Mandlung Summation of Dislocation Dipole Energy by MadSum

Wei Cai

January 19, 1999

Contents

1 Why MadSum?

Let's start with the following problem. Suppose a 2-D (infinite) lattice of dislocation dipoles, as shown in Fig. 1. The dislocation lines are perpendicular to the plane z. For simplicity all these dislocations are screw. The dark discs represent "positive" or right-hand-side screw dislocations, their Burgers vector being $b\hat{z}$, or b for short. The open circles represent "negative" or left-hand-side screw dislocations, their Burgers vector being $-b\hat{z}$, or $-b$ for short. The two lattice vector are $\vec{c}_0$ and $\vec{c}_1$. The "negative" dislocation lattice is offset by $\vec{r}$ from the "positive" dislocation lattice.

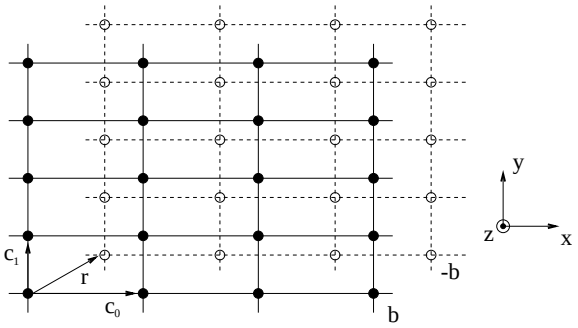

Figure 1: A screw dislocation dipole lattice is the superimpose of a lattice of "positive" (•) screws and a lattice of "negative" (∘) screws, offset by $\vec{r}$ from the former.

The interaction energy [1] W_{12} between two parallel dislocations are predicted by linear elastic theory as the following equation [1]. It is accurate, as long as the distance between the two dislocations is larger than the core radius, where the linear elastic theory breaks down.

$$\begin{aligned} W_{12} &= -\frac{\mu}{2\pi}(\vec{b}_1\cdot\vec{\xi})(\vec{b}_2\cdot\vec{\xi})\ln\frac{R}{r_c} \\ &\quad -\frac{\mu}{2\pi(1-\nu)}[(\vec{b}_1\times\vec{\xi})\cdot(\vec{b}_2\times\vec{\xi})]ln\frac{R}{r_c} \\ &\quad -\frac{\mu}{2\pi(1-\nu)R^2}[(\vec{b}_1\times\vec{\xi})\cdot R][(\vec{b}_2\times\vec{\xi})\cdot R], \end{aligned} \tag{1}$$

where $\vec{b}_{1,2}$ are the Burgers vectors of the two dislocations, ξ is the line direction of the dislocations, R is their separations, μ the shear modulus, ν the Poisson's ratio. In our case of parallel screw dislocations, we have $\vec{b}_{1,2} = \pm b\hat{z}$, and $\vec{\xi} = \hat{z}$. Eq. 1 can be simplified as,

$$W_{12} = -\frac{\mu b_1 b_2}{2\pi}\ln\frac{R}{r_c}, \tag{2}$$

where $b_{1,2} = \pm b$.

Then the question is, what is the interaction energy of the dipole lattice, predicted by the linear elastic theory. Well, the answer is infinity, because there are infinite number of dislocations interacting with each other. The fair question is then, *what is the interaction energy W_c associated with a unit cell, which contains a dipole.* The energy W_c can be interpreted as an *energy density* of the dipole lattice. Note that W_c is a function of $\vec{r}$ alone, where $\vec{r}$ is the offset of the negative

[1] All the energy discussed in this paper refers to energies per unit length along z direction

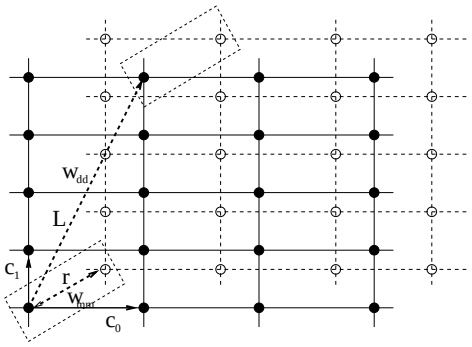

Figure 2: Interaction energy per dipole contains the self interaction term W_{mm} and the primary-image interaction terms W_{dd}.

dislocation w.r.t. the positive one, which we call *dipole separation* from now on. [2] In viewing $W_c(\vec{r})$ as the *energy density* of the dipole lattice, it is easy to see that it is periodic, i.e.

$$W_c(\vec{r}) = W_c(\vec{r} + \vec{c}_{0,1}), \tag{3}$$

because either $\vec{r}$, $\vec{r} + \vec{c}_0$ or $\vec{r} + \vec{c}_1$ represent the same lattice. In Fig. 3, we plot the periodic profile of $W_c(\vec{r})$ as $\vec{r}$ varies over several lattice vectors. This is the realistic calculation result performed by MadSum.

