

## A MULTIGRID-FOURIER METHOD FOR THE COMPUTATION OF ELASTIC FIELDS WITH APPLICATION TO HETEROEPITAXY\*

GIOVANNI RUSSO<sup>†</sup> AND PETER SMEREKA<sup>‡</sup>

**Abstract.** A multigrid-Fourier method is formulated for the computation of elastic fields with applications to heteroepitaxy. A discrete ball and spring model with an underlying cubic structure is considered, where the natural lattice spacing of the atoms comprising the deposited film is different than those of the substrate. This system is linearized resulting in a large linear system for the displacement field. An efficient method based on combining Fourier and multigrid formulations to solve this system is presented. In this algorithm, the atoms in the deposited film and the substrate atoms are handled differently. The equations for the elastic displacement of atoms in the film are extended to a rectangular region by the use of fictitious atoms and a connectivity matrix, allowing the application of standard multigrid ideas. Except for the top layer, the atoms in the substrate are completely removed and replaced by equivalent forces which can be efficiently evaluated using a fast Fourier transform. This formulation has been implemented in both two and three dimensions using V-cycles. It is found that the number of V-cycles needed to reach a certain level of accuracy is essentially independent of the system size. Numerical tests show that for large domains, the multigrid-Fourier method is approximately 6 to 10 times faster than conjugate-gradient-based methods.

**Key words.** multigrid, elasticity, heteroepitaxy

**AMS subject classifications.** 74B15, 74S20, 82-08

**DOI.** 10.1137/05063800X

**1. Introduction.** Heteroepitaxial growth is a process in which one atomic species is deposited upon a substrate of another. If the natural atomic spacing of the two species is different, then elastic effects can become important. In particular the film can lower its elastic energy by forming three-dimensional islands. Therefore, layer-by-layer growth can be destabilized by elastic effects. It has been suggested that new materials with interesting electronic properties could be developed by exploiting this instability.

One modeling approach is based on a discrete solid-on-solid model in which the system is evolved using kinetic Monte Carlo (KMC). One of first models, if not the first, in this direction is due to Orr et al. [14]. Later Lam, Lee, and Sander [10] provided a more efficient implementation of this model based on using a discrete Green's function, allowing them to perform two-dimensional simulations in larger domains and to use parameter values that were more physically reasonable. This work was extended to three dimensions by Lung, Lam, and Sander [9]. Russo and Smereka [18] also extended the model due to Orr et al. to three dimensions. In their formulation the force field in the substrate is determined by an exact solution which can be efficiently evaluated using a fast Fourier transform, whereas the force field of the deposited atoms is evaluated directly. The system for the displacement field is solved by an application of the conjugate gradient method.

---

\*Received by the editors August 11, 2005; accepted for publication September 27, 2005; published electronically March 24, 2006.

<http://www.siam.org/journals/mms/5-1/63800.html>

<sup>†</sup>Dipartimento di Matematica e Informatica, Università di Catania, Catania, Italy (russo@dmf.unict.it). This author was supported in part by grants from the Michigan Center for Theoretical Physics and the Italian Government.

<sup>‡</sup>Department of Mathematics and Michigan Center for Theoretical Physics, University of Michigan, Ann Arbor, MI 48109-1109 (psmerek@umich.edu). This author was supported in part by NSF grants DMS-0207402, DMS-0244419, and DMS-0509124.

Ratsch et al. [17] examined three-dimensional heteroepitaxy; however, they did not take explicitly into account the harmonic forces between atoms, but rather they used an approximate treatment [16] based on the Frenkel–Kontorova model. The model was used to investigate the island size distribution in heteroepitaxial growth [15]. Off lattice KMC simulations of heteroepitaxial growth in 1+1 dimensions were presented in a series of papers [12, 1, 13]. In these computations the forces between atoms were modeled using Lennard–Jones interactions. The misfit is easily incorporated by changing parameters in the potential. One advantage of this approach is that dislocations are naturally included, which is not the case with the ball and spring model.

The simulation of the solid-on-solid model by KMC is computationally challenging because for each configurational change of the atoms one must find the new displacement field so that the system is in mechanical equilibrium. This paper focuses on the efficient computation of strain field in the solid-on-solid model. We extend the work in [18] by replacing the conjugate gradient method with a multigrid formulation, thereby combining multigrid and Fourier methods. It might seem natural, for a discrete problem such as this, to use an algebraic multigrid method. Nevertheless, for solid-on-solid models there is enough structure to make it possible to formulate a multigrid model. Our approach is based on introducing fictitious atoms and something we call a connectivity matrix, which encodes how atoms are connected to each other. The discrete elastic equations coarse-grain in a straightforward fashion when written using the fictitious atoms and the connectivity matrix. In addition, the substrate forces, computed using a Fourier formulation, also can be coarse-grained quite easily. We mention that another approach to this problem could be to apply a multigrid method for nonrectangular boundaries [2] combined with a multigrid approach in the substrate. For the latter one would use a different substrate thickness for each multigrid level; see Brandt [4, section 6] for more details.

**2. Solid-on-solid model.** We shall use a three-dimensional version of the model proposed in [14, 10]. For the convenience of the reader we shall now describe it. To fix ideas we shall assume that the deposited atoms are germanium and the substrate is composed of silicon. The atoms occupy sites arranged on a simple cubic lattice with no over hanging atoms allowed. This means that the height of the surface is a function of the horizontal location. The elastic effects in this model are taken into account by assuming that the bonds will act like a spring between the atoms. We will use  $a_s$  and  $a_g$  to denote the lattice spacing between silicon and germanium atoms, respectively. We shall denote, respectively, by  $k_L$  and  $k_D$  the spring constants corresponding to longitudinal (nearest neighbor) and diagonal (next nearest neighbor) bonds. For ease of exposition, we shall assume that both silicon and germanium have the same spring constants. Since  $a_g \neq a_s$ , a mechanical force will arise (the calculation of which is described in detail below). In many systems, the time taken for sound waves to propagate across the sample is much smaller than the time scales associated to any growth process. Therefore we assume that our mass-spring system is always in mechanical equilibrium. In simulations of such a system with KMC the atomic displacement due to elastic effects must be performed repeatedly and as a consequence it is crucial to have a highly efficient algorithm for the computation of the elastic field.

**3. Elastic computations.** In this section we outline the basic systems of equations that are satisfied by the atomistic displacements. For the basic setup we follow Lam, Lee, and Sander [10] and Russo and Smereka [18].

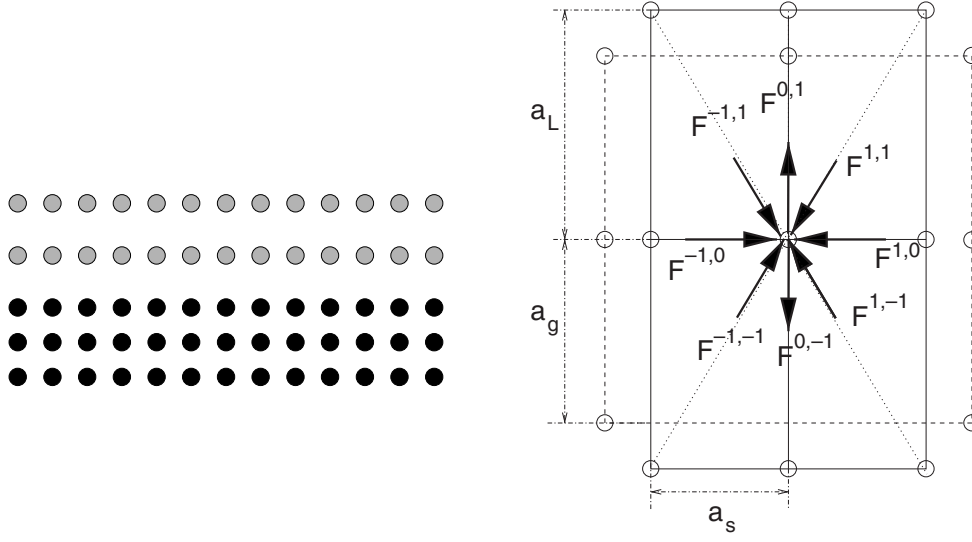


FIG. 1. The reference configuration is obtained by compressing the germanium atoms to have the same horizontal spacing as the silicon atoms. The vertical spacing is chosen so that complete layers of germanium are in mechanical equilibrium. The figure on the left shows two complete layers of germanium in mechanical equilibrium and the figure on the right shows the configurational forces.

