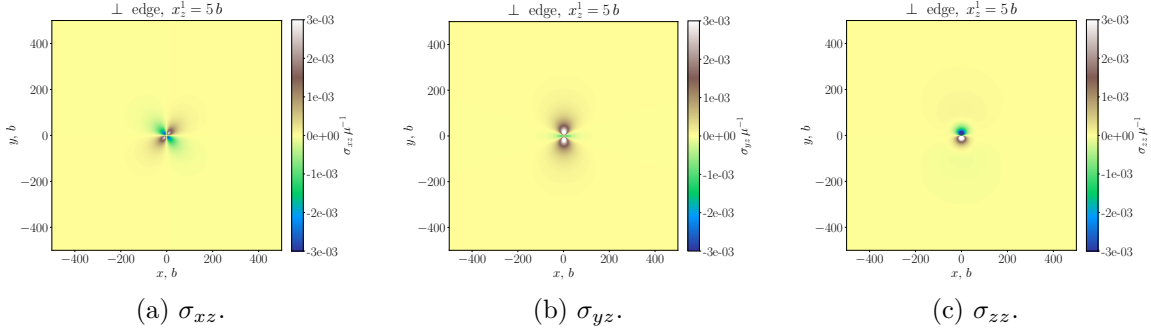


Figure 12: Relevant stress components to figure 10. The antisymmetry of  $\sigma_{xz}$  and  $\sigma_{zz}$  makes them vanish when integrating.  $\sigma_{yz}$  does not vanish, but its symmetry means Gauss quadrature works if sufficient quadrature points are used.

localised nature of these large errors, the  $\mathcal{O}(1/R)$  decay rate and chaotic nature of dislocation dynamics, means they often result in unwarranted topological changes. Such changes to the dislocation structure tend to have large and cascading effects as simulations advance. Which is 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 (20a), (21a) and (22a)<sup>‡</sup>. The stress field comparisons for all three cases are found in figures 13 to 15, where the dislocation is denoted by a white dot.

Figure 13 shows the stress fields corresponding to analytic expressions for image stress components,  $\hat{\sigma}_{ij}$  in equation (20a) where no superscript denotes the infinite-domain singular expressions, a A superscript are the stresses derived from analytic tractions and N for those coming from numeric ones. The setup corresponds to the one described in figure 9 where  $\mathbf{b} = \mathbf{b}_{e1}$ . It is clear edge effects play a role in the generated stresses. All three components have notable differences from the infinite-domain solutions resulting from the finite constraints. Regardless, the overall shape of the isolines is similar in all three solutions for corresponding stress components, particularly  $\hat{\sigma}_{yy}$ . The lobes of  $\hat{\sigma}_{xx}$  for analytic and numeric tractions are deformed by the displacement boundary conditions on all surfaces other than  $x = 0b$ , where tractions are calculated. The effect is most dramatic when looking near the top and bottom of the domain, where the final isolines flare towards the traction surface. The shear stresses  $\hat{\sigma}_{xy}$  are also somewhat different, with similar flaring. However, as they radiate in the  $x$ -direction, the deviations away from the infinite-domain solutions flare away from the

<sup>‡</sup> Since the first term in each equation corresponds to the real stresses, we omit them to view the image stresses.

