PHYSICAL REVIEW E VOLUME 52, NUMBER 3 SEPTEMBER 1995

Order-N method for force calculation in many-dislocation systems

D. B. Barts and A. E. Carlsson
Department of Physics, Washington University, St. Louis, Missouri 63130

(Received 9 February 1995)

We propose a methodology for calculating the elastic interaction forces of a large ($N \sim 10^6$ or more) number of dislocations in a solid, for the purposes of molecular-dynamics type modeling of problems such as dislocation patterning and strain hardening. In order to avoid the N^2 scaling resulting from the $1/r$ decay of the forces, we introduce a variation of the particle-particle–particle-mesh method of Hockney and Eastwood [*Computer Simulations Using Particles* (Institute of Physics, Bristol, 1988)] , applicable to arrangements of straight, parallel dislocations; extension to fully three-dimensional problems is discussed. Our version handles the anisotropic interdislocation forces with high accuracy and is easy to implement. Precision estimates are obtained; they indicate a very favorable scaling of CPU time with accuracy, which is supported by numerical results.

PACS number(s): 02.70.Ns, 61.72.Lk, 62.20.Fe

I. INTRODUCTION

Numerical simulation of many-dislocation systems is a promising approach to the problems of understanding and predicting dislocation patterning, and obtaining constitutive relations for materials from the underlying dislocation-dislocation interactions. For a review, see Kubin [1] and references therein. In these simulations, one typically uses the long-ranged elastic interaction between dislocations, together with some assumed rules about interactions, annihilations, and crossings at short distances. It has been shown both numerically [2] and theoretically [3] that in order to avoid nonphysical phenomena such as spurious patterning wavelengths, one must retain the whole long-ranged tail of the elastic interactions. This makes direct force calculation an order-N^2 process; indeed, in previous work, N has been limited to the order of 1000. While this is adequate for some two-dimensional problems, it is clearly insufficient in three dimensions. For example, if one represents a dislocation line as a train of 50 straight segments, then treating 500 dislocation lines would entail the computation of approximately 3×10^8 interactions for each time step in a simulation. Obviously, one needs an algorithm which is faster than $O(N^2)$. The above-mentioned observations of spurious patterning suggest that high accuracy in the computation of the forces is important. At the same time, the mathematical complexity of the dislocation interaction forces in three dimensions implies that a comparatively simple and straightforward method of obtaining $O(N)$ methods is needed.

Several $O(N)$ methods, in which the CPU time is proportional to only the first power of N despite the long range of the interaction, have been developed to treat elastic forces, as well as the mathematically analogous problems of Coulomb and gravitational interactions. See Ref. [4] for a discussion. These methods generaly fall into the categories of multipole expansions or "particle-particle—particle-mesh" (P^{3}M) methods. In the multipole methods, computational speed is achieved by "lumping" dislocations distant from a reference dislocation (or particles) into relatively large cells; the contribution from the distant dislocations is then obtained from the multipole moments of these cells, rather than individual treatment of the distant dislocations. At a given order of multipole expansion, the error is inversely proportional to a power of the CPU time. Higher powers can be achieved by going to higher and higher order in the multipole expansion, which can be mathematically cumbersome. Very recently, this method has been applied to the dislocation interaction problem treated here [5].

The P^{3}M method [6] is based on the Ewald principle of dividing the interaction force into a short-ranged part and a long-ranged part which is smooth in the sense that its Fourier transform decays rapidly in k space. The short-ranged part of the interaction is made $O(N)$ straightforwardly by the use of neighbor lists. The smooth part of the force is treated in several steps. First, the particle positions are converted into a particle density that is defined on a mesh. Second, using a combination of the convolution theorem of Fourier analysis and a fast Fourier transform method, the mesh particle density is converted into a mesh force density. Finally, the mesh force density is converted into forces on individual particles. Finite-ranged spreading functions are used to obtain the mesh particle density, and a finite-ranged interpolation is used to obtain the individual particle forces from the mesh forces. The accuracy of the method has a power-law behavior as a function of CPU time. Higher powers can be achieved by going to progressively higher- and higher-order spreading and interpolation functions. To our knowledge, the P^{3}M method has been implemented only for isotropic forces, which are appropriate for screw dislocations but not for edge dislocations. In fact, as is clear from the discussion of Ref. [5], it is not generally realized that the P^{3}M method can be applied to dislocation interactions.

Here we present and study a variation of the P^{3}M method which is straightforwardly applied to anisotropic forces, and has significant advantages over the multi-

1063-651X/95/52(3)/3195(10)/$06.00

pole method and existing implementations of the P^{3}M method. The method consists of using spreading and smoothing functions which are polynomials of a Gaussian function. The order of the polynomial depends on the anisotropy of the elastic interaction, and is, for example, 2 in the case of edge dislocations and 1 in the case of screw dislocations. This method leads to improvements both in performance and implementability. With regard to performance, the method can be thought of as a P^{3}M method which is "infinite order" in the sense that the error converges faster than any power of the CPU time (in fact, it is a decaying Gaussian function of the CPU time). This is more favorable then either the multipole method or existing implementations of the P^{3}M method. With regard to implementation, the formulas are quite simple, and improving the accuracy of the method is simply a matter of "turning a knob" by varying the length scale of the spreading and interpolating Gaussians, which is much simpler than, for example, adding another order to a multipole expansion. As presented here, the method is appropriate for the calculation of forces in two-dimensional dislocation arrays with periodic boundary conditions. However, generalization to three dimensions appears to be straightforward, and nonperiodic boundary conditions can be handled by existing methods [7] for removing periodic replicas.

The organization of the remainder of the paper is as follows. Section II presents the basic methodology for a simplified case in which the Burgers vectors are restricted and only certain components of the force are calculated. Section III describes the numerical implementation, provides estimates for the error and CPU time in terms of fundamental parameters of the method, and gives examples of both for specific dislocation configurations. Section IV outlines the extension to arbitrary Burgers vectors and force components. Section V gives concluding comments and discusses the extension to three dimensions.

II. DESCRIPTION OF THE METHOD

We treat the problem of N dislocations in a square repeated cell of size L. Since there is no interaction between edge and screw dislocations in two dimensions, we consider systems of dislocations of the same kind: either all edges or all screws. Throughout this paper, dislocation lines are considered to be parallel to the z axis.

In this section and the following one we give a detailed treatment of the simplest case, considering only the x component of the force, with all Burgers vectors in the x direction for edge dislocations and in the z direction for screw dislocations:

$$\mathbf{b}_{\rm scr} = (0,0,b) \ ,$$
$$\mathbf{b}_{\rm edge} = (b,0,0) \ . \qquad (2.1)$$

In Sec. IV we explain how to relax this restriction and calculate all the components of the force for arbitrary Burgers vectors.

The total force on the ith dislocation from all the rest is

$$F^i = {\sum_j}' \mathcal{F}_{ji} \ , \qquad (2.2)$$

where the force that dislocation j exerts on i is $\mathcal{F}_{ji} = b_i b_j f(\mathbf{r}_{ij})$, the usual elastic interaction which decays as $1/r$ and is angle dependent, and the prime indicates that $j = i$ terms are absent.