Let's go into detail of the procedure of calculating $W_c(\vec{r})$. It contains two contributions. One is the interaction energy within the dipole, i.e. between the two (opposite) dislocations in the unit cell, call it W_{mm} — monopole-monopole interaction. The other is the interaction energy of the primary dipole with its image dipoles, call it W_{dd} — dipole-dipole interaction. It needs to be summed over all images. This procedure is illustrated in Fig. 2, where the dash lines represented the interaction terms. Expressing it in equations, we have,

$$W_c(\vec{r}) = W_{mm}(\vec{r}) + \frac{1}{2} \sum_{\vec{L}}{}' W_{dd}(\vec{r}, \vec{L}), \tag{4}$$

where $\sum_{\vec{L}}{}'$ means to sum over all lattice vectors $\vec{L}$ except $\vec{L} = 0$ i.e.

$$\begin{aligned} \vec{L} &= i\vec{c}_0 + j\vec{c}_1 \\ i, j &= 0, \pm 1, \pm 2, \cdots \\ i \cdot j &\neq 0 \end{aligned}$$

Applying Eq. 2, we have

$$W_{mm}(\vec{r}) = -\frac{-\mu b^2}{2\pi} \ln(\frac{r}{r_c}), \tag{5}$$

[2] In the later chapters, the *dipole separation* is sometimes represented by $\vec{p}$, to avoid confusion.

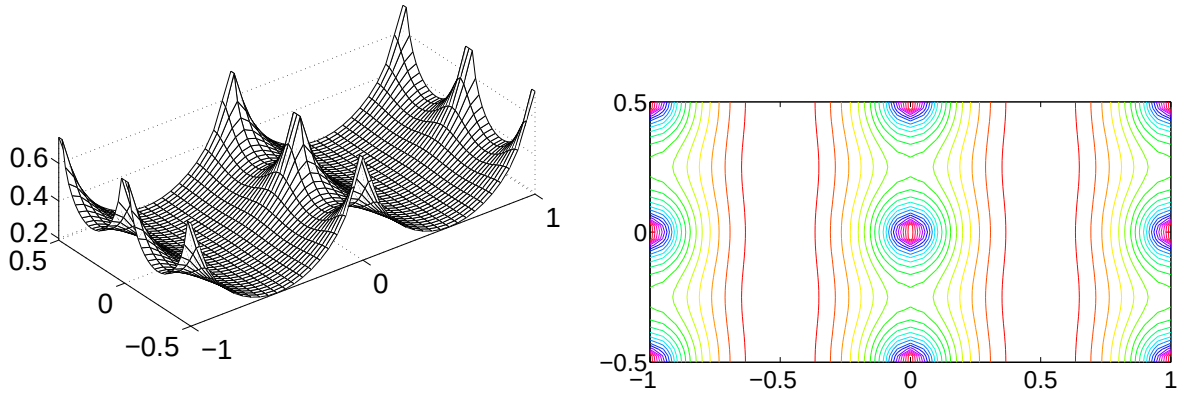

Figure 3: Energy $-W_c$ as a field function of dipole separation $\vec{r}$ by MadSum

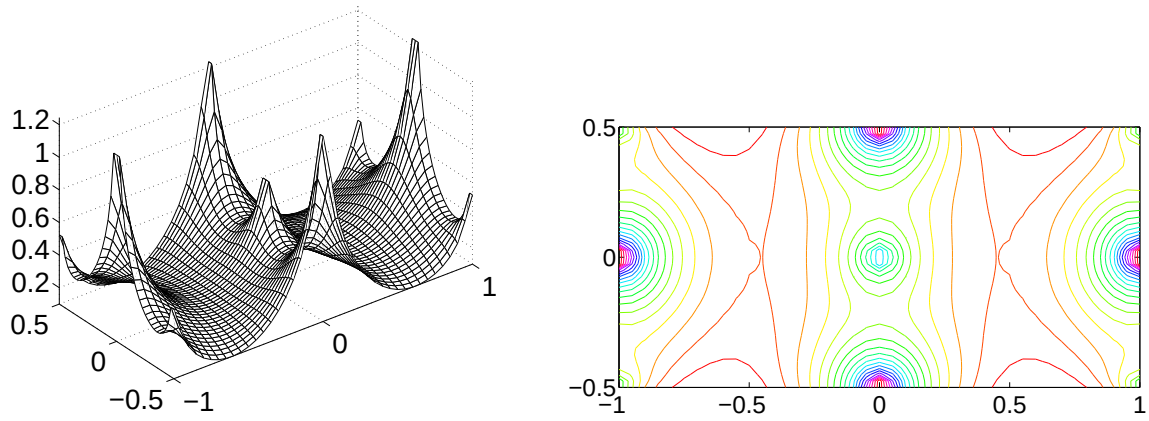

Figure 4: Energy $-W_c$ as a field function of dipole separation $\vec{r}$ by naive summation

and

$$W_{dd}(\vec{r},\vec{L}) = -\frac{\mu b^2}{2\pi}\ln\left(\frac{L^2}{||\vec{L}-\vec{r}||\cdot||\vec{L}+\vec{r}||}\right), \tag{6}$$

where $r = ||\vec{r}||$ and $L = ||\vec{L}||$. Note that there is an overall factor of $1/2$ in front of W_{dd} terms. This is because in the primary-image interactions, only one half of the energy is attributed to the primary dipole.

Eq. 4,5,6 are all the formulas we need to calculate $W_c(\vec{r})$. Numerically, we perform the summation in Eq. 4 over all the lattice vectors $\vec{L}$ within a cut-off, and hope our result will converge as we increase the cut-off range.