**3.1. The reference configuration.** The reference configuration we choose consists of a periodic array of complete layers of germanium atoms on top of a periodic array of silicon. The germanium atoms are compressed so that their horizontal lattice spacing matches that of the silicon atoms; see Figure 1. The vertical lattice spacing,  $a_L$ , is chosen so that the resulting system is in mechanical equilibrium. We will now describe the computation of  $a_L$  in two dimensions. It is useful to introduce the dimensionless quantity

$$\epsilon = \frac{a_g - a_s}{a_s},$$

which is denoted as the *misfit*. Typical values of  $\epsilon$  range from  $-0.05$  to  $0.05$ . For example, the misfit for germanium on a silicon substrate is  $0.04$ . In order to deduce the atom displacement with respect to the reference configuration we need to compute the forces experienced by an atom due to each of its neighbors. Elementary considerations show that, to first order in the ratio  $\epsilon$ , one has

$$\vec{F}^{1,0} = \begin{pmatrix} F_H \\ 0 \end{pmatrix}, \quad \vec{F}^{1,1} = \begin{pmatrix} F_{DV} \\ F_{DV} \end{pmatrix}, \quad \vec{F}^{0,1} = \begin{pmatrix} 0 \\ F_V \end{pmatrix}, \quad \vec{F}^{-1,1} = \begin{pmatrix} -F_{DV} \\ F_{DV} \end{pmatrix},$$

$$\vec{F}^{-1,0} = -\vec{F}^{1,0}, \quad \vec{F}^{-1,-1} = -\vec{F}^{1,1}, \quad \vec{F}^{1,-1} = -\vec{F}^{-1,1}, \quad \text{and} \quad \vec{F}^{0,-1} = -\vec{F}^{0,1},$$

where  $F_V = k_L(a_L - a_g)$ ,  $F_H = k_L(a_s - a_g)$ , and  $F_{DV} = k_D(a_L + a_s - 2a_g)/2$ .

The value of  $a_L$  is determined by requiring that

$$\sum_{m=-1}^1 \vec{F}^{m1} \cdot (0, 1)^T = 0,$$

which implies

$$k_D(2a_g - a_L - a_s) + k_L(a_g - a_L) = 0$$

and gives the following expression for  $a_L$ :

$$a_L = a_g + a_s \epsilon \frac{k_D}{k_L + k_D}.$$

A similar argument can be applied to the three-dimensional lattice; the detailed description can be found in [18]. The force experienced by an atom at site  $(\ell, j)$ , by a neighbor at site  $(\ell + m, j + n)$  is denoted by  $\vec{F}_{\ell j}^{mn}$ , where  $(m, n) \in \{-1, 0, 1\}$ . Using the above results one finds

$$(1) \quad \vec{F}_{\ell j} = \begin{pmatrix} F_{\ell j} \\ G_{\ell j} \end{pmatrix} = \sum_{m,n=-1}^1 \vec{F}_{\ell j}^{mn},$$

where it is understood that the sum is taken over sites that actually contain neighbors. This can be conveniently rewritten by introducing for each site  $(\ell, j)$  a *connectivity matrix*,  $\sigma_{i,j;m,n}$ , defined as

$$(2) \quad \sigma_{i,j;m,n} = \begin{cases} 1 & \text{if site } (i+m, j+n) \text{ contains an atom,} \\ 0 & \text{if site } (i+m, j+n) \text{ contains no atom,} \end{cases}$$

where, again,  $(m, n) \in \{-1, 0, 1\}$ . In this way, Eq. (1) can be rewritten as

$$(3) \quad \vec{F}_{\ell j} = \sum_{m,n=-1}^1 \sigma_{\ell,j;m,n} \vec{F}_{\ell j}^{mn}.$$

**3.2. Computation of the interaction.** We will consider the situation where  $j \geq 2$  are potential sites for deposited atoms and  $j \leq 1$  are atoms in the substrate. Let us denote the displacement, with respect to the reference configuration, of an atom at site  $(\ell, j)$  by the vector  $(u_{\ell j}, v_{\ell j})^T$ . In our formulation all the atoms below the first layer of substrate ( $j \leq 0$ ) are removed and replaced by forces that act on the atoms at  $j = 1$ . Since the system is in mechanical equilibrium, the net force on each atom has to be zero and this is expressed by the following system  $\forall \ell$  and  $j \geq 1$ :

$$(4) \quad \begin{aligned} F_{\ell j} = & k_L (\sigma_{\ell,j;1,0}(u_{\ell+1,j} - u_{\ell j}) + \sigma_{\ell,j;-1,0}(u_{\ell-1,j} - u_{\ell j})) \\ & + \frac{k_D}{2} (\sigma_{\ell,j;1,1}(u_{\ell+1,j+1} - u_{\ell j}) + \sigma_{\ell,j;-1,1}(u_{\ell-1,j+1} - u_{\ell j})) \\ & + \frac{k_D}{2} (\sigma_{\ell,j;1,-1}(u_{\ell+1,j-1} - u_{\ell j}) + \sigma_{\ell,j;-1,-1}(u_{\ell-1,j-1} - u_{\ell j})) \\ & + \frac{k_D}{2} (\sigma_{\ell,j;1,1}(v_{\ell+1,j+1} - v_{\ell j}) - \sigma_{\ell,j;-1,1}(v_{\ell-1,j+1} - v_{\ell j})) \\ & + \frac{k_D}{2} (-\sigma_{\ell,j;1,-1}(v_{\ell+1,j-1} - v_{\ell j}) + \sigma_{\ell,j;-1,-1}(v_{\ell-1,j-1} - v_{\ell j})) + f_{\ell j} \end{aligned}$$

and

$$(5) \quad \begin{aligned} G_{\ell j} = & k_L (\sigma_{\ell,j;0,1}(v_{\ell j+1} - v_{\ell j}) + \sigma_{\ell,j;0,-1}(v_{\ell j-1} - v_{\ell j})) \\ & + \frac{k_D}{2} (\sigma_{\ell,j;1,1}(v_{\ell+1,j+1} - v_{\ell j}) + \sigma_{\ell,j;-1,1}(v_{\ell-1,j+1} - v_{\ell j})) \\ & + \frac{k_D}{2} (\sigma_{\ell,j;1,-1}(v_{\ell+1,j-1} - v_{\ell j}) + \sigma_{\ell,j;-1,-1}(v_{\ell-1,j-1} - v_{\ell j})) \\ & + \frac{k_D}{2} (\sigma_{\ell,j;1,1}(u_{\ell+1,j+1} - u_{\ell j}) - \sigma_{\ell,j;-1,1}(u_{\ell-1,j+1} - u_{\ell j})) \\ & + \frac{k_D}{2} (-\sigma_{\ell,j;1,-1}(u_{\ell+1,j-1} - u_{\ell j}) + \sigma_{\ell,j;-1,-1}(u_{\ell-1,j-1} - u_{\ell j})) + g_{\ell j}. \end{aligned}$$

Note that in the above system, the term containing the unknown is written on the right-hand side, while the left-hand side contains known quantities. We point out that  $f_{\ell,j}$  and  $g_{\ell,j}$  are zero for  $j > 1$ , whereas the terms  $f_{\ell,1}$  and  $g_{\ell,1}$  represent the substrate forces from atoms at  $j = 0$  and below. These are given by

$$(6) \quad \begin{aligned} f_{\ell,1} = & \frac{k_D}{2}(u_{\ell+1,0} + u_{\ell-1,0} - 2u_{\ell,1}) \\ & + \frac{k_D}{2}(v_{\ell-1,0} - v_{\ell+1,0}), \end{aligned}$$

$$(7) \quad \begin{aligned} g_{\ell,1} = & k_L(-v_{\ell,1} + v_{\ell,0}) \\ & + \frac{k_D}{2}(v_{\ell+1,0} + v_{\ell-1,0} - 2v_{\ell,1}) \\ & + \frac{k_D}{2}(u_{\ell-1,0} - u_{\ell+1,0}). \end{aligned}$$

To evaluate  $f_{\ell,1}$  and  $g_{\ell,1}$  we need to determine  $u_{\ell,0}$  and  $v_{\ell,0}$ . This can be done using the method presented in [18], which we outline now. Since all positions in the substrate are occupied and there are no external forces on these atoms, then in equilibrium they must satisfy  $\forall \ell$  and  $\forall j < 1$

$$(8) \quad \begin{aligned} 0 = & k_L(u_{\ell+1,j} - 2u_{\ell,j} + u_{\ell-1,j}) \\ & + \frac{k_D}{2}(u_{\ell+1,j+1} + u_{\ell-1,j+1} + u_{\ell+1,j-1} + u_{\ell-1,j-1} - 4u_{\ell,j}) \\ & + \frac{k_D}{2}(v_{\ell+1,j+1} + v_{\ell-1,j-1} - v_{\ell+1,j-1} - v_{\ell-1,j+1}) \end{aligned}$$

$$(9) \quad \begin{aligned} 0 = & k_L(v_{\ell,j+1} - 2v_{\ell,j} + v_{\ell,j-1}) \\ & + \frac{k_D}{2}(v_{\ell+1,j+1} + v_{\ell-1,j+1} + v_{\ell+1,j-1} + v_{\ell-1,j-1} - 4v_{\ell,j}) \\ & + \frac{k_D}{2}(u_{\ell+1,j+1} + u_{\ell-1,j-1} - u_{\ell+1,j-1} - u_{\ell-1,j+1}). \end{aligned}$$