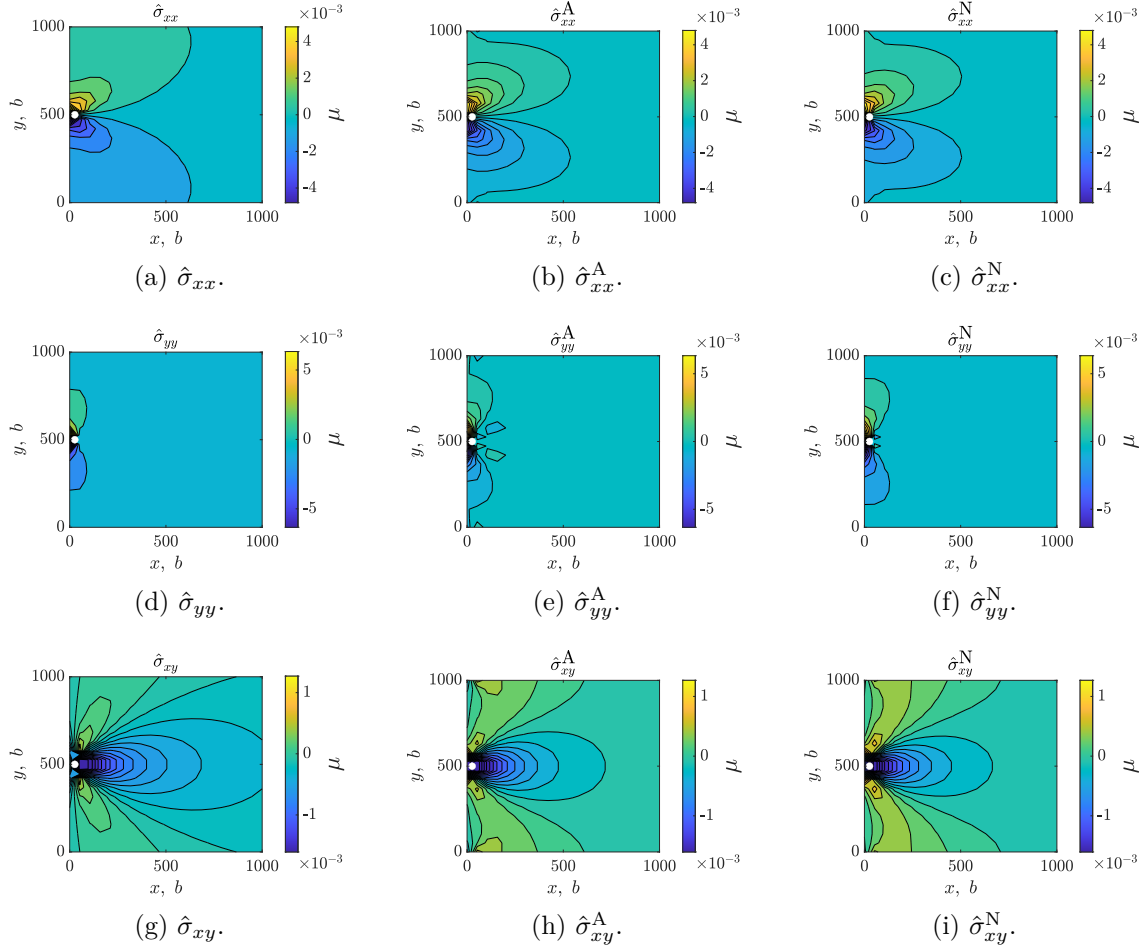


Figure 13: Image stresses for an edge dislocation with  $\mathbf{l} = [001]$ ,  $\mathbf{b} = [100]$ , with  $a = 10b$  with coordinates  $(26.3158, 500)b$ , i.e. the centre of the first FE from the surface at  $x = 0$ , and in the centre of the simulation box from top to bottom. Here  $b = \|\mathbf{b}\| \equiv 1$ .

$x = 1000b$  surface. That said, all stress fields calculated from analytic and numeric tractions look fairly similar for  $\mathbf{b} = \mathbf{b}_{e1}$ .

Things get markedly more interesting when looking at  $\mathbf{b} = \mathbf{b}_{e2}$  in figure 14. Of particular note is  $\hat{\sigma}_{xx}$ , where a comparison between figures 14a and 14b and figure 14c reveals one of the major issues with numeric tractions. If we look at the neighbourhood of the dislocation, we will find a sign inversion as well as a drastically different isosurface shape, where the isolines are not merely deformed, they are fundamentally different when calculated with numeric tractions. Stresses like those can lead to dislocations problems, particularly if it just so happens that the sign inversion also comes with an increased force magnitude, something that is possible judging from figure 11. 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 inverts the Archimidean principle. The other two stress fields  $\hat{\sigma}_{yy}$ ,  $\hat{\sigma}_{xy}$  follow similar trends to what we have already observed, where the isolines get squashed and flair towards the traction free surface.