Fig. 4 is the result from the summation over 200×200 cells. Surprisingly, the result is **not** periodic! This indicates that there is something tricky in the summation, which needs further attention! However, if we apply Madsum, over the same 200×200 cells, the result is what we have seen before in Fig. 3, which is periodic. Now you see the point I want to make — MadSum is doing the correct thing where naive approach fails.

Note that in both Fig. 3 and 4, $-W_c$ is plotted instead of W_c, for a better view angle. Other parameters are set as $\frac{\mu b^2}{2\pi} = 1$, $\vec{c}_0 = (0,1)$, $\vec{c}_1 = (0,1/2)$. $W_c(\vec{r})$ is actually singular logarithmically, at $\vec{r} = i\vec{c}_0 + j\vec{c}_1$.

2 What's the trick?

2.1 Conditional Convergence

The reason for the failure of direct summation is the problem of Conditional Convergence(CC). According to Eq. 6, the interaction energy term $W_{dd}(\vec{r}, \vec{L})$ is of the order of $O(1/L^2)$, for large L. Recall that $\vec{r}$ is the *dipole separation* and $\vec{L}$ is the distance between the primary and the image dipole. The number of image cells with distance L to the primary cell is on the order of $2\pi L$. Therefore, the summation over all image cells is of the form

$$\sum_{\vec{L}}{}' \sim \sum_{L=1}^{\infty} 2\pi L \frac{1}{L^2} \sim \int_1^{\infty} dL \frac{1}{L} \sim \ln(\infty). \tag{7}$$

The integral does not converge and the summation is called conditionally convergent.

The conditional convergence differs from divergence in that, the partial summation will apparently converge to a limit as the cut-off range is *systematically* increased. However, the number it converges to depends on the shape of the cut-off range. Whether you sum up all the images in a sphere or in a cube will affect the summation it converges to. All different summation schemes give different results, while only a single correct answer exists. That's the problem of CC.

If you try the naive method, your probability of hitting the correct answer has measure zero. But with MadSum, you are guaranteed to converge to the correct answer as long as you sum up enough terms.

2.2 The Old Trick of Sum by Quadruples

This problem of CC in summing dislocation dipole interactions is first noted by Vasily Bulatov. The problem he consider is the following. First, an atomistic calculation is carried out, to find the energy of a simulation cell containing a dislocation dipole. For the convenience of the calculation, periodic boundary condition is applied. Therefore, the energy from atomistic calculation includes both the dipole energy and the interaction energy between the primary dipole with all its images (corresponds to W_{dd}). To obtain the energy of a dislocation dipole embedded in an infinite medium, the latter part, i.e. the summation of primary-image interactions has to be subtracted. The summation has the feature of CC, which we have discussed in the last section. Therefore, direct summation within a cut-off will lead to erroneous results [2, 3], and *non-self-consistent* results, had they ever carry out the summation *again* with a slightly different cut-off range shape.

Following Dieter Wolf's argument of summing Coloumb interactions in NaCl crystals [4], VB suggests a trick to get around the problem of CC. Namely, primitive cells containing single dipoles are grouped into larger boxes, in such way that the net dipole moment cancels and only quadrupole moment remains. The interaction energy between the quadrupole boxes is then on the order of $O(1/L^4)$, whose summation over all lattice vectors are Absolutely Convergent. For details, readers are referred to one of my final term paper *Mandlung Summation in Dislocation Simulations*, which is available at *http://mmm.mit.edu/~caiwei/Archives/Academic/8.512/*. I wrote an implementation of such algorithm in 1997. This is called QuadSum and is included in the distribution of MadSum-1.0.