Let us now consider a Fourier expansion of the displacement in the  $x$ -direction. The generic Fourier mode will take the form

$$(10) \quad \begin{aligned} u_{\ell,j} = & \hat{u}_j(\xi)e^{i\ell\xi}, \\ v_{\ell,j} = & \hat{v}_j(\xi)e^{i\ell\xi}. \end{aligned}$$

By inserting this Fourier expansion in the expression of the surface force (6)–(7) one obtains

$$(11) \quad \begin{aligned} \hat{f}_1(\xi) = & k_D(\hat{u}_0 \cos \xi - \hat{u}_1 - i\hat{v}_0 \sin \xi), \\ \hat{g}_1(\xi) = & k_L(\hat{v}_0 - \hat{v}_1) + k_D(\hat{v}_0 \cos \xi - \hat{v}_1 - i\hat{u}_0 \sin \xi). \end{aligned}$$

On the other hand, substituting Eq. (10) into (8) and (9) one obtains

$$(12) \quad \begin{aligned} 0 = & 2k_L\hat{u}_j(\cos \xi - 1) + k_D[(\hat{u}_{j-1} + \hat{u}_{j+1}) \cos \xi - 2\hat{u}_j \\ & + i(\hat{v}_{j+1} - \hat{v}_{j-1}) \sin \xi], \\ 0 = & 2k_L(\hat{v}_{j+1} - 2\hat{v}_j + \hat{v}_{j-1}) + k_D[(\hat{v}_{j-1} + \hat{v}_{j+1}) \cos \xi - 2\hat{v}_j \\ & + i(\hat{u}_{j+1} - \hat{u}_{j-1}) \sin \xi]. \end{aligned}$$

In the relation above we have suppressed the dependence of the Fourier modes on  $\xi$ . The solution of (12) is a linear combination of terms of the form

$$\hat{u}_j = \alpha^j \hat{u}_0, \quad \hat{v}_j = \alpha^j \hat{v}_0.$$

Substituting these expression into (12), we obtain a homogeneous system that admits nontrivial solutions only if the determinant of the coefficient matrix is zero. The resulting equation for  $\alpha$  is a palindromic equation of degree 4 [8], with roots  $\alpha_1, \alpha_2, 1/\alpha_1, 1/\alpha_2$ , with  $|\alpha_1| > 1, |\alpha_2| > 1$  (except in some degenerate cases, in which one pair of roots goes to zero and infinity). Since we look for solutions that vanish as  $j \rightarrow -\infty$ , the solution to (12) takes the form

$$(13) \quad \begin{pmatrix} \hat{u}_j \\ \hat{v}_j \end{pmatrix} = Q(j)Q^{-1}(1) \begin{pmatrix} \hat{u}_1 \\ \hat{v}_1 \end{pmatrix},$$

where  $Q(j)$  is the invertible  $2 \times 2$  matrix given by

$$\begin{pmatrix} \vec{r}_1 \alpha_1^j & \vec{r}_2 \alpha_2^j \end{pmatrix}.$$

Finally,  $\vec{r}_p$  and  $\alpha_p$  are the ‘‘eigenvectors’’ and ‘‘eigenvalues’’ that arise when solving the discrete equation. The solution of (8) and (9) is a linear combination of Fourier modes of the form (10)

$$(14) \quad \vec{u}_{\ell j} = \sum_{\xi} Q(j)Q^{-1}(1) \widehat{\vec{u}}_1(\xi) e^{i\ell\xi}.$$

In order to proceed, we have to make some assumption on the computational domain. We assume that the system is periodic in the horizontal direction, with a period of  $M$  atoms. Then the coefficients of a linear combination of Fourier modes corresponding to a given configuration can be efficiently computed by the fast Fourier transform (FFT) [6]. To compute the forces  $(f_{\ell 1}, g_{\ell 1})$ , one first uses an FFT to determine  $(\hat{u}_1(\xi), \hat{v}_1(\xi))$  from  $\vec{u}_{\ell 1}$  and then uses (13) to compute  $(\hat{u}_0, \hat{v}_0)$ . These are then substituted into (11) to determine  $\hat{f}_1(\xi)$  and  $\hat{g}_1(\xi)$ , from which  $f_{\ell 1}$  and  $g_{\ell 1}$  are evaluated using an inverse discrete Fourier transform. For more detail see [18].

**3.3. Fictitious atoms and the connection strength.** In this section, we shall introduce fictitious atoms and the connection strength. These will greatly facilitate the implementation of the multigrid method. But first the following spring matrices are introduced:

$$\begin{aligned} K_{-1,-1}^x &= k_D/2, & K_{0,-1}^x &= 0, & K_{1,-1}^x &= k_D/2, \\ K_{-1,0}^x &= k_L, & K_{0,0}^x &= 0, & K_{1,0}^x &= k_L, \\ K_{-1,1}^x &= k_D/2, & K_{0,1}^x &= 0, & K_{1,1}^x &= k_D/2, \end{aligned}$$

and

$$\begin{aligned} K_{-1,-1}^y &= k_D/2, & K_{0,-1}^y &= k_L, & K_{1,-1}^y &= k_D/2, \\ K_{-1,0}^y &= 0, & K_{0,0}^y &= 0, & K_{1,0}^y &= 0, \\ K_{-1,1}^y &= k_D/2, & K_{0,1}^y &= k_L, & K_{1,1}^y &= k_D/2. \end{aligned}$$

Using these matrices we can rewrite (4) and (5) as

$$(15) \quad \sum_{m,n=-1}^1 K_{mn}^x \sigma_{\ell j; mn} ([u_{\ell+m, j+n} - u_{\ell j}] + mn[v_{\ell+m, j+n} - v_{\ell j}]) + f_{\ell j} = F_{\ell j},$$

$$(16) \quad \sum_{m,n=-1}^1 K_{mn}^y \sigma_{\ell j; mn} ([v_{\ell+m, j+n} - v_{\ell j}] + mn[u_{\ell+m, j+n} - u_{\ell j}]) + g_{\ell j} = G_{\ell j}.$$

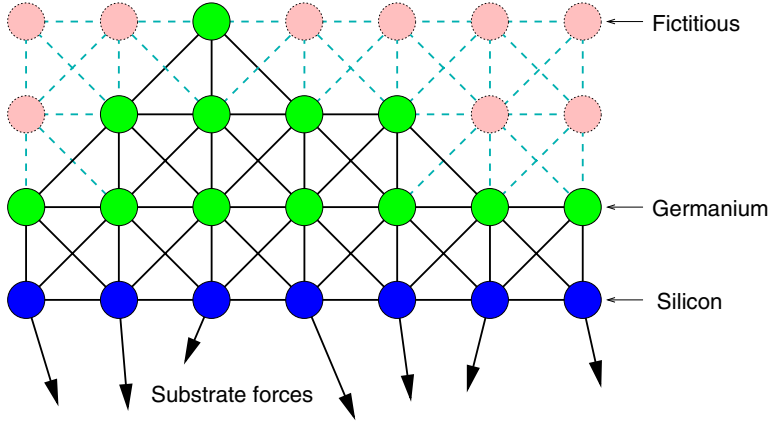


FIG. 2. Computational domain, with real and fictitious atoms. The bottom line of atoms represents the top layer of the silicon substrate. The null interaction with fictitious atoms is represented by dashed lines. The interaction of all the silicon atoms below the top layer are represented by forces.

For the next step we introduce fictitious atoms. This will be done as follows. Recall that the substrate is located at  $j = 1$  and suppose there is an arrangement of atoms where the highest atom is located at  $j_{max}$ ; let  $N$  be some integer such that  $N \geq j_{max}$ . Our computational domain will now be

$$\Omega = \{(i, j) : 1 \leq i \leq M, 1 \leq j \leq N\}.$$

All grid points in  $\Omega$  that are not actual atoms are called *fictitious atoms*. Equations (4) and (5) are extended to all fictitious atoms (see Figure 2) by assigning them a connectivity matrix all with zero entries.