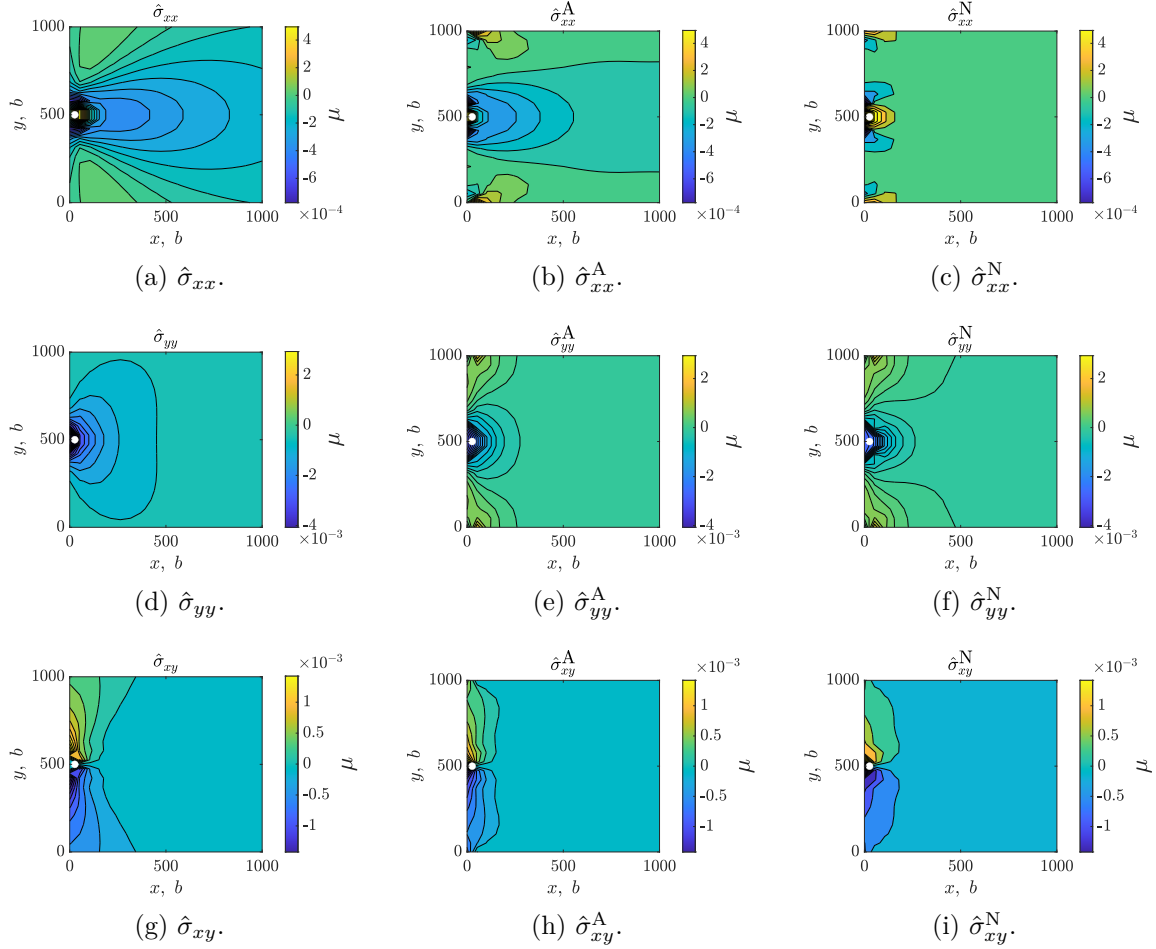


Figure 14: Image stresses for an edge dislocation with  $\mathbf{l} = [001]$ ,  $\mathbf{b} = [010]$ . The rest is the same as figure 13.

Perhaps the most evident deformation resulting from the displacement boundaries can be seen when  $\mathbf{b} = \mathbf{b}_s$  in figure 15, where the lobes of the isolines are highly deformed when compared to the infinite-domain solutions. The familiar lobes look very different, but both analytic and numeric tractions yield fairly similar fields.

From figures 13 to 15 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. Although these can have significant consequences later in a simulation's runtime, and particularly in simulations where dislocations regularly come into close contact with surfaces.