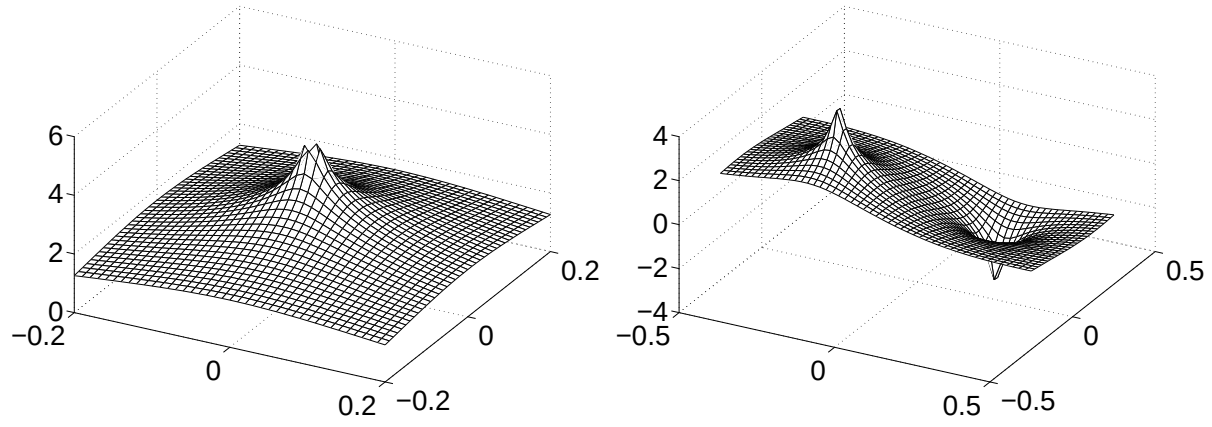

Figure 5: Potential field of a dislocation and a dislocation dipole

Although grouping dipole cells into quadrupole boxes before summation is indeed an ingenious idea, it has certain limitations in implementation. First, the idea only works when the *dipole separation* $\vec{r}$ is parallel to one of the lattice vectors, e.g. $\vec{c}_0$. Although we can always change the lattice vectors into linear combinations of $\vec{c}_0$ and $\vec{c}_1$, and by doing so, we might make our lattice vector parallel to $\vec{r}$, this will not work if $\vec{r}$ points in an irrationally indexed direction. The second constraint is that the magnitude r of the *dipole separation* has to be a rational fraction e.g. p/q, of the lattice vector to which it is parallel. Although we can approximate any real number by a rational number with arbitrary accuracy, calculations shows that for the case of $r = \frac{23}{48}c_0$, the convergence is already very slow. This is because the quadrupole box so constructed consists of 48 dipole cells and its aspect ratio is about $1:48$, which leads much slower convergence rate.

MadSum is designed to remove such constraints so as to provide a fast and correct way to evaluate the Mandlung summation in dislocation dipole interactions. Pictures such as Fig. 3 needs results of 1672 such summations. It takes about 2 hours for MadSum running on PentiumII 450. The same calculation would be impossible for QuadSum, which use the old algorithm.

2.3 MadSum's Trick

In order to understand how MadSum handle the problem of CC, we need to think deeper than just the interaction energy itself. Rather, we need to consider the *potential field* created by the image dipoles.

We can define a potential of a dislocation at an arbitrary point $\vec{r}$ to be the interaction energy between this dislocation and a test dislocation with unit Burgers vector, located at $\vec{r}$. Therefore, a screw dislocation with Burgers vector b generate the potential field of

$$\phi(\vec{r}) = -\frac{\mu b}{2\pi}\ln(\frac{r}{r_c}). \tag{8}$$

We plot the potential of a "positive" dislocation and that of a dislocation dipole in Fig. 5 (a) and (b) respectively.

The potential field of a dislocation dipole lattice is the sum of the potentials of all dipoles of the lattice. We expect this potential field to be periodic, because of the periodicity of the lattice itself. As an example, we plot in Fig. 6(a). the potential field in the primitive cell after summing up the

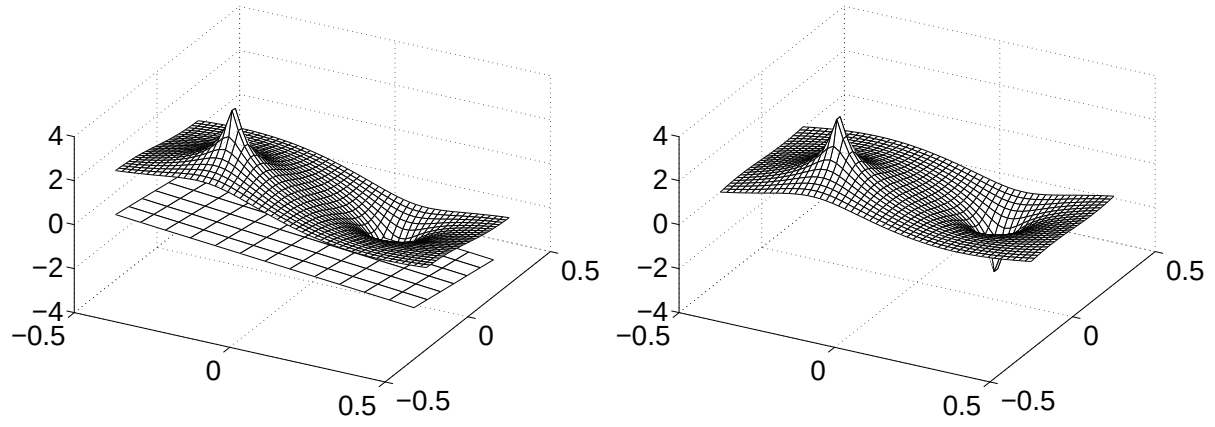

Figure 6: Potential field in the unit cell of a dislocation dipole lattice. (a) obtained from direct summation within 10×10 neighboring cells (b) after subtracting a linear term from (a) to make the potential field periodic

potential fields of all dipoles within the neighboring 10×10 cells. Surprisingly again, we observe that the potential field thus obtained does *not* satisfy the periodic boundary condition! Only after we subtract a linear term (the plane in Fig. 6(a)), we get the "true" periodic potential field of the dipole lattice, plotted in Fig. 6(b).

This kind of behavior is in general the case, and is not a finite size effect, i.e. summing up more terms will not change the non-periodicity. It suggests that, superimposed on the "true" potential field (periodic), the summation over image dipole fields includes an extra linear field, which could be different for different cut-off schemes. This idea is best described through the following theorem.

THEOREM *Suppose the summation of the dipole lattice potential field with two different cut-off schemes converges to $\phi_1(\vec{r})$ and $\phi_2(\vec{r})$ respectively. Then $\phi_1(\vec{r})$ and $\phi_2(\vec{r})$ are identical up to a linear term,*

$$\phi_1(\vec{r}) - \phi_2(\vec{r}) = \phi_0 - \vec{E} \cdot \vec{r}, \tag{9}$$

where ϕ_0 and $\vec{E}$ are constant scalar and constant vector depending on the specific numerical schemes.