To implement the P^{3}M approach, represent f as

$$f(\mathbf{r}) = f_{sr}(\mathbf{r}) + f_{lr}(\mathbf{r}) \ , \qquad (2.3)$$

where both $f_{sr}(\mathbf{r})$ and $\tilde{f}_{lr}(\mathbf{k})$ asymptotically have Gaussian decay as functions of their respective arguments (details of this splitting are given below). Denote the cutoff length of f_{sr} by $r_{\rm max}$, i.e., $|f_{sr}(\mathbf{r})|_{r>r_{\rm max}}$ is less than the desired precision. Equation (2.2) can then be rewritten as

$$F^i = b_i {\sum_j}' b_j \left[f_{sr}(\mathbf{r}_{ij}) + f_{lr}(\mathbf{r}_{ij})\right] \equiv F^i_{sr} + F^i_{lr} \ , \quad (2.4)$$

where

$$F^i_{sr} = b_i {\sum_{r_{ij}\le r_{\rm max}}}' b_j f_{sr}(\mathbf{r}_{ij}) \qquad (2.5)$$

and we write the long-range force contribution in the form

$$F^i_{lr} \equiv b_i S(\mathbf{r}_i) = b_i \int \rho(\mathbf{r}) f_{lr}(\mathbf{r}_i - \mathbf{r}) d\mathbf{r} \ , \qquad (2.6)$$

where

$$\rho(\mathbf{r}) = \sum_{j=1}^{N} b_j \delta(\mathbf{r} - \mathbf{r}_j) \qquad (2.7)$$

is the density of Burgers vectors. $S(\mathbf{r})$ is now a stress function defined for continuous argument $\mathbf{r}$ (it is equal to σ_{yx} or σ_{yz} in the cases of edge or screw dislocations, respectively).

Our procedure for evaluating the sum in Eq. (2.5) is indicated in Fig. 1. One could divide the simulation box into cells of size $a = r_{\rm max}$ and consider the interactions of all dislocations in cell C_n with those of neighboring cells (and C_n itself). The average number of interactions per dislocation would then be $\bar{\rho}\, 9 r^2_{\rm max}/2$, where $\bar{\rho}$ is the average density. Alternatively, one can choose $a \ll r_{\rm max}$ and consider all cells within $r_{\rm max}$ of C_n, with an average of $\bar{\rho}\pi r^2_{\rm max}/2$ interactions per dislocation; this method would in principle be nearly three times faster. However, going through the cells numerically takes time even if they are underpopulated; so it is not favorable to have too many of them. We choose $a = r_{\rm max}/4$ (see Fig. 1). As $\mathcal{F}_{ij} = -\mathcal{F}_{ji}$, we need to calculate the interactions of each dislocation pair only once, so we consider only the forces between cell C_n and those to the right of it.

The integral in Eq. (2.6) is a convolution of ρ and f_{lr} and can therefore be efficiently evaluated in Fourier space:

$$\tilde{S}(\mathbf{k}) = \tilde{\rho}(\mathbf{k}) \tilde{f}_{lr}(\mathbf{k}) \ , \qquad (2.8)$$

where the Fourier transforms are normalized as follows:

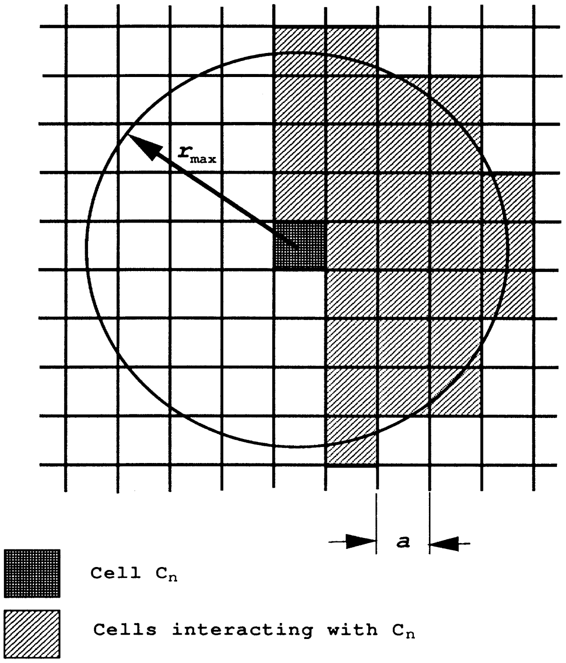

FIG. 1. Calculation of the short-range forces.

$$\tilde{f}(\mathbf{k}) = \int f(\mathbf{r}) e^{-i\mathbf{k}\cdot\mathbf{r}} d^2r, \quad f(\mathbf{r}) = (2\pi)^{-2} \int \tilde{f}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}} d^2k\,.$$

An obvious way to implement Eq. (2.8) would be to evaluate

$$\tilde{\rho}(\mathbf{k}) = \sum_{j=1}^{N} b_j e^{-i\mathbf{k}\cdot\mathbf{r}_j} \qquad \text{(for each value of } \mathbf{k}) \qquad (2.9a)$$

and

$$S(\mathbf{r}_i) = L^{-2} \sum_{\{\mathbf{k}\}} e^{i\mathbf{k}\cdot\mathbf{r}_i} \tilde{\rho}(\mathbf{k}) \tilde{f}_{lr}(\mathbf{k}), \qquad i = 1, \ldots, N\,. \qquad (2.9b)$$

However, performing each of these two steps would involve a number of operations proportional to the product of the number of particles (N) and the number of $\mathbf{k}$ values $[O(N)]$. Thus Eqs. (2.9) correspond to an $O(N^2)$ procedure and are unsuitable for numerical implementation.

Therefore, we instead use the P^{3}M strategy of transferring the Burgers vector density and stress density to a mesh. For this purpose, we introduce functions $S^{\text{mesh}}(\mathbf{r})$ and $\rho^{\text{mesh}}(\mathbf{r})$ as follows:

$$\rho^{\text{mesh}}(\mathbf{r}) = \frac{1}{2\pi r_1^2} \int \rho(\mathbf{r}') e^{-(r-r')^2/2r_1^2} d\mathbf{r}'\,, \qquad (2.10a)$$

$$S(\mathbf{r}) = \frac{1}{2\pi r_2^2} \int S^{\text{mesh}}(\mathbf{r}') e^{-(r-r')^2/2r_2^2} d\mathbf{r}'\,, \qquad (2.10b)$$

which corresponds to

$$\tilde{\rho}^{\text{mesh}}(\mathbf{k}) = e^{-\frac{1}{2}k^2 r_1^2} \tilde{\rho}(\mathbf{k})\,, \qquad (2.11a)$$

$$\tilde{S}^{\text{mesh}}(\mathbf{k}) = e^{+\frac{1}{2}k^2 r_2^2} \tilde{S}(\mathbf{k})\,; \qquad (2.11b)$$

$\tilde{S}(\mathbf{k})$ will always decay rapidly enough so that $\tilde{S}^{\text{mesh}}(\mathbf{k})$ has a well-defined Fourier transform. Here r_1 and r_2 are spreading parameters. Precision estimates for the numerical convolution and constraints on r_1 and r_2 are presented in Appendix B.

Substituting Eqs. (2.11) into Eq. (2.8) we obtain

$$\tilde{S}^{\text{mesh}}(\mathbf{k}) = \tilde{\rho}^{\text{mesh}}(\mathbf{k}) \tilde{f}_{lr}(\mathbf{k})\, e^{\frac{1}{2}k^2(r_2^2+r_1^2)}\,. \qquad (2.12)$$

We define one more function

$$\tilde{f}^{\text{mesh}}(\mathbf{k}) = \tilde{f}_{lr}(\mathbf{k})\, e^{\frac{1}{2}k^2(r_2^2+r_1^2)} \qquad (2.13)$$

[the choice of parameters will always ensure that $\tilde{f}^{\text{mesh}}(\mathbf{k}) \to 0$ as $k \to \infty$]. Equation (2.12) takes the form

$$\tilde{S}^{\text{mesh}}(\mathbf{k}) = \tilde{f}^{\text{mesh}}(\mathbf{k}) \tilde{\rho}^{\text{mesh}}(\mathbf{k})\,. \qquad (2.14)$$

Let us denote the size of the real-space mesh by A and its mesh points by $\mathbf{R}_m$. The corresponding k-space mesh [8] has spacing $2\pi/L$ and is bounded by

$$|\mathbf{k}_m^x| \le k_{\max}, \qquad |\mathbf{k}_m^y| \le k_{\max} \qquad (2.15)$$

with $k_{\max} = \frac{\pi}{A}$.

The numerical procedure for implementing this methodology will be determined by the precision parameter ϵ. Each time we make an approximation, we require that the resulting error be of order ϵ. Our procedure is linear, and therefore we expect the overall error to be of order ϵ as well.