It is also worth looking at the stress fields with a finer lens. Figure 16 shows line plots through the first set of nodes away from the traction surface, i.e. through the line  $x = 52.6316b$ . Figures 16a to 16c correspond to  $\mathbf{b} = \mathbf{b}_{e1}$ , figures 16d to 16f  $\mathbf{b} = \mathbf{b}_{e2}$  and figures 16g and 16h to  $\mathbf{b} = \mathbf{b}_s$ . Here the singular nature of the infinite-domain solutions is evidenced by the sharp spikes in the stresses. In general, the non-singular formulation smoothes out the stress line plots. However, there are a few instances where the numeric

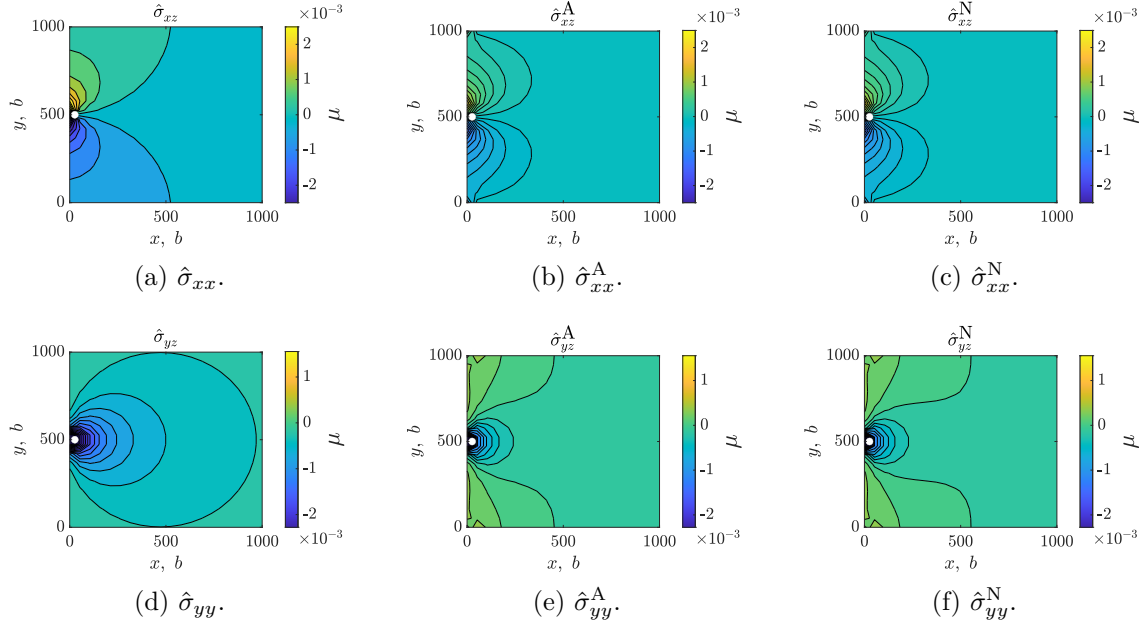


Figure 15: Image stresses for a screw dislocation with  $\mathbf{l} = [001]$ ,  $\mathbf{b} = [001]$ . The rest is the same as figure 13.

tractions lead to larger spikes than even the infinite-domain solutions. Again, the general shape of the line plots is the same but the tendency for numerical tractions to spike under specific circumstances is evident in every case.

We also show examples of contour figure 18 and line plots figure 18 of the image and total stress fields as an edge dislocation with  $\mathbf{b} = \mathbf{b}_{e2}$  moves from  $26.3158b$  (centre of the first element) to  $500b$  (centre of the 10th element). The line plots are taken from the closest set of nodes in the positive  $x$ -direction i.e. at  $x = 52.6316 \ 526.3158b$ .

From the line plots, it can be observed that both analytic and numeric tractions converge to similar shapes to each other, which is to be 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. However, the glaring issue is that total stresses for each solution are significantly different as dislocations near the surface. The instability of Gauss quadrature when integrating rational functions tends to produce values that are outside the range of the infinite-domain solution, as well as being less smooth than the two other solutions.