Proof For any point (in the unit cell) $\vec{r}_0$, the value of $\nabla\nabla\phi_1(\vec{r}_0)$ is *absolutely convergent*. This can be seen from the following equations, where I use $\vec{p}$ to represent the dipole separation,

$$\begin{aligned}
\phi_1(\vec{r}_0) &\sim \sum_{\vec{L}} \ln \frac{||\vec{L} - \vec{r}_0||}{||\vec{L} - \vec{r}_0 - \vec{p}||} \\
&\sim \sum_{\vec{L}} L^{-1} \\
&\sim \int^{\infty} dL 2\pi L \cdot L^{-1} \qquad (10) \\
\nabla\phi_1(\vec{r}_0) &\sim \sum_{\vec{L}} \nabla(\ln \frac{||\vec{L} - \vec{r}||}{||\vec{L} - \vec{r} - \vec{p}||})_{r=r_0} \\
&\sim \frac{1}{||\vec{L} - \vec{r}_0||} - \frac{1}{||\vec{L} - \vec{r}_0 - \vec{p}||}
\end{aligned}$$

$$
\begin{aligned}
&\sim \sum_{\vec{L}} L^{-2} \\
&\sim \int^{\infty} dL 2\pi L \cdot L^{-2} \qquad (11)
\end{aligned}
$$

$$
\begin{aligned}
\nabla\nabla\phi_1(\vec{r}_0) &\sim \sum_{\vec{L}} \nabla\nabla(\ln \frac{||\vec{L}-\vec{r}||}{||\vec{L}-\vec{r}-\vec{p}||})_{r=r_0} \\
&\sim \frac{1}{||\vec{L}-\vec{r}_0||^2} - \frac{1}{||\vec{L}-\vec{r}_0-\vec{p}||^2} \\
&\sim \sum_{\vec{L}} L^{-3} \\
&\sim \int^{\infty} dL 2\pi L \cdot L^{-3} \qquad (12) \\
& \qquad (13)
\end{aligned}
$$

Note that unlike Eq. 4, there is no prime sign in the summation here, which means that the contribution of the primary dipole needs to be included. We observe that the potential itself is not convergent at all! [3] The gradient of the potential is shown to be *conditionally convergent*, and the double gradient of the potential is *absolutely convergent*.

Absolute convergence means that if two summation with different cut-off schemes converges to $\phi_1(\vec{r})$ and $\phi_2(\vec{r})$ respectively, then

$$\nabla\nabla(\phi_1(\vec{r}) - \phi_2(\vec{r})) = 0. \qquad (14)$$

Integrate the above equation would give rise to

$$\phi_1(\vec{r}) - \phi_2(\vec{r}) = \phi_0 - \vec{E}\cdot\vec{r},$$

thus proves the theorem.

The constant ϕ_0 will not affect our result of the dipole-image interaction energy calculation, because the its effect on the two opposite dislocations of the dipole cancel. The constant vector $\vec{E}$ does alter our dipole-image interaction energy, however. And we have to be sure which value $\vec{E}$ should take.

The correct value of $\vec{E}$ is zero. There are two reasons for this. First, $\vec{E}$ has to be zero to ensure the periodicity of the lattice potential. [4] Second, a non-zero $\vec{E}$ corresponds to the presence of a finite internal macroscopic stress field in the medium. If no external stress is applied to the system, $\vec{E}$ will be zero to lower the energy of the system. If external stress is applied, then $\vec{E}$ will be non-zero so as to balance the external stress. But still, $\vec{E}$ is directly related to the external stress. Its value is a definite number, rather than a flimsy unknown, introduced "unpredictably" from our summation cut-off scheme. [5]

Now we have to estimate the value of $\vec{E}$ introduced in a given summation. Note that for if a potential field satisfies periodic boundary condition, $\phi(\vec{r}_0)$ and $\phi(\vec{r}_0 + \vec{c}_{0,1})$ should equal, for arbitrary $\vec{r}_0$. Any difference between them indicates a finite $\vec{E}$, i.e.

$$\phi(\vec{r}_0 + \vec{c}_0) - \phi(\vec{r}_0) = -\vec{E}\cdot\vec{c}_0 \qquad (15)$$

$$\phi(\vec{r}_0 + \vec{c}_1) - \phi(\vec{r}_0) = -\vec{E}\cdot\vec{c}_1 \qquad (16)$$

[3] This is no problem, because in calculating the dipole interaction energy, we never need to evaluate the potential itself. The potential field is introduced only for theoretical convenience.

[4] A characteristic which we believe is the case intuitively.

[5] Every cut-off scheme will introduce its own effective $\vec{E}$, with no simple relationship between the cut-off and $\vec{E}$. However, for a given summation, the value of $\vec{E}$ it introduces can be calculated by the method discussed later.