A simple way to represent the connectivity matrix for all atoms (real and fictitious) is obtained as follows. We define a *site atomic density*  $p_{i,j}$  as follows:

$$p_{i,j} = \begin{cases} 1 & \text{if } (i, j) \text{ is a real atom,} \\ 0 & \text{if } (i, j) \text{ is a fictitious atom.} \end{cases}$$

With these definitions, the connectivity matrix can be written as

$$(17) \quad \sigma_{i,j;m,n} = p_{i,j} p_{i+m,j+n}.$$

It is convenient to introduce a parameter that represents the strength of the connection between one atom and its neighbors. Given a pair of atoms,  $(\ell, j)$  and  $(\ell + m, j + n)$ ,  $m, n \in \{-1, 0, 1\}$ , the connection strength, denoted by  $c_{\ell,j;mn}$ , is given by

$$c_{\ell,j;m,n} = \frac{1}{2} (\sigma_{\ell,j;m,n} + \sigma_{\ell+m,j+n;-m,-n}).$$

It should be pointed out that  $c_{\ell,j;m,n}$  is equal to  $\sigma_{\ell,j;m,n}$  on the fine grid; however, after  $\sigma_{\ell,j;m,n}$  is coarse-grained this is no longer true.

We can now write the equilibrium equations as

$$(18) \quad \sum_{m,n=-1}^1 c_{\ell,j;mn} K_{mn}^x (u_{\ell+m,j+n} - u_{\ell,j} + mn(v_{\ell+m,j+n} - v_{\ell,j})) + f_{\ell,j} = F_{\ell,j},$$

$$(19) \quad \sum_{m,n=-1}^1 c_{\ell,j;mn} K_{mn}^y (v_{\ell+m,j+n} - v_{\ell,j} + mn(u_{\ell+m,j+n} - u_{\ell,j})) + g_{\ell,j} = G_{\ell,j}.$$

In this way, the problem has been reduced to a discrete elliptic problem on a rectangular region. Equations (18) and (19) are identical to (15) and (16) since  $c_{\ell,j;m,n}$  is equal to  $\sigma_{\ell,j;m,n}$  on the fine grid. When the system is coarse-grained, then  $\sigma_{\ell,j;m,n}$ , defined on the atoms, is coarse-grained, while  $c_{\ell,j;m,n}$ , defined on the pair of bonds, is the number that is used to actually define the strength of the connectivity. Equations (18) and (19) will be used in the multigrid scheme formulated below.

**3.3.1. Boundary conditions.** We shall use periodic boundary conditions in the  $x$ -direction. Let us next consider the  $y$ -direction. Recall that there are no atoms for  $j > N$  and for  $j < 1$ , and this means we must choose the values  $\sigma_{\ell j;mn}$  to reflect this fact. This is done as follows. We have

$$(20) \quad \sigma_{\ell,1;m,-1} = \sigma_{\ell,N;m,1} = 0 \quad \text{for } \ell = 1 \text{ to } M \text{ and } m = -1 \text{ to } 1.$$

In addition we have

$$(21) \quad \sigma_{\ell,0;m,n} = \sigma_{\ell,N+1;m,n} = 0 \quad \text{for } \ell = 1 \text{ to } M \text{ and } m, n = -1 \text{ to } 1.$$

In view of (20) and (21) no boundary conditions for  $\vec{u}$  are needed at  $j = 1$  or  $j = N$ .

**3.3.2. Compact form.** Since  $f_{\ell 1}$  and  $g_{\ell 1}$  are completely determined by  $\vec{u}_{1,1}, \dots, \vec{u}_{M,1}$ , we may write (18) and (19) in the compact form

$$(22) \quad \mathbf{A}\mathbf{u} = \mathbf{F},$$

where  $\mathbf{u} = (\vec{u}_{1,1}, \dots, \vec{u}_{M,N})^T$  and  $\mathbf{F} = (\vec{F}_{1,1}, \dots, \vec{F}_{M,N})^T$ .

System (22) can be written as

$$(23) \quad \begin{pmatrix} S & B \\ B^T & A \end{pmatrix} \begin{pmatrix} u_S \\ u_G \end{pmatrix} = \begin{pmatrix} F_S \\ F_G \end{pmatrix},$$

where  $A$  and  $B$  are sparse matrices describing the Ge-Ge and Si-Ge interactions, while  $S$  is a block circulant matrix describing the Si-Si interaction. The blocks in  $S$  are  $2 \times 2$  in two dimensions and  $3 \times 3$  in three dimensions. Matrix  $S$  can be separated into two terms,  $S = S_s + S_b$ , the first term denoting the direct interaction between silicon atoms at  $j = 1$ , while  $S_b$  is responsible of the interaction due to the rest of the substrate (i.e., with atoms with  $j < 1$ ). The term  $S_b u_S$  is given by the force  $(f_{\ell 1}, g_{\ell 1})$  (equations (18) and (19)). In performing the SOR relaxation, we write, for each point,  $(\ell, j) \in \Omega$ , a  $2 \times 2$  system, for the unknowns  $(u_{\ell j}, v_{\ell j})$ , where we write the terms that do not depend on  $(u_{\ell j}, v_{\ell j})$  on the right-hand side. The  $2 \times 2$  matrices that appear in each system contain the  $2 \times 2$  diagonal block of matrices  $A$  and  $S_s$  but not the diagonal terms of matrix  $S_b$ , which are contained on the right-hand side. As we shall see, this has to be taken into account when performing the method of SOR.

**4. Multigrid-Fourier algorithm.** The goal of our multigrid-Fourier method is to solve the equation  $\mathbf{A}\mathbf{u} = \mathbf{F}$  (equation (22)). For an excellent introduction on multigrid methods see Briggs [5]. A more advanced treatment of multigrid techniques can be found in [3], [7], or [19], for example.

Multigrid methods have three key ingredients, namely, a relaxation method, operators for coarse-graining and prolongation, and a coarse-grained version of system of equations. In this section, we will discuss these in relation to our problem.

**4.1. Relaxation scheme.** We write (18) and (19)  $\forall j = 1, \dots, N$  and  $\ell = 1, \dots, M$  in the form

$$(24) \quad \begin{pmatrix} a_{xx} & a_{xy} \\ a_{yx} & a_{xx} \end{pmatrix} \begin{pmatrix} u_{\ell j} \\ v_{\ell j} \end{pmatrix} = \begin{pmatrix} c_x \\ c_y \end{pmatrix} + \begin{pmatrix} f_{\ell j} \\ g_{\ell j} \end{pmatrix} + \begin{pmatrix} F_{\ell j} \\ G_{\ell j} \end{pmatrix},$$

where

$$\begin{aligned} a_{xx} &= \sum_{m,n=-1}^1 c_{\ell j;mn} K_{mn}^x, & a_{xy} &= \sum_{m,n=-1}^1 c_{\ell j;mn} K_{mn}^x mn, \\ a_{yx} &= \sum_{m,n=-1}^1 c_{\ell j;mn} K_{mn}^y mn, & a_{yy} &= \sum_{m,n=-1}^1 c_{\ell j;mn} K_{mn}^y, \\ c_x &= \sum_{m,n=-1}^1 c_{\ell j;mn} K_{mn}^x (u_{\ell+m,j+n} + mnv_{\ell+m,j+n}), \end{aligned}$$

and

$$c_y = \sum_{m,n=-1}^1 c_{\ell j;mn} K_{mn}^y (v_{\ell+m,j+n} + mn u_{\ell+m,j+n}).$$

and we omit writing the explicit dependence of the  $a$  and  $c$  coefficients on  $\ell$  and  $j$ .