As expected, all solutions converge as the distance from the dislocation to surface increases because the magnitude of the real stresses increases in relation to that of the image stresses§.

In order to show the effects numerical tractions have on simulations, we ran a

§ The image stress components of equations (20a) and (21a) are reflections about the line  $x = 0$  plus a corrective term that decays faster than the reflection term. In equation (22a), the image term is a reflection about the line  $x = 0$ . Therefore in both cases, as the dislocation moves away from the surface, the real stresses get relatively larger when compared to the image ones.

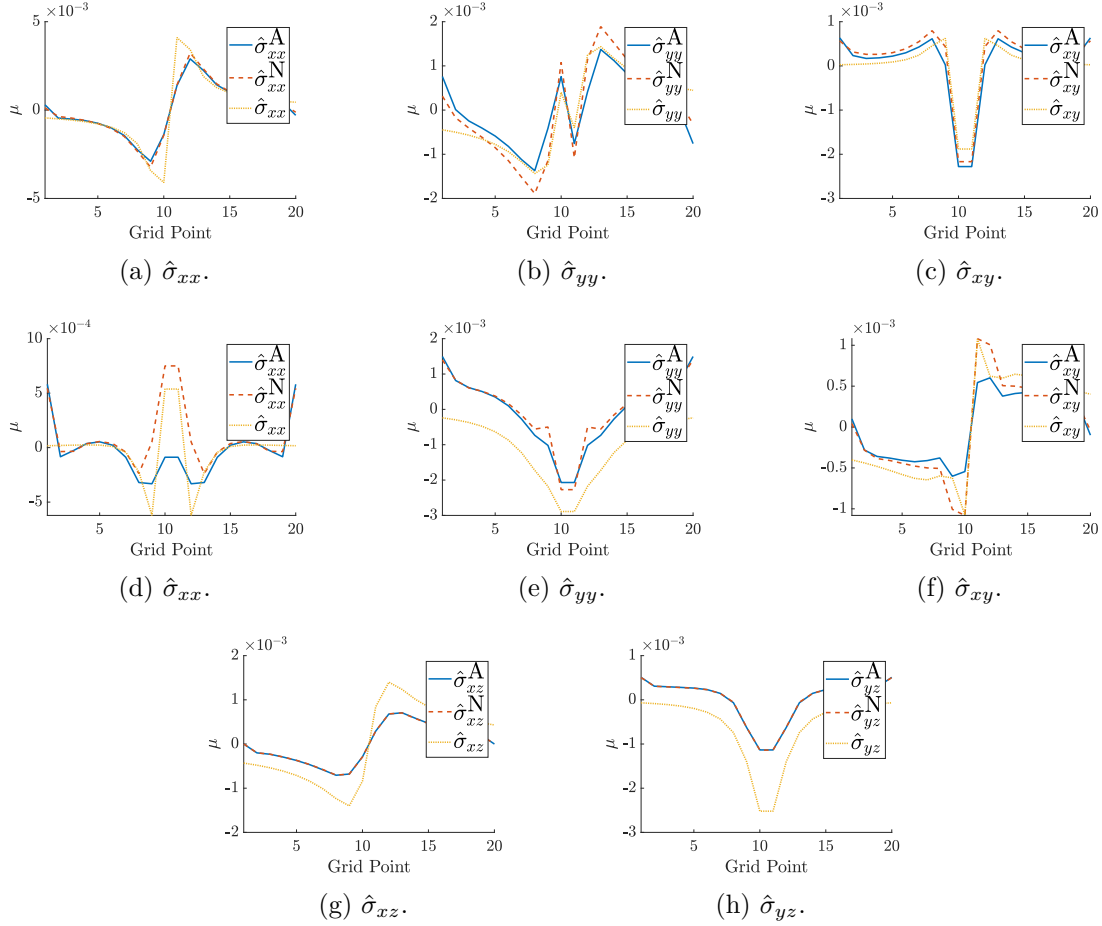


Figure 16: Line plots of the image stresses at  $x = 52.6316b$  for the analytic image stresses, as well as those calculated with numeric and analytic tractions. The line corresponds to the first set of nodes away from the boundary at  $x = 0b$ . (a) to (c) correspond to  $\mathbf{b} = \mathbf{b}_{e1}$ , (d) to (f)  $\mathbf{b} = \mathbf{b}_{e2}$  and (g) to (h) to  $\mathbf{b} = \mathbf{b}_s$ .