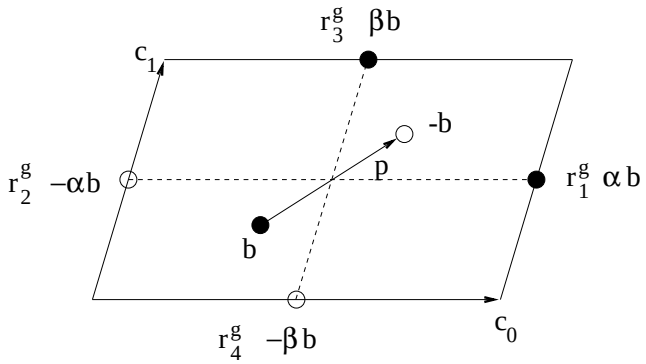

Figure 7: Positioning of "ghost" dipoles in MadSum

The effect of $\vec{E}$ on the interaction energy summation is easy to write down. If the primary dipole has *dipole separation* $\vec{p} = \alpha\vec{c}_0 + \beta\vec{c}_1$, the interaction energy between the dipole and $\vec{E}$ is

$$\Delta W = b\vec{p}\cdot\vec{E}, \tag{17}$$

where b is the Burgers vector of the positive dislocation of the dipole. One half [6] of ΔW needs to be subtracted from our summation, i.e. the correction to Eq. 4 is now

$$W_c(\vec{r}) = W_{mm}(\vec{r}) + \frac{1}{2}{\sum_{\vec{L}}}' W_{dd}(\vec{r},\vec{L}) - \frac{1}{2}\Delta W, \tag{18}$$

Since Eq. 15,16 are valid for any $\vec{r}_0$, we can write Eq. 17 as

$$\begin{aligned}\Delta W &= b(\alpha\vec{c}_0 + \beta\vec{c}_1)\cdot\vec{E} \\ &= -b\alpha[\phi(\vec{r}_1^g) - \phi(\vec{r}_2^g)] - b\beta[\phi(\vec{r}_3^g) - \phi(\vec{r}_4^g)],\end{aligned} \tag{19}$$

where I name $\vec{r}_{1,2,3,4}^g$ as position of "ghost" dislocations, which satisfies

$$\vec{r}_1^g - \vec{r}_2^g = \vec{c}_0 \tag{20}$$

$$\vec{r}_3^g - \vec{r}_4^g = \vec{c}_1. \tag{21}$$

Now we note that to *subtract* ΔW is equivalent to *add* the interaction energy between the dipole lattice footnoteIncluding the primary dipole and all the images which are counted in the interaction energy summation. and two "ghost" dipoles, or say four "ghost" dislocations, which has Burgers vector αb,$-\alpha b$,βb,$-\beta b$, and located at $\vec{r}_1^g$,$\vec{r}_2^g$,$\vec{r}_3^g$,$\vec{r}_4^g$, respectively. Fig. 7 illustrates how MadSum choose the position of the "ghost" dislocations. Although the choice of $\vec{r}_{1,2,3,4}^g$ will not change the results, theoretically, as long as Eq. 20,21 are satisfied. The symmetric arrangement shown in Fig. 7 is believed to give the best numerical behavior. In summary, the corrected formula for Eq. 4 is

$$W_c(\vec{r}) = W_{mm}(\vec{r}) + \frac{1}{2}{\sum_{\vec{L}}}' W_{dd}(\vec{r},\vec{L}) + \frac{1}{2}\sum_{\vec{L}} W_{dg}(\vec{r},\vec{L})$$

[6] For the same reason as that of Eq. 4.

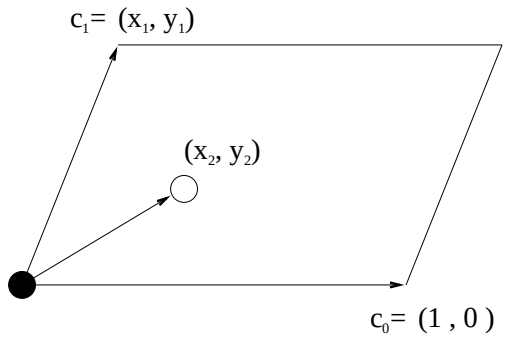

Figure 8: Parameter specification of function `madsum()`

$$
\begin{aligned}
&= W_{mm}(\vec{r}) + \frac{1}{2} W_{dg}(\vec{r}, 0) \\
&\quad + \frac{1}{2} {\sum_{\vec{L}}}' W_{dd}(\vec{r}, \vec{L}) + W_{dg}(\vec{r}, \vec{L}),
\end{aligned} \tag{22}
$$

where W_{dg} is the interaction energy of a dipole with the "ghost" dipoles, which reads

$$
\begin{aligned}
W_{dg}(\vec{r}, \vec{L}) = -\frac{\mu b}{2\pi} \Bigg\{ & \alpha \ln \frac{||\vec{L} - \vec{p}/2 + \vec{c}_0/2|| \cdot ||\vec{L} + \vec{p}/2 + \vec{c}_0/2||}{||\vec{L} - \vec{p}/2 - \vec{c}_0/2|| \cdot ||\vec{L} + \vec{p}/2 - \vec{c}_0/2||} \\
& + \beta \ln \frac{||\vec{L} - \vec{p}/2 + \vec{c}_1/2|| \cdot ||\vec{L} + \vec{p}/2 + \vec{c}_1/2||}{||\vec{L} - \vec{p}/2 - \vec{c}_1/2|| \cdot ||\vec{L} + \vec{p}/2 - \vec{c}_1/2||} \Bigg\}
\end{aligned} \tag{23}
$$

3 User's Manual

3.1 Function Interface

The main function implemented in MadSum is

```
void madsum(double coords[4], double sums[4]);
```

`coords[4]` specifies the geometry of the dipole lattice,

$$
\begin{aligned}
\texttt{coords[0]} &= x_1, \\
\texttt{coords[1]} &= y_1, \\
\texttt{coords[2]} &= \hat{x}_2, \\
\texttt{coords[3]} &= \hat{y}_2,
\end{aligned}
$$

where (x_1, y_1) is the coordinate of the lattice vector $\vec{c}_1$, $(\hat{x}_2, \hat{y}_2)$ is the *relative* coordinate of the primary dipole, as shown in Fig. 8. The lattice vector $\vec{c}_0$ is fixed at $(0, 1)$. Therefore, the coordinate of the primary dipole is $(x_2, y_2) = (\hat{x}_2 + \hat{y}_2 x_1, \hat{y}_2 y_1)$.