The mesh version of Eqs. (2.10) is

$$\rho^{\text{mesh}}(\mathbf{R}_m) = \frac{1}{2\pi r_1^2} \sum_{j=1}^{N} b_j\, e^{-(\mathbf{R}_m - \mathbf{r}_j)^2/2r_1^2} \simeq \frac{1}{2\pi r_1^2} \sum_{|\mathbf{R}_m - \mathbf{r}_j| < r_1^{\max}} b_j\, e^{-(\mathbf{R}_m - \mathbf{r}_j)^2/2r_1^2}\,, \qquad (2.16a)$$

$$S(\mathbf{r}_i) = \frac{A^2}{2\pi r_2^2} \sum_{\mathbf{R}_m} e^{-(\mathbf{R}_m - \mathbf{r}_i)^2/2r_1^2} S^{\text{mesh}}(\mathbf{R}_m) \simeq \frac{A^2}{2\pi r_2^2} \sum_{|\mathbf{R}_m - \mathbf{r}_i| < r_2^{\max}} e^{-(\mathbf{R}_m - \mathbf{r}_i)^2/2r_2^2} S^{\text{mesh}}(\mathbf{R}_m), \qquad (2.16b)$$

where $r_{1,2}^{\max} = r_{1,2}\sqrt{2\ln(\frac{1}{\epsilon})}$. Typically, $r_1^{\max} \gtrsim r_2^{\max} \sim 5A$, so each dislocation is spread over (and the force is reconstructed from) about 100 mesh points.

The sequence of operations is as follows.

(1) Choose the mesh size A and r_0, r_1 and r_2 (described below).

(2) Using Eq. (2.16a), find $\{\rho^{\text{mesh}}(\mathbf{R}_m)\}$.

(3) Fourier-transform $\{\rho^{\text{mesh}}(\mathbf{R}_m)\}$ to obtain $\{\tilde{\rho}^{\text{mesh}}(\mathbf{k}_m)\}$.

(4) Calculate $\{\tilde{f}^{\text{mesh}}(\mathbf{k}_m)\}$ using Eq. (2.13) (this has

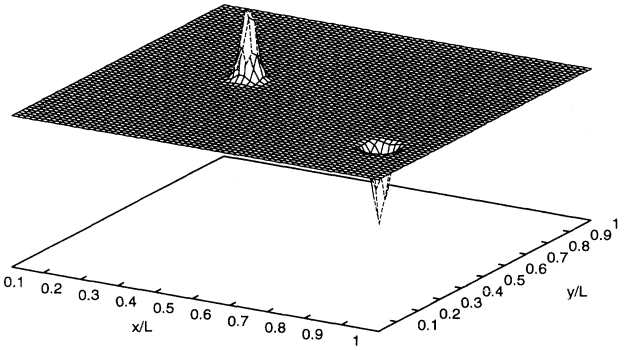

FIG. 2. $\rho^{\rm mesh}$ (in relative units) for a pair of edge dislocations on a 64×64 mesh. Parameters are chosen as $r_0 = 2.38A$, $r_1 = 1.08A$, $r_2 = 1.00A$, which corresponds to the overall accuracy of 10^{-5}. Each dislocation is spread over 113 mesh points.

to be done only once).

(5) For each value of $\mathbf{k}_m$, use Eq. (2.14) to get $\{\tilde{S}^{\rm mesh}(\mathbf{k}_m)\}$.

(6) Fourier-transform $\{\tilde{S}^{\rm mesh}(\mathbf{k}_m)\}$ to obtain $\{S^{\rm mesh}(\mathbf{R}_m)\}$.

(7) Recover $S(\mathbf{r}_i)$ from $\{S^{\rm mesh}(\mathbf{R}_m)\}$ using Eq. (2.16b); finally, $F^i_{lr} = b_i S(\mathbf{r}_i)$.

Figures 2 and 3 show an example of $\{\rho^{\rm mesh}(\mathbf{R}_m)\}$ and $\{S^{\rm mesh}(\mathbf{R}_m)\}$ created by a pair of edge dislocations. Note the long-range and anisotropic character of $S^{\rm mesh}$.

In the two following subsections we present the actual form of the right-hand side of Eq. (2.3) for both types of dislocations; the corresponding integrals are evaluated in Appendix A.

A. Screw dislocations

Throughout this paper, k and ϕ denote polar coordinates in Fourier space, corresponding to r, θ in real space. The force between two dislocations is [9,10]

$$f(\mathbf{r}) \equiv f^x(\mathbf{r}) = \mu\frac{\cos\theta}{2\pi r}\,, \tag{2.17a}$$

$$\tilde{f}(\mathbf{k}) = \mu\frac{-i\cos\phi}{k}\,. \tag{2.17b}$$

We rewrite the right-hand side of Eq. (2.17a) as

$$f(\mathbf{r}) = \mu\frac{\cos\theta}{2\pi r}\left[e^{-r^2/2r_0^2} + \left(1 - e^{-r^2/2r_0^2}\right)\right]\,. \tag{2.18}$$

Here, $e^{-r^2/2r_0^2}$ is rapidly decaying in r space, so we choose

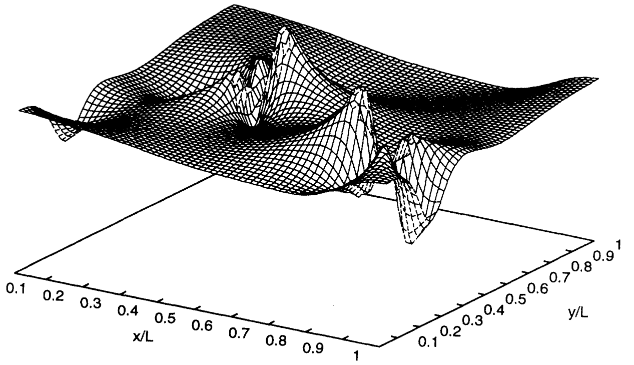

FIG. 3. $S^{\rm mesh}$ corresponding to $\rho^{\rm mesh}$ shown in Fig. 2.

$$f_{sr}^{\rm scr}(\mathbf{r}) = \mu\frac{\cos\theta}{2\pi r}\, e^{-r^2/2r_0^2}\,, \qquad r_{\rm max} \simeq r_0\sqrt{2\ln\frac{1}{\epsilon}} \tag{2.19a}$$

while the Fourier transform of $\mu\frac{\cos\theta}{2\pi r}(1 - e^{-r^2/2r_0^2})$ is

$$\tilde{f}_{lr}^{\rm scr}(\mathbf{k}) = \mu\frac{-i\cos\phi}{k}\, e^{-k^2r_0^2/2}\,. \tag{2.19b}$$

B. Edge dislocations

In this case [9,10],

$$f(\mathbf{r}) \equiv f^x(\mathbf{r}) = \frac{\mu}{1-\nu}\,\frac{\cos\theta\cos 2\theta}{2\pi r}, \tag{2.20a}$$

$$\tilde{f}(\mathbf{k}) = -\frac{\mu}{1-\nu}\,\frac{i\sin\phi\sin 2\phi}{k}\,. \tag{2.20b}$$

It turns out that using a splitting analogous to Eq. (2.18) would lead to an $\tilde{f}_{lr}^{\rm edge}(\mathbf{k})$ that has only power-law decay. However, introducing one more Gaussian does the trick:

$$f_{sr}^{\rm edge}(\mathbf{r}) = \frac{\mu}{1-\nu}\,\frac{\cos\theta\cos 2\theta}{2\pi r}\,\left(2e^{-r^2/2r_0^2} - e^{-r^2/r_0^2}\right),$$

$$r_{\rm max} \simeq r_0\sqrt{2\ln\frac{1}{\epsilon}},$$

$$f_{lr}^{\rm edge}(\mathbf{r}) = \frac{\mu}{1-\nu}\,\frac{\cos\theta\cos 2\theta}{2\pi r}\,\left(1 - e^{-r^2/2r_0^2}\right)^2\,, \tag{2.21a}$$

so that the factor $\left(1 - e^{-r^2/2r_0^2}\right)^2 \propto r^4$ $(r \to 0)$ cancels the singularity of $f(\mathbf{r})$ at the origin. Using Appendix A, we have

$$\tilde{f}_{lr}^{\rm edge}(\mathbf{k}) = \frac{\mu}{1-\nu}\,\frac{i}{k}\, e^{-k^2r_0^2/4}\left[\cos\phi\cos 2\phi\left(4\frac{1 - e^{-k^2r_0^2/4}}{k^2r_0^2} + 1 - 2e^{-k^2r_0^2/4}\right) - 4\,\frac{\sin\phi\sin 2\phi}{k^2r_0^2}\left(1 - e^{-k^2r_0^2/4}\right)\right]\,. \tag{2.21b}$$

Note that the k-space decay is slower than in the case of screw dislocations but is nevertheless Gaussian.