very simple simulation where the only difference was the traction calculation method. A very simple yet illustrative qualitative example is to produce a scatter plot of the numerically calculated tractions vs analytically calculated ones at any time during any simulation. To obviate the problem even more, it helps making the axes proportional. What is observed in practically every case, is a positive correlation (though as we have seen in figures 14a to 14c, numerical tractions can produce sign inversions under specific conditions), but the axis that represents the numerically calculated tractions will be visually larger, as numeric tractions produce more extreme values. Statistically speaking, the closer the  $R$ -value is to 1 and the closer a linear fit is to  $y = x$ , the more accurate numerical tractions are for any given moment in a simulation. We have yet to find a single example where this happens. Furthermore, the chaotic nature of dislocation dynamics means that slight deviations will yield noticeably different results for the same starting conditions.

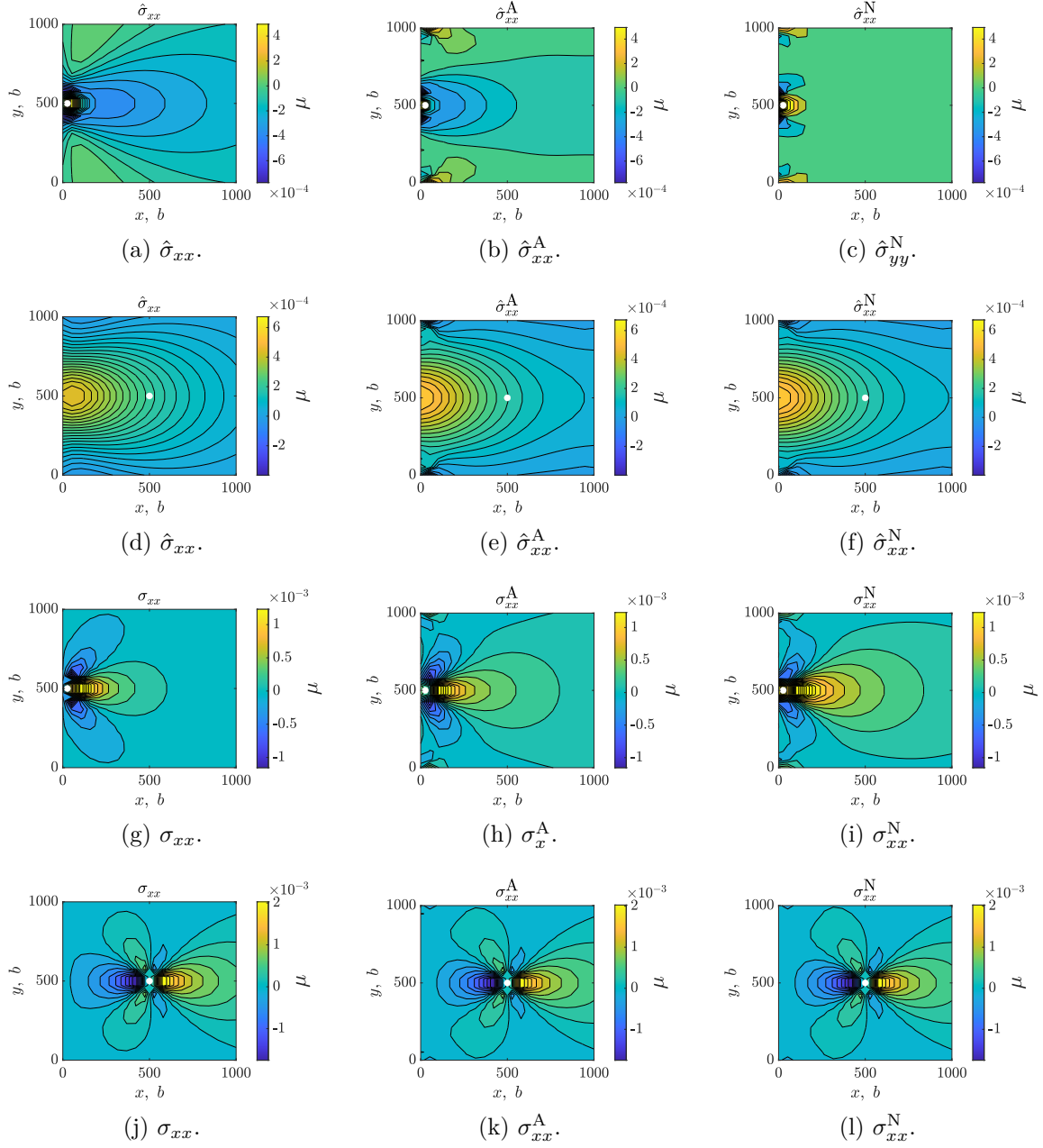


Figure 17: Image stresses for an edge dislocation with  $\mathbf{l} = [001]$ ,  $\mathbf{b} = [010]$ . (a) to (c) and (g) to (i) at  $(26.3158, 500)b$ ; (d) to (f) and (j) to (l) at  $(500, 500)b$ . (a) to (g) image stresses; (g) to (l) total stresses.

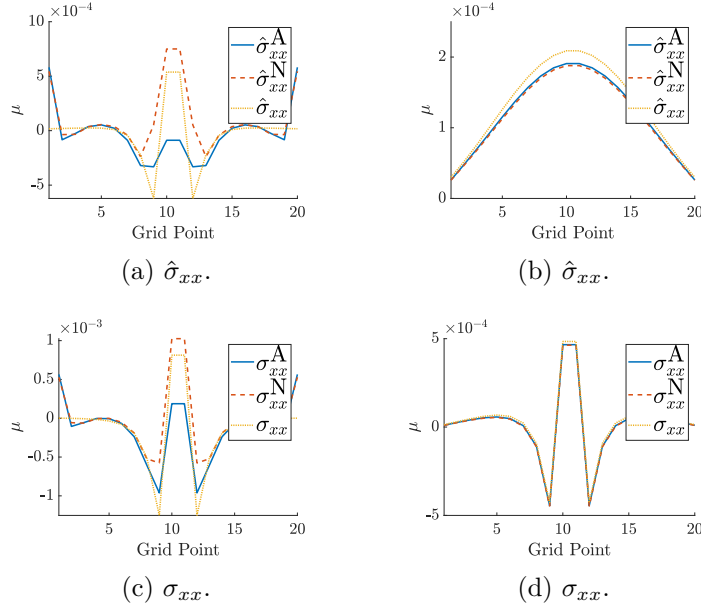