The output is not the dipole energy itself, but four summation terms,

$$
\texttt{sums[0]} = \sum \ln r,
$$

$$\begin{aligned} \texttt{sums[1]} &= \sum \frac{x^2}{r^2}, \\ \texttt{sums[2]} &= \sum \frac{y^2}{r^2}, \\ \texttt{sums[3]} &= \sum \frac{xy}{r^2}, \end{aligned}$$

where $r = (x^2 + y^2)^{1/2}$ and $\sum$ may or may not include the primary dipole contribution, depending on the value of some global variables, which will be explained later.

According to Eq. 1, the dipole interaction energy can be easily be calculated as linear combinations from the four summations produced by `madsum()`. The following function

```
double evalenergy(double sums[4], double bx, double by, double bz,
                  double nu, double mu);
```

is designed to do this for you. $\vec{b} = (\texttt{bx}, \texttt{by}, \texttt{bz})$ specifies the Burgers vector. $\nu = \texttt{nu}$ is the Poisson's ratio and $\mu = \texttt{mu}$ is the shear modulus.

The following 8 global variables can tune the performance of function `madsum()`,

```
extern int _CUTX, _CUTY, _THICKX, _THICKY;
extern int _noghost, _verbose, _primary, _primaryonly;
```

where `_CUTX, _CUTY` (default 200) set the cut off range of the summation in $\vec{c}_0$ and $\vec{c}_1$ directions, respectively. The summation are performed layer by layer with thickness specified by `_THICKX, _THICKY` (default 20), which would not change the output of `madsum()` unless you want to print out the intermediate results per layer.

Remember that `madsum()` calculate the correct summation by introducing two "ghost" dipoles. If `_noghost` is set to 1 (default 0), then no such "ghost" dipoles are introduced, and you can see by how much the naive summation is wrong. If `_verbose` is set to 1 (default 0), `madsum()` will print out (into `stderr`) the intermediate results per layer, whose thickness is specified by `_THICKX, _THICKY`.

By default, only the second term (the summation over images) in Eq. 4 is calculated. If `_primary` is set to 1 (default 0), then the first term (interaction within the primary dipole) $W_{mm}(\vec{r})$ is included also, and the result corresponds to the interaction energy of the dipole lattice $W_c(\vec{r})$. If `_primaryonly` is set to 1 (default 0), then only $W_{mm}(\vec{r})$ is calculated.

If you want to use MadSum to do a lot of calculations, you will need to call the function `madsum()` in your code. `madmesh.c` is such an example, which generate the data point for Fig. 3, 4.

3.2 Running MadSum by Command Lines

You may want to see how `madsum()` works without bothering with source code itself, or any additional programming. The command `madd` is designed for this purpose.

After you obtain the source code of MadSum-1.0, you need to go to the directory of the `madsum()` source codes and type the following to build the executables, [7]

```
setenv SYS `uname`
make
```

[7] Do not mix ` with ' in the first command line. If you are not sure, just skip this line.

The code has been successfully compiled on Linux and IRIX64 machines. If there is problem with other platforms, you may want to contact with the author at `caiwei@mit.edu`. The first command is usually not mandatory. After this you should have executables `madd madmesh quadsum test` in your directory.

If you run `madd` without any arguments, you will get the following message,

```
% ./madd
Usage: madd [options] <x1> <y1> <x2r> <y2r> [-b <bx> <by> <bz> <nu> <mu>]
options:
         -v --verbose            dump output per layer
         -n --noghost            without ghost dipoles
         -cx=<n> --cutx=<n>      set cut-off in x (200)
         -cy=<n> --cuty=<n>      set cut-off in y (200)
         -tx=<n> --thickx=<n>    set layer thickness in x (20)
         -ty=<n> --thicky=<n>    set layer thickness in y (20)
defaults:
         x1=y1=x2r=y2r=bx=by=bz=0
         nu=0.3
         mu=1
ranges:
         -0.5 < x1,x2r,y2r < 0.5
         y1 > 0
```