Note that in view of the observation at the end of the previous section, the force  $(f_{\ell 1}, g_{\ell 1})$  contains a term proportional to  $(u_{\ell 1}, v_{\ell 1})$ . It is necessary to bring such diagonal term on the left-hand side to implement any iteration scheme. This is accomplished by adding to the  $2 \times 2$  matrix appearing in (24) the diagonal part of the diagonal block of matrix  $S_b$  denoted by  $\text{diag}(\beta^x, \beta^y)$ . Of course this must be compensated for by modifying the right-hand side and we rewrite (24) as

$$(25) \quad \begin{pmatrix} a_{xx} + \beta_j^x & a_{xy} \\ a_{yx} & a_{yy} + \beta_j^y \end{pmatrix} \begin{pmatrix} u_{\ell j} \\ v_{\ell j} \end{pmatrix} = \begin{pmatrix} c_x \\ c_y \end{pmatrix} + \begin{pmatrix} f_{\ell j} \\ g_{\ell j} \end{pmatrix} + \begin{pmatrix} F_{\ell j} \\ G_{\ell j} \end{pmatrix} + \begin{pmatrix} \beta_j^x u_{\ell j} \\ \beta_j^y v_{\ell j} \end{pmatrix}.$$

The relaxation scheme for the above system is based on treating the right-hand side as known. In more detail, let us denote by  $(u_{\ell j}^k, v_{\ell j}^k)$  the value of the displacement at iteration  $k$ . Then the solution at the next iteration,  $(u_{\ell j}^{k+1}, v_{\ell j}^{k+1})$ , is computed by relaxing system (25) by one SOR iteration. More precisely, the SOR relaxation is performed as

$$(26) \quad u_{\ell,j}^* = (\beta_j^x u_{\ell j}^k + c_x^k + f_{\ell j}^k + F_{\ell j} - a_{xy} v_{\ell j}^k) / (a_{xx} + \beta_j^x),$$

$$u_{\ell j}^{k+1} = \omega u_{\ell,j}^* + (1 - \omega) u_{\ell j}^k,$$

$$(27) \quad v_{\ell,j}^* = (\beta_j^y v_{\ell j}^k + c_y^k + g_{\ell j}^k + G_{\ell j} - a_{yx} u_{\ell j}^{k+1}) / (a_{yy} + \beta_j^y),$$

$$v_{\ell j}^{k+1} = \omega v_{\ell,j}^* + (1 - \omega) v_{\ell j}^k,$$

and  $\beta_j^{x,y} = \beta^{x,y} \delta_{j,1} + \text{tol}(1 - \delta_{j,1})$ . The small number  $\text{tol}$  is added to avoid division by zero in expressions (26) and (27). In our computation we use  $\text{tol} = 10^{-16}$ .

*Remarks.* The superscript  $k$  on  $c_x$ ,  $c_y$ ,  $f_{\ell j}$ , and  $g_{\ell j}$  indicates that these terms should be evaluated using  $(u_{\ell j}^k, v_{\ell j}^k)$ . As is clear from their definition, the parameters  $\beta^x$  and  $\beta^y$  depend on the level  $L$ .

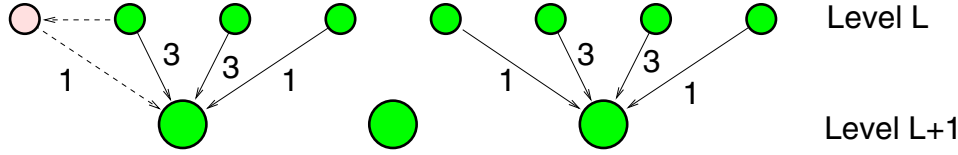


FIG. 3. Coarsening operation near the left boundary and in the interior. The number represent the relative weights that are used in the weighting from fine to coarse. The same relative weights are used in the prolongation, when interpolating values from the coarse to the fine grid. The dashed arrows are a schematic representation on how to compute the weights near the boundary.

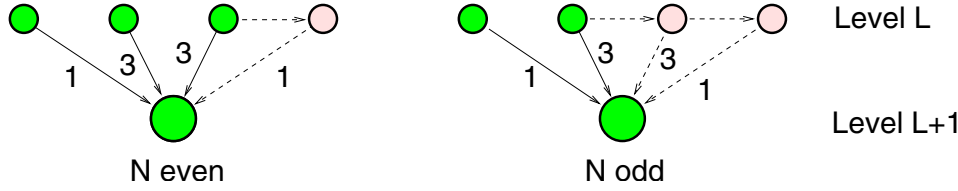


FIG. 4. Coarse from fine operator between levels  $L$  and  $L + 1$  near the right boundary, in the case of even (left) and odd (right) values of  $N_L$ .

**4.2. Coarsening and prolongation.** In this section, we describe the relationship between the fine mesh and the coarse mesh and the operators that take variables from the fine mesh to the coarse mesh and back. We shall do this direction by direction; the two-dimensional coarse-grain operator will be given by the Cartesian product of one-dimensional operators.

**4.2.1. Operators in the  $y$ -direction.** In the  $y$ -direction the number of mesh points on the finest level is  $N$ . The number of mesh points at level  $L$  is denoted as  $N_L$  with  $N_1 = N$ . Then the number of grid points on subsequent coarse meshes is given by

$$(28) \quad N_L = \lfloor (N_{L-1} + 1)/2 \rfloor.$$

Here  $\lfloor \cdot \rfloor$  denotes integer part. We shall use a cell-centered stencil of the domain (see Figure 3).

The coarsening operator is now described. We consider a quantity  $\{q_j^L, j = 1, \dots, N_L\}$ , which is defined on grid level  $L$ , and we wish to restrict this variable to the next grid level  $L + 1$ . On the next grid level this quantity is denoted  $\{q_j^{L+1}, j = 1, \dots, N_{L+1}\}$ . Clearly  $N_{L+1} \leq N_L$ . The variables on the level  $L + 1$  are determined by averaging the values from level  $L$  as follows:

$$q_1^{L+1} = \frac{1}{8}(4q_1^L + 3q_2^L + q_3^L),$$

$$q_j^{L+1} = \frac{1}{8}(q_{2j-2}^L + 3q_{2j-1}^L + 3q_{2j}^L + q_{2j+1}^L) \quad \text{for } j = 2, \dots, N_{L+1} - 1,$$

and

$$q_{N_{L+1}}^{L+1} = \begin{cases} \frac{1}{8}(q_{2N_L-2}^L + 3q_{2N_L-1}^L + 4q_{2N_L}^L) & \text{if } N_L \text{ is even,} \\ \frac{1}{8}(q_{2N_L-1}^L + 7q_{2N_L}^L) & \text{if } N_L \text{ is odd.} \end{cases}$$

See Figure 4. We note that the above equations are valid only if  $N_L \geq 3$ . A less



A cell-centered coarsening is used, as in the case of the  $y$ -direction. In view of the above facts the operator that take variables from level  $L$  to level  $L + 1$  can be phrased in the same fashion as (29) except  $C$  contains terms on the upper right and lower left parts, to account for periodicity. A convenient way to implement periodic boundary conditions is by the use of an index array and is explained in the appendix.

**4.2.3. Operators in two dimensions.** The extension to two dimensions is achieved by taking the Cartesian product of the operators in the  $x$ - and  $y$ -directions. These operators are best expressed in terms of an algorithm rather than as a matrix; the algorithms are given in the appendix.

Let  $\mathbf{q}^L = (q_{11}^L, \dots, q_{M_L, N_L}^L)^T$ . Algorithm 1 provides the coarse from fine operator, which will be written as

$$\mathbf{q}^{L+1} = \text{CF}(\mathbf{q}^L),$$

and Algorithm 2 provides the fine from coarse operator, which we write as

$$\mathbf{q}^L = \text{FC}(\mathbf{q}^{L+1}).$$

**4.3. Coarse-grained elastic equations.** It is now straightforward to coarse-grain the discrete elastic equations. We denote by  $N_{gl}$  the number of grid levels. We first coarse-grain  $\sigma_{\ell j; mn}$  using the fine to coarse operator

$$\sigma_{\ell j; mn}^{L+1} = \text{CF}(\sigma_{\ell j; mn}^L), \quad L = 1, \dots, N_{gl} - 1,$$

and so

$$c_{\ell j; mn}^L = \frac{1}{2} (\sigma_{\ell j; mn}^L + \sigma_{\ell+m, j+n; -m, -n}^L), \quad L = 1, \dots, N_{gl}.$$

It is important to apply the boundary conditions on  $\sigma$  given by (20) and (21) after the coarse-graining procedure. The substrate force term is coarse-grained as follows. First, we note that

$$f_{\ell j}^L = g_{\ell j}^L = 0 \quad \text{for } j \geq 2.$$

To compute  $\vec{f}_{\ell 1}^L$  one uses  $\vec{u}_{\ell 1}^L$  and  $M_L$  in place of  $\vec{u}_{\ell 1}$  and  $M$  in (13) to compute  $\vec{u}_{\ell, 0}^L$ , which is then used in (11) to deduce  $(\hat{f}_1^L, \hat{g}_1^L)^T$ . The coarse-grained substrate force is then

$$\begin{pmatrix} f_{\ell 1}^L \\ g_{\ell 1}^L \end{pmatrix} = 2^{2-2L} \mathcal{F}_{M_L}^{-1} \begin{pmatrix} \hat{f}_1^L \\ \hat{g}_1^L \end{pmatrix},$$

where  $\mathcal{F}_M^{-1}$  is the inverse discrete Fourier transform using  $M$  modes.