III. NUMERICAL IMPLEMENTATION AND ERROR ESTIMATES

We use standard fast Fourier transform (FFT) routines [8] which require that the number of mesh points M in each direction be a power of 2, therefore, in our choice of A, r_0, r_1, and r_2 we start with $A = L/M$; r_0, r_1, and r_2 are then given by Eqs. (B7).

We now estimate the operation count Ω. In the short-range part, each dislocation interacts with an average of $\frac{1}{2}\bar{\rho}\pi r_{\max}^2$ others, where $\bar{\rho} = N/L^2$ and $r_{\max} = K_T(\epsilon)A = K_T(\epsilon)L/M$, where $K_T(\epsilon)$ depends on the precision ϵ and dislocation type T [$K_T(\epsilon)$ is the product of $r_{\max}/r_0$ given by Eq. (2.21a) or Eq. (2.19a) and r_0/A defined by Eqs. (B7); note that $K_T(\epsilon) \propto \ln(1/\epsilon)$]. Thus the operation count for computing the short-range interactions is

$$\Omega_{sr} \simeq \frac{N^2 K_T^2}{16\pi M^2} \equiv \frac{\gamma_0 N^2}{M^2} , \tag{3.1}$$

where γ_0 is independent of N and M but depends on ϵ.

The long-range part of the force calculation involves two FFT's using $O(M^2 \log_2 M) \equiv \gamma_1 M^2 \ln M^2$ operations, where γ_2 is a constant independent of all the parameters of the calculation. Performing steps (2) and (7) from Sec. II involves a number of operations proportional to N; also, assigning dislocations to cells is an $O(N)$ procedure. We denote all the $O(N)$ parts by $\gamma_2 N$. The total number of operations is therefore

$$\Omega_N(\eta) = \gamma_0 N^2/\eta + \gamma_1 \eta \ln \eta + N\gamma_2 , \tag{3.2}$$

where $\eta \equiv M^2$. Minimizing this expression with respect to η gives $\eta = N\sqrt{\frac{\gamma_0}{\gamma_1(1+\ln\eta)}}$, which can be solved iteratively,

$$\eta_0 = N\sqrt{\gamma_0/\gamma_1}, \qquad \eta_1 = N\sqrt{\frac{\gamma_0}{\gamma_1(1+\ln N)}} , \qquad \ldots,$$

so for simplicity we take $\eta' = N\sqrt{\frac{\gamma_0}{\gamma_1 \ln N}}$; substituting this into Eq. (3.2) gives

$$\Omega_N(\eta') = \sqrt{\gamma_0\gamma_1}\; N\Big[2\sqrt{\ln N} - \frac{\ln\left(\frac{\gamma_1}{\gamma_0}\ln N\right)}{2\sqrt{\ln N}}\Big] + \gamma_2 N .$$

Since $\Omega_N(\eta^{opt}) \leq \Omega_N(\eta')$ by definition, and for large N, $\frac{\ln(\frac{\gamma_1}{\gamma_0}\ln N)}{2\sqrt{\ln N}} > 0$, we may write

$$\Omega_N(\eta^{\rm opt}) \;\leq 2\sqrt{\gamma_0\gamma_1}\; N\sqrt{\ln N} + \gamma_2 N . \tag{3.3}$$

Therefore, our CPU time scales not worse than $O(N\sqrt{\ln N})$.

Figure 4 shows our empirical results for CPU time as a function of N, for various values of ϵ. As seen in the figure, the dependence on N is nearly linear, since the function $\sqrt{\ln N}$ is very slowly varying. The computational speed of the present method appears to be higher than that of the multipole method of Wang and LeSar [5]. For the purposes of comparing to other methods, we note that the computational efficiency may be given in terms of the speedup relative to the direct calculation using the same computer and compiler options. *In the present method, for an accuracy of* 10^{-3} ($\epsilon = 10^{-4}$), *the speedup is roughly equal to the number of dislocations divided by 1000.*

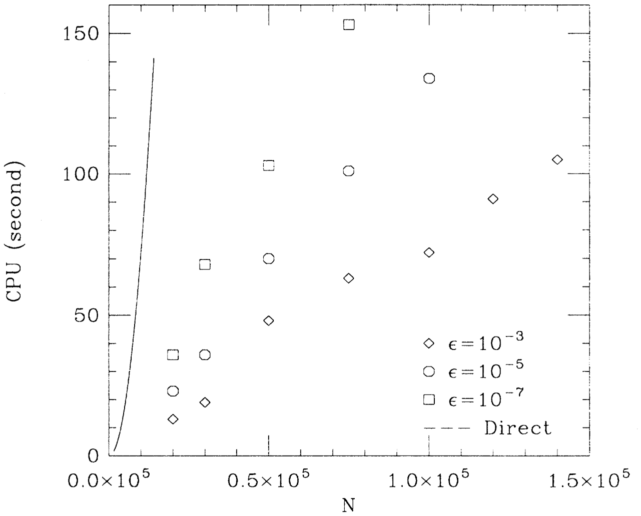

FIG. 4. CPU time the SGI R 4000 uses to calculate the set of forces for N edge dislocations with ϵ of 10^{-3}, 10^{-5}, and 10^{-7}. For comparison, the line shows the CPU time for direct (nonperiodic) calculation using the same compiler options.

It is also useful to establish how the CPU time depends on ϵ. As mentioned above, $\gamma_0 \propto \ln^2(1/\epsilon)$ and γ_1 is a characteristic of the FFT code and thus does not depend on ϵ. The numbers of summands in Eq. (2.16) are proportional to r_1^2 and r_2^2 respectively, while both r_1^2 and r_2^2 themselves are proportional to $\ln(1/\epsilon)$. Therefore, γ_2 is quadratic in $\ln(1/\epsilon)$. We can finally rewrite Eq. (3.3) as

$$\Omega(N,\epsilon) \simeq N\ln\left(\frac{1}{\epsilon}\right)\;\left[\alpha\sqrt{\ln N} + \beta\ln\left(\frac{1}{\epsilon}\right)\right] . \tag{3.4}$$

This indicates a very favorable scaling of CPU with respect to the accuracy. Indeed, Fig. 4 shows that increasing the accuracy by 4 orders of magnitude results in only moderate increase in CPU time.

In order to evaluate the accuracy of ϵ in predicting the errors in the forces, we compare the values of the forces with those obtained by direct summation (still in the supercell geometry):

$$F^i_{\rm exact} = b_i \lim_{\kappa\to 0}\left[\sum_j{}' b_j \sum_{p,q=-\infty}^{\infty} f(x_{ji}+pL, y_{ji}+qL) \times e^{-\kappa\left[(x_{ji}+pL)^2+(y_{ji}+qL)^2\right]}\right] , \tag{3.5}$$

where p and q are integers. Table I shows that our precision measure ϵ approximates the actual error within a factor of 2.