This shows that there are several options from which you can set the global variables described in the last section. Let's run `madd` to calculate the Mandlung Summation with $x_1 = 0$,$y_1 = 0$,$\hat{x}_2 = 0.5$,$\hat{y}_1 = 0$,

```
% ./madd -v 0 0.5 0.5 0
5.4171083779e-01 -1.1883704238e+00 1.1883704238e+00 1.2101458073e-15
5.4164645510e-01 -1.1884219013e+00 1.1884219013e+00 2.4966058951e-15
5.4163423009e-01 -1.1884316797e+00 1.1884316797e+00 6.4750993196e-15
5.4162991216e-01 -1.1884351337e+00 1.1884351337e+00 8.4056161554e-15
5.4162790394e-01 -1.1884367402e+00 1.1884367402e+00 3.7143411177e-15
5.4162680979e-01 -1.1884376155e+00 1.1884376155e+00 1.7014249168e-15
5.4162614869e-01 -1.1884381444e+00 1.1884381444e+00 1.4072402098e-15
5.4162571898e-01 -1.1884384882e+00 1.1884384882e+00 -1.0090805145e-15
5.4162542403e-01 -1.1884387241e+00 1.1884387241e+00 1.0785101111e-15
5.4162530516e-01 -1.1884388192e+00 1.1884388192e+00 4.9085356005e-15

            _slnr             _sxx             _syy             _sxy
   fmad  5.4162530516e-01 -1.1884388192e+00 1.1884388192e+00 4.9085356005e-15
+ 2fprm  1.3862943611e+00 -2.0000000000e+00 0.0000000000e+00 0.0000000000e+00
=  ftot  1.9279196663e+00 -3.1884388192e+00 1.1884388192e+00 4.9085356005e-15
```

`fmad` shows the various terms in dipole-image interaction. When added with `2fprm` i.e., twice the primary dipole energy, `ftot` is ready to be used to calculate the total energy of the lattice. If you tell `madd` the Burgers vector $\vec{b}$, Poisson's Ratio ν and shear modulus μ, it will print out the corresponding energy. This time let's suppress the intermediate result output, by omitting the `-v` switch in the command line.

```
% ./madd 0 0.5 0.5 0 -b 0 0 1

           _slnr             _sxx             _syy             _sxy
   fmad  5.4162530516e-01 -1.1884388192e+00 1.1884388192e+00 4.9085356005e-15
+ 2fprm  1.3862943611e+00 -2.0000000000e+00 0.0000000000e+00 0.0000000000e+00
=  ftot  1.9279196663e+00 -3.1884388192e+00 1.1884388192e+00 4.9085356005e-15

Emad = -1.2357600888e+00
Eprm = -9.9021025794e-01
Etot = -2.2259703468e+00
```

The Burgers vector is set to $\vec{b} = (0, 0, 1)$ (screw). $\nu = 0.3$ and $\mu = 1$ are the default values. The output `Emad` is one half of the dipole-image interaction energy, because only one half of it is associated to the primary cell. `Eprm` is the interaction within the primary dipole. `Etot = Emad + Eprm` is the energy of the dipole lattice, per unit cell and per unit length.

We will see that `madd` is consistent with `quadsum`, which is an older version of the Mandlung summation program, using the algorithm of grouping dipoles into quadrupole boxes before summation. Let's run `quadsum`

```
% ./quadsum
input a[0] a[1] p[0] p[1] in sequence, please:
0 0.5 1 2
layer[10]: 0 1 2 3 4 5 6 7 8 9 10
0.541624 -1.188379 1.188379 0.000056
```

The inputs `a[0]` and `a[1]` are the same as `x1` and `y1` in `madsum`, which specify the coordinate of $\vec{c}_1$ (Fig. 8). In `quadsum`, the dipole is parallel to $\vec{c}_0$. The length of the dipole is a rational fraction of the length of $\vec{c}_0$ (kept at constant $(1, 0)$), specified by `p[0]/p[1]`. The four output is to be compared with the `fmad` entry of that of `madsum`. We observe that the comparison is very good. The result of `madsum` is believed to be more accurate, e.g. it gives a sharper zero in the last entry.

If we pass the option `-n` to `madsum`, it will mimic the naive way of summation, disregarding the "ghost" dipoles.

```
% ./madd -n 0 0.5 0.5 0

           _slnr             _sxx             _syy             _sxy
   fmad  7.1294171248e-01 -1.3157339045e+00 1.3157339045e+00 3.7274870690e-15
+ 2fprm  1.3862943611e+00 -2.0000000000e+00 0.0000000000e+00 0.0000000000e+00
=  ftot  2.0992360736e+00 -3.3157339045e+00 1.3157339045e+00 3.7274870690e-15
```

If we change the cut-off range (from the default 200), we will see that the naive summation converges to yet another result.

```
% ./madd -n -cx=300 0 0.5 0.5 0

           _slnr             _sxx             _syy             _sxy
   fmad  9.9623648815e-01 -1.2320403249e+00 1.2320403249e+00 1.2448240138e-14
+ 2fprm  1.3862943611e+00 -2.0000000000e+00 0.0000000000e+00 0.0000000000e+00
=  ftot  2.3825308493e+00 -3.2320403249e+00 1.2320403249e+00 1.2448240138e-14
```

The reader is encouraged to see whether or not the result of the correct `madsum` (without `-n` option) will depend on cut-off settings.

References

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

[2] T.A.Arias and J.D.Joannopoulos, *Ab Initio Theory of Dislocation Interactions: From Close-Range Spontaneous Annihilation to the Long-Range Continuum Limit*, Phys. Rev. Lett. **73**, 680 (1994)

[3] L.B.Hansen, K.Stokbro, B.I.Lundqvist, K.W.Jacobsen, and D.M.Deaven, *Nature of Dislocation in Silicon*, Phys. Rev. Lett. **75**, 4444 (1995)

[4] D. Wolf, *Reconstruction of NaCl Surfaces from a Dipolar Solution to the Madelung Problem*, Phys. Rev. Lett. **68**, 3315 (1992)