Figure 18: Line plots corresponding to figure 17. (a) and (c) are of image stresses; (b) and (d) are of total stresses. (a) and (b) are of a dislocation at  $(26.3158, 500)b$ , taken along the line  $x = 52.6316b$ . (b) and (d) are of a dislocation at  $(500, 500)b$  taken along the line  $x = 526.3158b$ .

This does not mean numeric tractions are useless, in fact, they mostly work quite well. However, the impact their inherent numerical instability can be quite large. For us, analytic tractions despite being approximately an order of magnitude slower than numerically calculated tractions with  $Q = 1$  [14] (our tests agree with their findings), yield more stable and ultimately faster simulations. Even simple simulations are typically faster when using analytic tractions over numeric ones. The margin grows the more complex the simulation and the further each simulation gets. It's all down to better behaved segments at or near surfaces. As such, 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.

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

We would like to thank Prof. Sylvain Queyreau and Lawrence Livermore National Laboratory for their invaluable input. This work was supported by the Consejo

Nacional de Ciencia y Tecnologia, Fondo Sectorial CONACYT-Secretaria de Energia-Sustentabilidad Energetica Cuarto Periodo [291129]. This work was supported by the Engineering and Physical Sciences Research Council Centre for Doctoral Training in the Science and Technology of Fusion Energy EP/L01663X/1 and Fellowship grant EP/N007239/1.

## 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.