TABLE I. Comparison of the forces obtained directly using Eq. (3.5) with those obtained using our P^{3}M algorithm. Random test configurations were taken except those marked by "†," where we chose a configuration with vanishing moments up through the octupole in order to speed up the convergence of the sum in Eq. (3.5). The numerical values are normalized so that the absolute values of the forces are of order 1.

N	Type	M	ϵ	rms $(F - F_{\text{exact}})$
18	edge	128	10^{-3}	1.67×10^{-3}
18	edge	128	10^{-4}	1.21×10^{-4}
16†	edge	128	10^{-7}	1.52×10^{-7}
16†	edge	128	10^{-9}	1.38×10^{-9}

IV. VECTOR FORCES

We now describe the generalization of the preceding results in order to obtain all the components of the force for non-parallel Burgers vectors.

In the case of screw dislocations, in addition to Eq. (2.17a) we have to consider the y component of the force given by

$$f^y = \mu \frac{\sin\theta}{2\pi r} \ ,$$

which can be tackled analogously to f_x, as indicated in Table II.

The force between two edge dislocations with Burgers vectors $\mathbf{b}_1$ and $\mathbf{b}_2$ is [10]

$$\mathcal{F}^x = \frac{\mu}{2\pi(1-\nu)} \left[b_1^x \left(b_2^x \frac{\cos\theta\cos 2\theta}{r} + b_2^y \frac{\sin\theta\cos 2\theta}{r} \right) \right.$$
$$+ b_1^y \left(b_2^x \frac{\sin\theta\cos 2\theta}{r} \right.$$
$$\left.\left. + b_2^y \frac{\cos\theta(2-\cos 2\theta)}{r} \right) \right] , \tag{4.1a}$$

$$\mathcal{F}^y = + \frac{\mu}{2\pi(1-\nu)} \left[b_1^x \left(b_2^x \frac{\sin\theta(2+\cos 2\theta)}{r} \right. \right.$$
$$\left. - b_2^y \frac{\cos\theta\cos 2\theta}{r} \right)$$
$$\left. - b_1^y \left(b_2^x \frac{\cos\theta\cos 2\theta}{r} + b_2^y \frac{\sin\theta\cos 2\theta}{r} \right) \right] . \tag{4.1b}$$

We have already described the treatment of the first term in Eq. (4.1a). In addition, there are terms with three different θ dependencies: $\cos\theta/r$, $\sin\theta/r$, and $\sin\theta\cos 2\theta/r$. Table II summarizes the implementation of Eq. (2.3) for all the four functions.

The density of Burgers vectors is now a vector and $S(\mathbf{r})$ is a tensor:

$$\vec{F}^i_{lr} = \overset{\leftrightarrow}{S}(\mathbf{r}_i) \ \vec{b}_i \ .$$

From the Peach-Koehler formula [9,10] it follows that $\overset{\leftrightarrow}{S}$ is related to the stress tensor as follows:

$$\begin{pmatrix} S_{xx} & S_{xy} \\ S_{yx} & S_{yy} \end{pmatrix} = \begin{pmatrix} \sigma_{yx} & \sigma_{yy} \\ -\sigma_{xx} & -\sigma_{xy} \end{pmatrix} .$$

Equation (2.6) becomes

TABLE II. Splitting into the long-range and short-range parts of the edge dislocation forces for the terms in Eqs. (4.1).

$f(\mathbf{r})$	$f_{sr}(\mathbf{r})$	$\tilde{f}_{lr}(\mathbf{k})$
$\frac{\cos\theta}{r}$	$\frac{\cos\theta}{r} e^{-r^2/2r_0^2}$	$-2\pi \frac{i\cos\phi}{k} e^{-\frac{1}{2}k^2 r_0^2}$
$\frac{\sin\theta}{r}$	$\frac{\sin\theta}{r} e^{-r^2/2r_0^2}$	$2\pi \frac{i\sin\phi}{k} e^{-\frac{1}{2}k^2 r_0^2}$
$\frac{\cos\theta\cos 2\theta}{r}$	$\frac{\cos\theta\cos 2\theta}{r} e^{-r^2/2r_0^2} \left(2 - e^{-r^2/2r_0^2}\right)$	$\frac{2\pi i}{k} e^{-k^2 r_0^2/4} \left[-4 \frac{\sin\phi\sin 2\phi}{k^2 r_0^2} \left(1 - e^{-k^2 r_0^2/4}\right) + \cos\phi\cos 2\phi \left(4 \frac{1-e^{-k^2 r_0^2/4}}{k^2 r_0^2} + 1 - 2e^{-k^2 r_0^2/4} \right) \right]$
$\frac{\sin\theta\cos 2\theta}{r}$	$\frac{\sin\theta\cos 2\theta}{r} e^{-r^2/2r_0^2} \left(2 - e^{-r^2/2r_0^2}\right)$	$\frac{2\pi i}{k} e^{-k^2 r_0^2/4} \left[4 \frac{\cos\phi\sin 2\phi}{k^2 r_0^2} \left(1 - e^{-k^2 r_0^2/4}\right) + \sin\phi\cos 2\phi \left(4 \frac{1-e^{-k^2 r_0^2/4}}{k^2 r_0^2} + 1 - 2e^{-k^2 r_0^2/4} \right) \right]$

$$S_{\alpha\beta}(\mathbf{r}_i) = \int f_{lr}^{\alpha\beta\gamma}(\mathbf{r}_i - \mathbf{r})\rho_\gamma(\mathbf{r})\, d(\mathbf{r}) \ ,$$

where the components of the $2 \times 2 \times 2$ matrix $f_{lr}^{\alpha\beta\gamma}$ are easily constructed from the eight summands in Eqs. (4.1) using the table. Note that from Eq. (4.2) it is clear that $S_{xx}(\mathbf{r}) = -S_{yy}(\mathbf{r})$ [which is consistent with Eqs. (4.1)], therefore only three components of $\overleftrightarrow{S}$ need to be calculated.

It turns out that the evaluation of the forces in this case takes only about 2.5 times longer than in the simple case described in Secs. II and III because much of the calculation is shared by different terms, e.g., the exponentials in Eqs. (2.16). Overall accuracy in this case is several times lower than in Table I and is about 10ϵ.

V. CONCLUSION

The main conclusion of the above is that it is possible to obtain a useful $O(N)$ particle-particle–particle-mesh method for dislocation interactions. This method has the virtues of formal simplicity, computational efficiency, and a very attractive scaling of CPU time with increasing accuracy. It can also be easily parallelized. Our method is comparable in speed to the recently published multipole method of Wang and LeSar [5,11], and we believe that the simplicity of the present method may be very advantageous in developing a methodology for a fully three-dimensional calculation.

A possible extension to three dimensions is as follows: start with the expression for the interaction energy of two dislocation loops C_1 and C_2 [10]:

$$\begin{aligned} W_{12} = &-\frac{\mu}{2\pi}\oint_{C_1}\oint_{C_2}\frac{[\mathbf{b}_1 \times \mathbf{b}_2]\cdot[d\mathbf{l}_1 \times d\mathbf{l}_2]}{r} \\ &+\frac{\mu}{2\pi}\oint_{C_1}\oint_{C_2}\frac{(\mathbf{b}_1 \cdot d\mathbf{l}_1)(\mathbf{b}_2 \cdot d\mathbf{l}_2)}{r} \\ &+\frac{\mu}{4\pi(1-\nu)}\oint_{C_1}\oint_{C_2}(\mathbf{b}_1 \times d\mathbf{l}_1)\cdot\mathbf{T}\cdot(\mathbf{b}_2 \times d\mathbf{l}_2) \ , \end{aligned}$$

where $T_{\alpha\beta} = \partial^2 r/\partial x_\alpha \partial x_\beta$. The force that loop 2 exerts on $d\mathbf{l}_1$ is

$$\begin{aligned} \mathfrak{F} = &+\frac{\mu}{2\pi}\oint_{C_2}\nabla\frac{1}{r}([\mathbf{b}_1 \times \mathbf{b}_2]\cdot[d\mathbf{l}_1 \times d\mathbf{l}_2]) \\ &-\frac{\mu}{2\pi}\oint_{C_2}\nabla\frac{1}{r}(\mathbf{b}_1 \cdot d\mathbf{l}_1)(\mathbf{b}_2 \cdot d\mathbf{l}_2) \\ &-\frac{\mu}{4\pi(1-\nu)}\oint_{C_2}(\mathbf{b}_1 \times d\mathbf{l}_1)\cdot\nabla\mathbf{T}\cdot(\mathbf{b}_2 \times d\mathbf{l}_2) \ . \end{aligned}$$