It is now easy to write down a coarse-grained version of  $\mathbf{A}\mathbf{u}$ . We have

$$\mathbf{A}^L \mathbf{u}_{\ell j}^L \equiv \begin{pmatrix} 2^{2-2L} \sum_{m,n=-1}^1 c_{\ell j; mn}^L K_{mn}^x (u_{\ell+m, j+n}^L - u_{\ell j}^L + mn(v_{\ell+m, j+n}^L - v_{\ell j}^L)) + f_{\ell j}^L \\ 2^{2-2L} \sum_{m,n=-1}^1 c_{\ell j; mn}^L K_{mn}^y (v_{\ell+m, j+n}^L - v_{\ell j}^L + mn(u_{\ell+m, j+n}^L - u_{\ell j}^L)) + g_{\ell j}^L \end{pmatrix}.$$

The scaling factor  $2^{2-2L}$  appearing above is typical of elliptic equations. Physically, this maybe interpreted as the springs becoming weaker on coarser levels.

In the implementation of our multigrid-Fourier algorithm we must relax the following system:

$$\mathbf{A}^L \mathbf{u}_{\ell_j}^L = \mathbf{F}^L.$$

This is accomplished using the relaxation scheme proposed in section 4.1.

**4.4. V-cycle implementation.** In our computations we implement the multigrid algorithm using a standard V-cycle which, for sake of completeness, we will now describe. The V-cycle starts with the following steps:

**Precomputation**

1. Compute  $\mathbf{F}^1$  using (3).
2. Compute  $N_L$  and  $M_L$  for  $L = 2$  to  $N_{gl}$  using (28) and (31).
3. Coarse-grain the connectivity matrix for  $L = 2$  to  $N_{gl}$
4. Initialize first guess for  $\mathbf{u}^1$  (usually  $\mathbf{u}^1 = 0$ ).

**V-cycle**

For  $L = 1$  to  $N_{gl} - 1$  do the following  
 Relax  $\mathbf{A}^L \mathbf{u}^L = \mathbf{F}^L$  for  $\eta$  steps  
 Compute residual  $\mathbf{r}^L = \mathbf{F}^L - \mathbf{A}^L \mathbf{u}^L$   
 Coarse grain the residual:  $\mathbf{r}^{L+1} = \text{CF}(\mathbf{r}^L)$   
 Set  $\mathbf{F}^{L+1} = \mathbf{r}^{L+1}$   
 Set  $\mathbf{u}^{L+1} = 0$  (This is the initial condition for the next relaxation)

End Loop

Solve  $\mathbf{A}^{N_{gl}} \mathbf{u}^{N_{gl}} = \mathbf{F}^{N_{gl}}$  by relaxation

For  $L = N_{gl} - 1$  to 1 do the following  
 Prolong the solution from the coarse mesh:  $\mathbf{v}^L = \text{FC}(\mathbf{u}^{L+1})$   
 Let  $\mathbf{w}^L = \mathbf{v}^L + \mathbf{u}^L$   
 Relax  $\mathbf{A}^L \mathbf{u}^L = \mathbf{F}^L$  with initial guess  $\mathbf{w}^L$  for  $\eta$  steps

End Loop

**4.5. Extension to three dimensions.** The procedure outlined above can be straightforwardly extended to three dimensions. The computational domain will consist of a set  $\Omega = \{(i, j, k) \mid 1 \leq i \leq M, 1 \leq j \leq M, 1 \leq k \leq N\}$ , with periodic boundary conditions on the  $x$  and  $y$  direction. The site atomic density  $p_{\ell,j,k}$  is defined as

$$p_{\ell,j,k} = \begin{cases} 1 & \text{if } (\ell, j, k) \text{ is a real atom,} \\ 0 & \text{if } (\ell, j, k) \text{ is a fictitious atom,} \end{cases}$$

and the connectivity matrix, which now has 27 components, 18 of which may be nonzero, is defined as

$$(32) \quad \sigma_{\ell,j,k;m,n,q} = p_{\ell,j,k} p_{\ell+m,j+n,k+q}, \quad (m, n, q) \in \{-1, 0, 1\}^3.$$

The interaction between silicon atoms of the substrate is also implemented through the FFT. The details of how this is performed are described in [18].

The spring matrix  $K_{\ell,j,k}^{(s)}$ ,  $s = 1, 2, 3$ , where 1, 2, 3 denote, respectively,  $x$ ,  $y$ , and  $z$ , can be compactly written as

$$K_{\ell,j,k}^{(1)} = K_{k,\ell,j}^{(2)} = K_{j,k,\ell}^{(3)} = k_L \ell^2 \delta_{j0} \delta_{k0} + \frac{k_D}{2} \ell^2 (j^2 \delta_{k0} + k^2 \delta_{j0}).$$

The formula for the forces, equivalent to formulas (15) (16) in two dimensions, can be expressed as

$$F_{\ell_1, \ell_2, \ell_3}^{(s)} = \sum_{m_1, m_2, m_3 = -1}^1 m_s K_{m_1, m_2, m_3}^{(s)} \sigma_{\ell_1, \ell_2, \ell_3; m_1, m_2, m_3} \Delta u_{\ell_1, \ell_2, \ell_3; m_1, m_2, m_3}$$

where

$$\Delta u_{\ell_1, \ell_2, \ell_3; m_1, m_2, m_3} = \sum_{r=1}^3 (u_{\ell_1+m_1, \ell_2+m_2, \ell_3+m_3}^{(r)} - u_{\ell_1, \ell_2, \ell_3}^{(r)}) m_r$$

and  $u^{(r)}$ ,  $r = 1, \dots, 3$  denote the three space components of vector  $\vec{u}$ .

**4.6. An alternative approach.** The method described above is based on coarse-graining the connectivity matrix. This method gives good results in both two and three dimensions. However, it has the drawback of requiring a lot of memory, since the matrix  $\sigma$  has to be stored at all levels (except at the first level, since in this case it may be computed from a boolean vector, using (17)). An alternative scheme is obtained by coarse-graining the site atomic density  $p$  on the various grids. The strength of the connection  $c_{\ell_j; m, n}$  can be directly computed as the geometric mean of the pair of site atomic densities,

$$(33) \quad c_{\ell_j; m, n} = \sqrt{p_{\ell_j}} \sqrt{p_{\ell+m, j+n}},$$

and a similar formula can be used in three dimensions. For computational efficiency, it is better to store  $\sqrt{p_{\ell_j}}$  rather than  $p_{\ell_j}$ , in order to save floating point operations during the cycle. In three-dimensional calculations, this approach gives results which are very similar in terms of convergence rate, with respect to the ones where the connectivity matrix was coarse-grained, and it is therefore generally preferable, since its implementation requires less computer memory. However, it does not provide the same convergence rate for two-dimensional computations.

**5. Results.** In this section we present some numerical results comparing the computational speed of our multigrid-Fourier method to the conjugate gradient method and minimum residual method. In no case is any preconditioning performed. In this regard, we mention that we implemented incomplete Cholesky preconditioning for the conjugate gradient method and found that in most cases it reduced the convergence rate.

In the computations presented here we take  $k_L = 1$ ,  $k_D = 1/2$ , and  $\epsilon = 0.04$ . For a test problem we consider several mounds on a substrate. In all our tests we used two relaxation steps at each level ( $\eta = 2$ ) and a relaxation parameter  $\omega = 1.2$ . In the two-dimensional computations we found that multiplying the underrelaxation parameter  $\beta$  by 1.1 (see section 4.1) improved the rate of convergence. All the computations were performed on a 750 MHz Sun UltraSPARC processor. The algorithm was coded in FORTRAN 77 and compiled with the `-fast` option.

**5.1. Two dimensions.** We consider a configuration in which the location of the atoms is given by

$$\{(i, j) | 1 \leq j \leq J(i), \quad 1 \leq i \leq M\},$$

where

$$(34) \quad J(i) = \max(\Phi(i), 1),$$

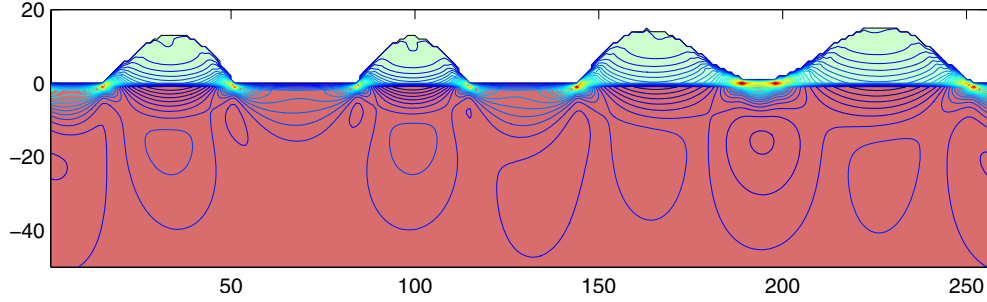


FIG. 5. Energy density for  $N = 20$ ,  $M = 256$ . The height profile is given by (34) with  $D = 64$  and  $H = 20$ . The dark region below denotes the silicon substrate. Normally we do not compute in this region but in this case we have for illustrative purposes.