Since multiplication of the r-space functions by even powers of r fails to regularize them around the origin, it is necessary to do the splitting in k space. The Fourier transform of $\nabla\frac{1}{r}$ is $-4\pi i\mathbf{k}/k^2$ which suggests a splitting of the form $\tilde{f}_{lr}(\mathbf{k}) = -4\pi i\mathbf{k}\exp(-\frac{1}{2}k^2r_0^2)/k^2$ and $\tilde{f}_{sr}(\mathbf{k}) = -4\pi i\mathbf{k}\frac{1-\exp(-\frac{1}{2}k^2r_0^2)}{k^2}$ for the first two terms. In the third term, the formal Fourier transform of r is $8\pi/k^4$, so $\partial^3 r/\partial x_\alpha \partial x_\beta \partial x_\gamma$ transforms to $-8\pi i k_\alpha k_\beta k_\gamma/k^4$. Therefore a splitting such as

$$\tilde{f}_{sr}^{\alpha\beta\gamma}(\mathbf{k}) = -8\pi i k_\alpha k_\beta k_\gamma \ \frac{[1-\exp(-\frac{1}{2}k^2r_0^2)]^2}{k^4} \ ,$$

$$\tilde{f}_{lr}^{\alpha\beta\gamma}(\mathbf{k}) = -8\pi i k_\alpha k_\beta k_\gamma \ \frac{2\exp(-\frac{1}{2}k^2r_0^2) - \exp(-k^2r_0^2)}{k^4} \ ,$$

will work.

Otherwise, the procedure should be completely analogous to that described here, once one has resolved problems such as, for example, how to parametrize and store the dislocation lines, and treat dislocation crossings.

ACKNOWLEDGMENTS

We appreciate stimulating and useful discussions with Robb Thomson and Lilly Canel and a careful reading by Jun Zou. We thank James Rice for allowing D.B. the opportunity to visit Harvard University, where part of this work was performed. This work was supported by the Department of Energy under Grant No. DE-FG02-84ER45130.

APPENDIX A: CALCULATION OF FOURIER TRANSFORMS OF FORCES

We use two general formulas [12]. First,

$$\oint \exp(i\xi\cos\theta)d\theta = 2\pi J_0(\xi)$$

(where $\oint d\theta \equiv \int_0^{2\pi} d\theta$), from which it follows that

$$\oint \cos^n(\theta)\exp(i\xi\cos\theta)d\theta = 2\pi\frac{d^n}{i^n d\xi^n}J_0(\xi) \ ,$$

which we will need for $n = 1$ and 3; using well-known relations for the Bessel functions, we have

$$\oint \cos(\theta)\exp(i\xi\cos\theta)d\theta = 2\pi i J_1(\xi) \ , \ \oint \cos^3(\theta)\exp(i\xi\cos\theta)d\theta = \pi i\ \frac{3J_1(\xi) - J_3(\xi)}{2} \ .$$

Second,

$$\int_0^\infty e^{-\alpha x^2}J_n(\beta x)dx = \frac{\sqrt{\pi}}{2\sqrt{\alpha}}\,e^{-\beta^2/8\alpha}I_{\frac{n}{2}}\left(\frac{\beta^2}{8\alpha}\right) \ ,$$

which we also need for $n = 1$ and 3:

$$\int_0^\infty e^{-\alpha x^2} J_1(\beta x)dx = \frac{1}{\beta}(1-e^{-\beta^2/4\alpha}), \tag{A1}$$

$$\int_0^\infty e^{-\alpha x^2} J_3(\beta x)dx = \frac{1}{\beta}\left[1-\frac{8\alpha}{\beta^2}+\left(1+\frac{8\alpha}{\beta^2}\right)e^{-\beta^2/4\alpha}\right]. \tag{A2}$$

We begin by deriving Eq. (2.19b) (we omit μ and ν-dependent constant factors for clarity). We have

$$\begin{aligned}\tilde{f}(k,\phi) &= \int_0^\infty r dr \oint d\theta \, \frac{\cos\theta}{r} \, e^{-ikr\cos(\theta-\phi)} \\ &= \int_0^\infty dr \oint d\theta \, \cos(\theta+\phi) \, e^{-ikr\cos\theta} \\ &= -\frac{2\pi\cos\phi}{i}\int_0^\infty dr\left[\frac{d}{d(kr)}\right]J_0(kr) \\ &= -2\pi i\cos\phi/k \, ,\end{aligned}$$

where we have used angle-addition formulas. Note that the $\sin\phi$ term vanishes because the function $\sin\theta \; e^{-ikr\cos\theta}$ is odd.

It is now easy to see that

$$\begin{aligned}\tilde{f}_{lr}(k,\phi) &= \int_0^\infty r dr \oint d\theta \, \frac{\cos\theta}{r}(1-e^{-r^2/2r_0^2}) \, e^{-ikr\cos(\theta-\phi)} \\ &= -2\pi i\cos\phi\left(\frac{1}{k}+\int_0^\infty dr \; e^{-r^2/2r_0^2}\left[\frac{d}{d(kr)}\right]J_0(kr)\right) \\ &= -2\pi i\cos\phi\left(\frac{1}{k}-\int_0^\infty dr \; J_1(kr)e^{-r^2/2r_0^2}\right) \\ &= -\frac{2\pi i\cos\phi}{k} \; e^{-k^2r_0^2/2} \, .\end{aligned}$$

Now derive similar expressions for edge dislocations:

$$\begin{aligned}\tilde{f}(k,\phi) &= \int_0^\infty r dr \oint d\theta \, \frac{\cos\theta\cos(2\theta)}{r} \; e^{ikr\cos(\theta-\phi)} \\ &= \int_0^\infty dr \oint d\theta \, \cos(\theta+\phi)\cos(2\theta+2\phi) \; e^{-ikr\cos\theta} \\ &= \cos\phi\cos 2\phi\int_0^\infty dr \oint d\theta \, \cos\theta\cos 2\theta e^{-ikr\cos\theta} + \sin\phi\sin 2\phi\int_0^\infty dr \oint d\theta \, \sin\theta\sin 2\theta e^{-ikr\cos\theta} \\ &= \cos\phi\cos 2\phi\int_0^\infty dr \oint d\theta(2\cos^3\theta-\cos\theta)e^{-ikr\cos\theta} + \sin\phi\sin 2\phi\int_0^\infty dr \oint d\theta(2\cos\theta-2\cos^3\theta)e^{-ikr\cos\theta} \\ &= -\pi i\left[\cos\phi\cos 2\phi\int_0^\infty dr\Big[J_1(kr)-J_3(kr)\Big]+\sin\phi\sin 2\phi\int_0^\infty dr\Big[J_1(kr)+J_3(kr)\Big]\right] \\ &= -\frac{2\pi i}{k}\sin\phi\sin 2\phi \, .\end{aligned}$$