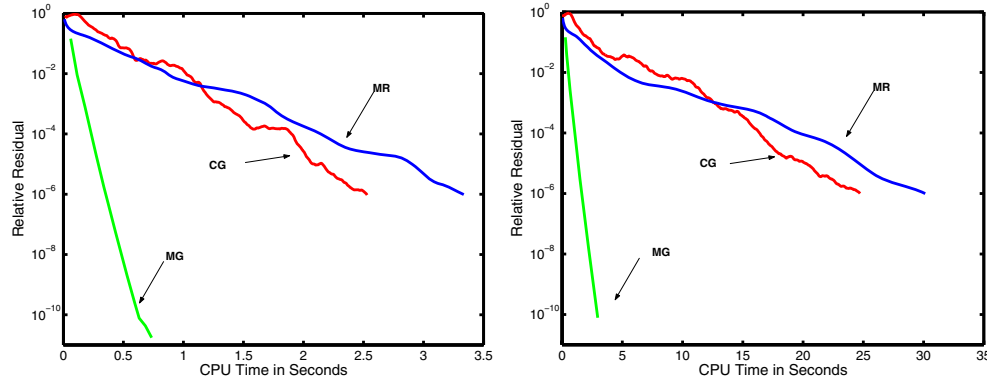


FIG. 6. Plots of relative 2-norm of the residual as a function of the CPU time in two dimensions. In both cases the interface profile was given by (34) with  $H = 20$  and  $D = 256$ . On the left  $M = 512$ , and on the right  $M = 2048$ .  $N = 20$  in both cases. CG = conjugate gradient, MR = minimum residual, and MG = our multigrid-Fourier method.

with

$$(35) \quad \Phi(i) = H - \frac{H}{2} \left[ 1 + \sin^2 \left( \frac{\pi(i + 43)}{M} \right) \right] \left[ 1 + \frac{3}{4} \cos \left( \frac{2\pi(i - 67)}{D} \right) \right].$$

In Figure 5 we plot the contour of the elastic energy density stored in the crystal. For the definition of the elastic energy density, see [18]. It appears that the energy density is highly concentrated in regions of negative curvature, while it is very small in regions of positive curvature. Also, the figure shows that the deviation of the energy density from its mean value in the bulk silicon extends well below the surface.

Several runs are performed, keeping the number  $D$  fixed and increasing the size of the problem by varying  $M$ , while  $N$  remains unchanged. Each test problem is solved with a conjugate gradient method and with a minimal residual method [11].

The results of the comparison are summarized in Figure 6, and in Tables 1 through 3. As it appears from the figure and from the tables, the number of V-cycles required to reach a given tolerance in the relative norm of the residual is basically independent of the size of the problem, while this is not the case for the CG and the MR methods.

TABLE 1

The CPU time in seconds required to reduce the relative residual to a prescribed level in two dimensions for  $N = 20$ ,  $M = 512$ . The interface profile is given by (34) with  $D = 256$ ,  $H = 20$ .  $E$  is the relative 2-norm of the residual,  $T$  is the CPU time in seconds,  $N_I$  is the number of iterations,  $N_V$  is the number of V-cycles.

Timing Comparisons - Two Dimensions			
20 × 512			
scheme	$E = 9.11 \times 10^{-3}$	$E = 1.45 \times 10^{-4}$	$E = 3.26 \times 10^{-6}$
CG	$T = 0.67, N_I = 192$	$T = 1.19, N_I = 330$	$T = 1.51, N_I = 417$
MR	$T = 0.51, N_I = 109$	$T = 1.25, N_I = 257$	$T = 1.85, N_I = 383$
MG	$T = 0.09, N_V = 2$	$T = 0.17, N_V = 4$	$T = 0.25, N_V = 6$

TABLE 2

Algorithm performance. Everything as in Table 1 except  $M = 1024$ .

Timing Comparisons - Two Dimensions			
20 × 1024			
scheme	$E = 8.83 \times 10^{-3}$	$E = 1.46 \times 10^{-4}$	$E = 2.94 \times 10^{-6}$
CG	$T = 1.74, N_I = 226$	$T = 3.16, N_I = 411$	$T = 4.54, N_I = 587$
MR	$T = 1.27, N_I = 117$	$T = 3.92, N_I = 364$	$T = 5.75, N_I = 530$
MG	$T = 0.19, N_V = 2$	$T = 0.36, N_V = 4$	$T = 0.54, N_V = 6$

TABLE 3

Algorithm Performance. Everything as in Table 1 except  $M = 2048$ .

Timing Comparisons - Two Dimensions			
20 × 2048			
scheme	$E = 8.91 \times 10^{-3}$	$E = 1.48 \times 10^{-4}$	$E = 3.02 \times 10^{-6}$
CG	$T = 4.72, N_I = 256$	$T = 9.66, N_I = 523$	$T = 13.5, N_I = 733$
MR	$T = 2.87, N_I = 115$	$T = 10.7, N_I = 430$	$T = 15.2, N_I = 610$
MG	$T = 0.42, N_V = 2$	$T = 0.80, N_V = 4$	$T = 1.18, N_V = 6$

The multigrid-Fourier method is about six times faster than CG for  $M = 512$  and about 11 times faster for  $M = 2048$ .

**5.2. Three dimensions.** For the three-dimensional case, the test problem is defined by the location of atoms,

$$\{(i, j, k) \mid 0 \leq k \leq K(i, j), \quad 1 \leq i \leq M, \quad 1 \leq j \leq M\},$$

where

$$K(i, j) = \max(\Phi(i, j), 1)$$

with

$$(36) \quad \Phi(i, j) = H - \frac{H}{2} \cdot \left[ 1 + \frac{3}{2} \sin^2 \left( \frac{\pi(i + 43)}{M} \right) + \frac{3}{2} \sin^2 \left( \frac{\pi(j + 74)}{M} \right) \right] \\ \cdot \left[ \frac{3}{2} + \frac{3}{4} \cos \left( \frac{2\pi(i - 67)}{D} \right) + \frac{1}{4} \cos \left( \frac{2\pi(j - 39)}{D} \right) \right].$$

The test problem is solved for various values of  $M$ , by using both the multigrid-Fourier method and the conjugate gradient method. The results, summarized in

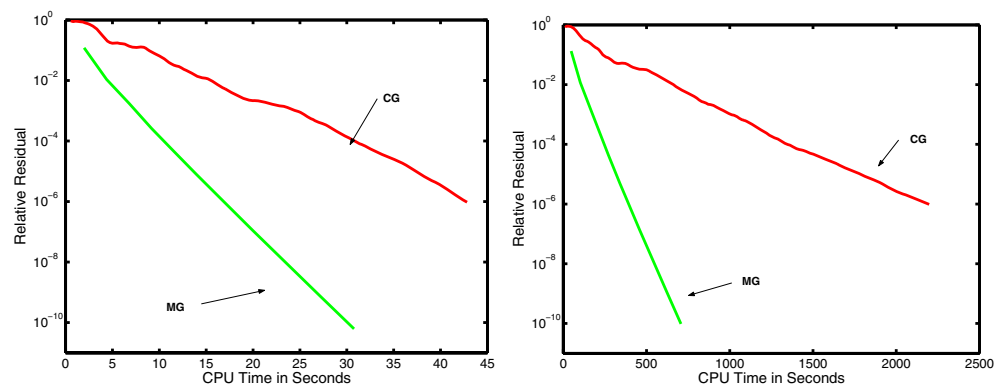


FIG. 7. Plots of relative 2-norm of the residual as a function of the CPU time in three dimensions. In both cases the interface profile is given by (36) with  $H = 34$ ,  $D = 128$ . On the left  $M = 128$  and on the right  $M = 512$ . In both figures,  $N = 16$ .

TABLE 4

The CPU time in seconds required to reduce the relative residual to a prescribed level in three dimensions for  $N = 16$ ,  $M = 128$ . The interface profile is given by (34) with  $D = 128$ ,  $H = 34$ .  $E$  is the 2-norm of the relative residual,  $T$  is the CPU time in seconds,  $N_I$  is the number of iterations,  $N_V$  is the number of V-cycles.