We use this derivation to write

$$\begin{aligned}\tilde{f}_{lr}^{\rm edge}(k,\phi) &= \int_0^\infty r dr \oint d\theta\frac{\cos\theta\cos 2\theta}{r}(1-2e^{-r^2/2r_0^2}+e^{-r^2/r_0^2}) \; e^{-ikr\cos(\theta-\phi)} \\ &= -\pi i\left[\frac{2\sin\phi\sin 2\phi}{k}+\cos\phi\cos 2\phi\int_0^\infty dr[J_1(kr)-J_3(kr)](-2e^{-r^2/2r_0^2}+e^{-r^2/r_0^2})\right. \\ &\qquad \left. +\sin\phi\sin 2\phi\int_0^\infty dr[J_1(kr)+J_3(kr)](-2e^{-r^2/2r_0^2}+e^{-r^2/r_0^2})\right] \, ,\end{aligned}$$

which we can now expand and use Eq. (A2) to calculate each of the four resulting integrals. The result is

$$\begin{aligned}\tilde{f}_{lr}^{\rm edge}(\mathbf{k}) = -\frac{2\pi i}{k}&\left[\cos\phi\cos 2\phi\left(4\frac{e^{-k^2r_0^2/2}-e^{-k^2r_0^2/4}}{k^2r_0^2}+2e^{-k^2r_0^2/2}-e^{-k^2r_0^2/4}\right)\right. \\ &\left. +4\frac{\sin\phi\sin 2\phi}{k^2r_0^2}\left(e^{-k^2r_0^2/4}-e^{-k^2r_0^2/2}\right)\right] \, .\end{aligned}$$

APPENDIX B: PRECISION ESTIMATE AND CONDITIONS FOR $\mathbf{r}_0$, $\mathbf{r}_1$, AND $\mathbf{r}_2$

The long-range force calculation is approximate. The errors are caused at the following steps.

(1) Truncation of the second sum in Eq. (2.16a) for $\rho^{\text{mesh}}\{\mathbf{R}\}$.

(2) Sampling of $\rho^{\text{mesh}}(\mathbf{r})$ to perform the discrete Fourier transform to obtain $\rho^{\text{mesh}}(\mathbf{k})$.

(3) Inverse discrete Fourier transform to obtain $\overline{(f^{\text{mesh}} \circ \rho^{\text{mesh}})}(\mathbf{R})$ from $f^{\text{mesh}}(\mathbf{k})\tilde{\rho}^{\text{mesh}}(\mathbf{k})$.

(4) Approximation of the integral in Eq. (2.10b) with the sum in Eq. (2.16b).

We evaluate the errors at each step separately, with overbarred symbols denoting the approximate values and require that those errors be of order ϵ so that the overall error will be of the same order.

In this section, we repeatedly use the formula

$$A^2 \sum_{\{\mathbf{R}_m\}} \delta(\mathbf{r} - \mathbf{R}_m) = \sum_{\{\mathbf{K}_l\}} e^{-i\mathbf{K}_l \cdot \mathbf{r}}, \tag{B1}$$

where $\{\mathbf{R}_m\}$ means the real-space square mesh with size A while the reciprocal $\{\mathbf{K}_l\}$ is a square mesh with spacing $2\pi/A$. (Note that the k mesh we use for the Fourier transforms has the size $2\pi/L$.)

1. Calculation of the mesh-point density

For each mesh point $\mathbf{R}_m$, we ignore the contributions to $\rho^{\text{mesh}}(\mathbf{R}_m)$ in the second sum of Eq. (2.16a) from the dislocations that are $r_1^{\max}$ and further away, causing an error of order ϵ at each mesh point.

2. Errors of Fourier transform in obtaining $\tilde{\rho}^{\text{mesh}}$

For the Fourier transform of ρ^{mesh}, we require that

$$\left|\tilde{f}^{\text{mesh}}(\mathbf{k}_m)[\tilde{\rho}^{\text{mesh}}(\mathbf{k} = \mathbf{k}_m) - \overline{\tilde{\rho}^{\text{mesh}}}(\mathbf{k}_m)]\right| \lesssim \epsilon, \qquad \forall m \, . \tag{B2}$$

$[\tilde{f}^{\text{mesh}}(\mathbf{k}_m)$ is exact up to the computer precision.] We have

$$\begin{aligned}\overline{\tilde{\rho}^{\text{mesh}}}(\mathbf{k}_m) &= A^2 \int d\mathbf{r}\rho^{\text{mesh}}(\mathbf{r})e^{-i\mathbf{k}_m \cdot \mathbf{r}} \sum_{\{\mathbf{R}_m\}} \delta(\mathbf{r} - \mathbf{R}_m) \\ &= \sum_{\{\mathbf{K}_l\}} \int d\mathbf{r}\rho^{\text{mesh}}(\mathbf{r})e^{-i\mathbf{k}_m \cdot \mathbf{r}}e^{-i\mathbf{K}_l \cdot \mathbf{r}} \\ &= \sum_{\{\mathbf{K}_l\}} \tilde{\rho}^{\text{mesh}}(\mathbf{k}_m + \mathbf{K}_l) \\ &= \tilde{\rho}^{\text{mesh}}(\mathbf{k}_m) + \sum_{\{\mathbf{K}_l \neq 0\}} \tilde{\rho}^{\text{mesh}}(\mathbf{k}_m + \mathbf{K}_l) \, ,\end{aligned}$$

so Eq. (B2) means that

$$\left|\sum_{\{\mathbf{K}_l \neq 0\}} \tilde{f}^{\text{mesh}}(\mathbf{k}_m)\tilde{\rho}^{\text{mesh}}(\mathbf{k}_m + \mathbf{K}_l)\right| \lesssim \epsilon \, , \qquad \forall m \, . \tag{B3}$$

Notice that the values of $\mathbf{k}_m$ are within π/A of the origin in both x and y directions [cf. Eq. (2.15)] while the nonzero values of $\mathbf{K}_l$ begin at $2\pi/A$. Let us estimate the largest term in this sum.

(i) *Edge*: The long-range decay of $\tilde{f}^{\text{mesh}}$ is determined by $e^{-\frac{k^2}{2}(\frac{1}{2}r_0^2 - r_1^2 - r_2^2)}$, and that of $\tilde{\rho}^{\text{mesh}}$ by $e^{-\frac{1}{2}k^2 r_1^2}$, so the term in the sum in Eq. (B3) is proportional to

$$\exp\left[-k_m^2\left(\frac{r_0^2}{4} - \frac{r^2}{2}\right) - \mathbf{k}_m \cdot \mathbf{K}_l r_1^2 - \frac{K_l^2 r_1^2}{2}\right] .$$

Its maximum value with respect to $\mathbf{k}_m$ is

$$\exp\left(-K_l^2 r_1^2 \frac{r_0^2 - r_1^2 - r_2^2}{2r_0^2 - 4r_1^2}\right) ,$$

whose maximum is reached at the minimum (nonzero) value of K_l^2, i.e., $(2\pi/A)^2$. We must therefore demand that

$$\exp\left[-\frac{2\pi^2 r_1^2(r_0^2 - 2r_1^2 - 2r_2^2)}{A^2(r_0^2 - 2r_2^2)}\right] < \epsilon \quad \text{(edge)} \, . \tag{B4a}$$

(ii) *Screw*: The difference is that $\tilde{f}^{\text{mesh}}$ decays like $e^{-\frac{k^2}{2}(r_0^2 - r_1^2 - r_2^2)}$. Changing $\frac{1}{2}r_0^2 \to r_0^2$ in Eq. (B4a) we obtain

$$\exp\left[-\frac{2\pi^2 r_1^2(r_0^2 - r_1^2 - r_2^2)}{A^2(r_0^2 - r_2^2)}\right] < \epsilon \quad \text{(screw)} \, . \tag{B4b}$$

3. Inverse Fourier transform to obtain $(f^{\text{mesh}} \circ \rho^{\text{mesh}})(\mathbf{R})$

The mesh Fourier transform $\overline{\tilde{f}^{\text{mesh}}(\mathbf{k}_m)\tilde{\rho}^{\text{mesh}}(\mathbf{k}_m)}$ differs from the continuous $f^{\text{mesh}}(\mathbf{k})\rho^{\text{mesh}}(\mathbf{k})$ in two ways. First, the latter integrates over the entire k plane while the values of $\mathbf{k}_m$ are confined within a finite box (of size $k_{\max}$). Also, the $\mathbf{k}_m$'s are discrete while $\mathbf{k}$ is continuous: this leads to the periodicity of the force with the period L.