Timing Comparisons - Three Dimensions			
16 × 128 × 128			
scheme	$E = 1.08 \times 10^{-2}$	$E = 2.61 \times 10^{-4}$	$E = 1.42 \times 10^{-6}$
CG	$T = 15.5$ , $N_I = 58$	$T = 28.5$ , $N_I = 108$	$T = 36.6$ , $N_I = 139$
MG	$T = 4.37$ , $N_V = 2$	$T = 9.16$ , $N_V = 4$	$T = 14.0$ , $N_V = 7$

TABLE 5  
As Table 4 except  $M = 256$ .

Timing Comparisons - Three Dimensions			
16 × 256 × 256			
scheme	$E = 1.17 \times 10^{-2}$	$E = 3.01 \times 10^{-4}$	$E = 1.74 \times 10^{-6}$
CG	$T = 117.$ , $N_I = 94$	$T = 217.$ , $N_I = 176$	$T = 361.$ , $N_I = 293$
MG	$T = 20.5$ , $N_V = 2$	$T = 42.6$ , $N_V = 4$	$T = 75.8$ , $N_V = 7$

TABLE 6  
As Table 4 except  $M = 512$ .

Timing Comparisons - Three Dimensions			
16 × 512 × 512			
scheme	$E = 1.17 \times 10^{-2}$	$E = 2.86 \times 10^{-4}$	$E = 1.62 \times 10^{-6}$
CG	$T = 638.7$ , $N_I = 112$	$T = 1198.0$ , $N_I = 211$	$T = 2096.$ , $N_I = 372$
MG	$T = 102.3$ , $N_V = 2$	$T = 212.3$ , $N_V = 4$	$T = 377.1$ , $N_V = 7$

Figure 7 and in Tables 4 through 6, show that the multigrid-Fourier method is almost six times faster than CG in the case  $M = 512$ , a speedup that is similar to the analogous case in two dimensions.

**6. Summary.** In this paper, we have formulated a multigrid-Fourier method for the solution of the large linear system for the displacement field that results from a model of heteroepitaxy. In our algorithm the displacements of the atoms in the deposited film and the top layer of substrate atoms are isolated from the rest of the substrate. The latter is replaced by equivalent forces which can be efficiently computed using the fast Fourier transform, following the approach developed in [18]. The equations of the displacement field can be cast as unknowns on a rectangular Cartesian grid by using fictitious atoms and a connectivity matrix. Written in this form, they readily coarse-grain using a cell centered stencil. Since the substrate forces are evaluated using a Fourier approach, they can be easily coarse-grained. The equations are relaxed at each level using the method of successive over relaxations; however, the substrate forces must be under relaxed.

The multigrid-Fourier method has been implemented using a standard V-cycle in both two and three dimensions. We observe that the number of V-cycles needed to achieve a specified error (the relative 2-norm of the residual) is practically independent of the size of the domain. A comparison shows that for relatively small domains our multigrid-Fourier is 2 to 3 times faster than the method of conjugate gradients, whereas for larger domains it can be 6 to 10 times faster.

Finally we mention that the multigrid formulation offers another advantage when it is used in conjunction with the KMC method to simulate heteroepitaxy. It allows one to compute a local correction to the displacement field after each hop. Such a local correction would provide enough accuracy to estimate the hopping rate. Of course, after several hops a complete calculation of the displacement field would be performed.

**Appendix.** In this section we provide algorithms for the implementation of the coarse from fine and the fine from coarse operators.

We consider  $qf(i, j)$  on a fine mesh  $mf \times nf$  and we wish to find coarse-grain variables,  $qc(i, j)$ , on the coarse mesh  $mc \times nc$ . We first give the following definitions:

$$c(-2)=1 \quad c(-1)=3, \quad c(0)=3, \quad c(1)=1,$$

$$p(i) = \begin{cases} mf & \text{if } i=0 \\ i & \text{if } 1 \leq i \leq mf \\ 1 & \text{if } i=mf+1, \end{cases}$$

and

$$t(j) = \begin{cases} 1 & \text{if } j=0 \\ j & \text{if } 1 \leq j \leq 2nc \\ 2nc & \text{if } i=2nc+1 \end{cases}$$

**The coarse from fine algorithm.**

```
do j=1, nc
  do i=1, mc
    do i'=-2,1
      do j'=-2,1
        qc(i,j) = qc(i,j) + c(i')c(j')qf(p(2i+i'),t(2j+j'))/64
      enddo
    enddo
  enddo
enddo
```

**The fine from coarse algorithm.**

```

do j=1, nc
  do i=1, mc
    do i'=-2,1
      do j'=-2,1
        qf(p(2i+i'),t(2j+j')) = qf(p(2i+i'),t(2j+j')) +
                                c(i')c(j')qc(i,j)/16
      enddo
    enddo
  enddo
enddo

```

## REFERENCES

- [1] M. BIEHL, M. AHR, W. KINZEL, AND F. MUCH, *Kinetic Monte Carlo simulations of heteroepitaxial growth*, Thin Solid Films, 428 (2003), pp. 52–55.
- [2] M. A. BRANDT, *Multi-Level Adaptive Technique (MLAT) for Fast Numerical Solutions to Boundary Value Problems*, in Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics, H. Cabannes and R. Temam, eds., Lecture Notes in Phys. 18, Springer, New York, 1973, pp. 82–89.
- [3] M. A. BRANDT, *Multi-level adaptive solutions in boundary value problems*, Math. Comput., 31 (1977), pp. 333–390.
- [4] M. A. BRANDT, *Multiscale Scientific Computation: Review 2000*, <http://www.wisdom.weizmann.ac.il/~achi/>.
- [5] M. W. L. BRIGGS, *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.
- [6] M. FRIGO AND S. G. JOHNSON, *FTTW*, <http://www.ftw.org/>.
- [7] M. W. HACKBUSCH, *Multi-Grid Methods and Applications*, Springer, Berlin, 1985.
- [8] M. A. HILLIGES, C. MEHL, AND V. MEHRMANN, *On the Solution of Palindromic Eigenvalue Problems*, Proceedings of the European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS), Jyväskylä, Finland, 2004.
- [9] M. M. T. LUNG, C.-H. LAM, AND L. M. SANDER, *Island, pit and groove formation in strained heteroepitaxy*, Phys. Rev. Lett., 95 (2005), pp. 1–4.
- [10] M. C. H. LAM, C. K. LEE, AND L. M. SANDER, *Competing roughening mechanisms in strained heteroepitaxy: A fast kinetic Monte Carlo study*, Phys. Rev. Lett., 89 (2002), pp. 1–4.
- [11] C. C. PAIGE AND M. A. SAUNDERS, *MINRES: Sparse Symmetric Equations*, 2003. The software package is available at the System Optimization Laboratory of Stanford University, <http://www.stanford.edu/group/SOL/software/minres/html>.
- [12] M. F. MUCH, M. AHR, M. BIEHL, AND W. KINZEL, *A kinetic Monte Carlo method for the simulation of heteroepitaxial growth*, Comput. Phys. Commun., 147 (2002), pp. 226–229.
- [13] M. F. MUCH AND M. BIEHL, *Simulation of wetting-layer and island formation in heteroepitaxial growth*, Europhys. Lett., 63 (2003), pp. 14–20.
- [14] M. B. G. ORR, D. A. KESSLER, C. W. SNYDER, AND L. M. SANDER, *A model for strain-induced roughening and coherent island growth*, Europhys. Lett., 19 (1992), pp. 33–38.
- [15] M. C. RATSCH, A. ZANGWILL, AND P. ŠMILAUER, *Scaling of heteroepitaxial island sizes*, Surface Science Letters, 314 (1994), pp. L937–L942.
- [16] M. C. RATSCH AND A. ZANGWILL, *Equilibrium theory of the Stranski-Krastanov epitaxial morphology*, Surface Science, 293 (1993), pp. 123–131.
- [17] M. C. RATSCH, P. ŠMILAUER, D. D. VVEDENSKY, AND A. ZANGWILL, *Mechanism for Coherent Island Formation during Heteroepitaxy*, J. Phys. I France, 6 (1996), pp. 575–581.
- [18] M. G. RUSSO AND P. SMEREKA, *Computation of Strained Epitaxial Growth in Three Dimensions by Kinetic Monte Carlo*, J. Comput. Physics, accepted.
- [19] M. U. TROTTENBERG, C. W. OOSTERLEE, AND A. SCHÜLLER, *Multigrid*, Academic Press, New York, 2001.