We address the first of these by requiring that $f^{\text{mesh}}(\mathbf{k})\rho^{\text{mesh}}(\mathbf{k})|_{k \geq k_{\max}} \lesssim \epsilon$. Using Eq. (2.12), we obtain

$$\exp\left[-\frac{\pi^2}{2A^2}(\tfrac{1}{2}r_0^2 - r_2^2)\right] < \epsilon \quad \text{(edge)} \, , \tag{B5a}$$

$$\exp\left[-\frac{\pi^2}{2A^2}(r_0^2 - r_2^2)\right] < \epsilon \quad \text{(screw)} \, . \tag{B5b}$$

4. Obtaining F at dislocation positions from $S^{\text{mesh}}(\mathbf{R})$

We need to approximate the integral in Eq. (2.10b) with the second sum in Eq. (2.16b). There are two errors: that of cutting the integration off at $r' = r_2^{\max}$ and that of doing the sum discretely over the mesh points:

$$\bar{F}(\mathbf{r}_i) = \frac{A^2}{2\pi r_2^2} \int_{r'<r_2^{\max}} d\mathbf{r}' S^{\text{mesh}}(\mathbf{r}') e^{-(\mathbf{r}_i-\mathbf{r}')^2/2r_2^2} \times \sum_{\{\mathbf{R}_m\}} \delta(\mathbf{r}' - \mathbf{R}_m) \; .$$

The first error is of order ϵ because of our choice of $r_2^{\max}$. In order to evaluate the second, we use Eq. (B1) to write

$$\bar{F}(\mathbf{r}_i) = \sum_{\{\mathbf{K}_l\}} \int d\mathbf{r}' \; S^{\text{mesh}}(\mathbf{r}') e^{-(\mathbf{r}_i-\mathbf{r}')^2/2r_2^2} e^{i\mathbf{K}_l \cdot \mathbf{r}'} \; .$$

Therefore,

$$\begin{aligned}
\bar{F}(\mathbf{r}_i) - F(\mathbf{r}_i) &= \frac{b_i}{2\pi r_2^2} \sum_{\{\mathbf{K}_l \neq 0\}} \int d\mathbf{r}' S^{\text{mesh}}(\mathbf{r}') e^{-(\mathbf{r}_i-\mathbf{r}')^2/2r_2^2} e^{-i\mathbf{K}_l \cdot \mathbf{r}'} \\
&= \frac{b_i}{2\pi r_2^2} \sum_{\{\mathbf{K}_l \neq 0\}} \int d\mathbf{r}' e^{-(\mathbf{r}_i-\mathbf{r}')^2/2r_2^2} e^{-i\mathbf{K}_l \cdot \mathbf{r}'} \int \frac{d\mathbf{k}}{(2\pi)^2} \tilde{S}^{\text{mesh}}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}'} \\
&= \frac{b_i}{2\pi r_2^2} \sum_{\{\mathbf{K}_l \neq 0\}} \int \frac{d\mathbf{k}}{(2\pi)^2} \tilde{f}^{\text{mesh}}(\mathbf{k}) \tilde{\rho}^{\text{mesh}}(\mathbf{k}) \int d\mathbf{r}' e^{-(\mathbf{r}_i-\mathbf{r}')^2/2r_2^2} e^{i\mathbf{r}'\cdot(\mathbf{k}-\mathbf{K}_l)} \\
&= \frac{b_i}{2\pi r_2^2} \sum_{\{\mathbf{K}_l \neq 0\}} e^{-i\mathbf{K}_l\cdot\mathbf{r}_i} \int \frac{d\mathbf{k}}{(2\pi)^2} \tilde{f}^{\text{mesh}}(\mathbf{k}) \tilde{\rho}^{\text{mesh}}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}_i} \int d\mathbf{r} e^{-\mathbf{r}^2/2r_2^2} e^{i\mathbf{r}\cdot(\mathbf{K}_l-\mathbf{k})} \\
&= b_i \sum_{\{\mathbf{K}_l \neq 0\}} e^{-i\mathbf{K}_l\cdot\mathbf{r}_i} \int \frac{d\mathbf{k}}{(2\pi)^2} \tilde{f}^{\text{mesh}}(\mathbf{k}) \tilde{\rho}^{\text{mesh}}(\mathbf{k}) e^{-\frac{1}{2}(\mathbf{k}-\mathbf{K}_l)^2 r_2^2} e^{i\mathbf{k}\cdot\mathbf{r}_i} \; .
\end{aligned}$$

As in the derivation of Eqs. (B4), we require that the integrand be bounded by ϵ and find its maximum value using asymptotic expressions for $\tilde{\rho}^{\text{mesh}}$ and $\tilde{f}^{\text{mesh}}$. The expression to maximize is now $\exp\left[-\frac{k^2 r_0^2}{4} + \mathbf{k}\cdot\mathbf{K}_l\, r_2^2 - \frac{1}{2}K_l^2 r_2^2\right]$ for edge and $\exp\left[-\frac{1}{2}k^2 r_0^2 + \mathbf{k}\cdot\mathbf{K}_l\, r_2^2 - \frac{1}{2}K_l^2 r_2^2\right]$ for screw dislocations. The conditions become

$$\exp\left[-\frac{2\pi^2 r_2^2(r_0^2 + 2r_2^2)}{A^2 r_0^2}\right] < \epsilon \quad (\text{edge}) \; , \qquad \text{(B6a)}$$

$$\exp\left[-\frac{2\pi^2 r_2^2(r_0^2 + r_2^2)}{A^2 r_0^2}\right] < \epsilon \quad (\text{screw}) \; . \qquad \text{(B6b)}$$

5. Resulting relations for r_0, r_1, and r_2

The parameters r_0, r_1, and r_2 have to meet three constraints: Eqs. (B4), (B5), and (B6). For optimization, we turn them all into equalities and find that

$$r_0^2 = \frac{32}{3}l^2 \, , \; r_1^2 = 2l^2 \, , \; r_2^2 = \frac{4}{3}l^2 \quad (\text{edge}) \; , \qquad \text{(B7a)}$$

$$r_0^2 = \frac{16}{3}l^2 \, , \; r_1^2 = 2l^2 \, , \; r_2^2 = \frac{4}{3}l^2 \quad (\text{screw}) \; . \qquad \text{(B7b)}$$

where $l^2 = \frac{A^2}{2\pi^2} \ln \frac{1}{\epsilon}$.

[1] L. Kubin, in *Treatise on Materials Science and Technology* (VCH, Weinberg, 1991), Vol. 6, Chap. 4.

[2] A. N. Gulluoglu, D. J. Srolovitz, R. LeSar, and P. S. Lomdahl, Scr. Met. **23**, 1347 (1989).

[3] D. L. Holt, J. Appl. Phys. **41**, 1397 (1970); see also discussion in Ref. [1].

[4] J. Shimada, H. Kaneko, and T. Takada, J. Comput. Chem. **15**, 28 (1994).

[5] H. Y. Wang and R. LeSar, Philos. Mag. A **71**, 149 (1995).

[6] R. W. Hockney and J. W. Eastwood, *Computer Simulations Using Particles* (Institute of Physics, Bristol, 1988).

[7] See, for example, Ref. [6], Sec. 6-5-4.

[8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C* (Cambridge University Press, Cambridge, 1991).

[9] J. Friedel, *Dislocations* (Pergamon, New York, 1964).

[10] J. P. Hirth and J. Lothe, *Theory of Dislocations* (Wiley, New York, 1982).

[11] For 200 000 edge dislocations and the accuracy of 1%, Wang and LeSar (Ref. [5]) quote a CPU time of over 300 sec on a Silicon Graphics workstation with a 150 MHz RS 4400 chip. Our full two-dimensional program takes 186 sec on a similar computer ($\epsilon = 10^{-3}$).

[12] I. S. Gradshteyn and I. M. Ryzhik, *Table of Integrals, Series and Products* (Academic, New York, 1980).