Available online at www.sciencedirect.com

SCIENCE DIRECT®

Comput. Methods Appl. Mech. Engrg. 193 (2004) 1579–1601

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

Molecular dynamics boundary conditions for regular crystal lattices

Gregory J. Wagner [a], Eduard G. Karpov [b], Wing Kam Liu [c,*]

[a] Sandia National Laboratories, Livermore, CA 94551 USA
[b] Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, USA
[c] Department of Mechanical Engineering, Northwestern University, The Technological Institute, 2145 Sheridan Road, Evanston, IL 60208, USA

Received 10 January 2003; received in revised form 30 April 2003; accepted 2 December 2003

Abstract

We present a method for deriving molecular dynamics boundary conditions for use in multiple scale simulations that can be applied at a planar boundary for any solid that has a periodically repeating crystal lattice. The method is based on a linearization in the vicinity of the boundary, and utilizes a Fourier and Laplace transforms in space and time to eliminate the degrees of freedom associated with atoms outside the boundary. This method is straightforward to implement numerically, and thus can be automated for a general crystal lattice. We show that this method reproduces the known kernel for a 1D linear chain, and apply the approach to obtain the damping kernel matrices for two real crystal lattices: the graphene and diamond structures of carbon.

Keywords: Molecular dynamics; Boundary conditions; Multiple scale simulations; Generalized Langevin equation

1. Introduction

Many of the most difficult problems in computational solid mechanics involve physical phenomena at multiple length and time scales, often as small as the atomic scale. Examples include solid fracture, plasticity, and the behavior of nano- and micro-electromechanical systems. In general, these phenomena cannot be described by continuum mechanics alone, since the continuum assumption is violated at the smallest scales of the dynamics. On the other hand, atomistic simulation methods such as molecular dynamics (MD) cannot be used directly, since the largest scales of interest can span (at least) microns and involve tens of billions of atoms; this is far too large for a simulation on current computers for any reasonable simulation time.

[*] Corresponding author. Tel.: +1-847-491-7094; fax: +1-847-491-3915.
E-mail addresses: gjwagne@sandia.gov (G.J. Wagner), ekarpov@northwestern.edu (E.G. Karpov), w-liu@northwestern.edu (W.K. Liu).

0045-7825/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2003.12.012

An important goal, then, is to be able to combine continuum and atomistic descriptions of a problem in a single computation. Most often, this means simulating the dynamics in a localized region, where atomic motions are important, with MD, while using finite elements or similar techniques to represent a surrounding continuum. One technical difficulty that is encountered in this approach, however, is that small-scale waves originating in the MD region cannot be represented in the continuum, and instead are reflected internally back into the MD region. Any attempt to use this sort of multiscale approach should therefore include some boundary condition on the MD region that prevents such internal reflection. The need for some form of damping at the boundaries to remove the small-scale energy is clear, but it is not immediately apparent what form this damping should take.

The problem was solved in principle by Adelman and Doll [1], who studied the interaction of gas molecules on a solid surface. In order to limit computations to a small number of solid atoms on the surface, they assumed a linear form for the interatomic forces in the solid and eliminated the unwanted degrees of freedom by solving the Laplace transformed equations of motion. The equation of motion for the remaining atoms includes a time history integral over the atomic velocities multiplied by a matrix function $\boldsymbol{\beta}(t)$, known as the damping kernel matrix. This time history integral mimics the dissipation of small-scale energy into the eliminated degrees of freedom.

However, except in simple cases such as 1D lattices, it is difficult to solve analytically for $\boldsymbol{\beta}(t)$. More recently, several authors have presented numerical approaches to solving for $\boldsymbol{\beta}(t)$. Cai et al. [4] show how an MD simulation on a domain somewhat larger than the desired region can be used to compute the damping kernel. E and Huang [6] replace the time history integral altogether with a truncated discrete summation, the coefficients of which are optimized to minimize internal reflections. Both of these methods require nontrivial computational effort, and give damping kernels that are not necessarily transferable to atomic lattices other than those on which they were computed.

In this paper, we present a general technique for computing the matrix $\boldsymbol{\beta}(t)$ on atomic lattices whose structure consists of the spatial replication of a basic unit cell. This technique is semi-analytical, in that we use Laplace transforms in time and discrete Fourier transforms in space to reduce the equation of motion to a manageable form. The use of spatial Fourier transforms in the analysis of regular structures was recently discussed by Karpov et al. [7]. The method yields the solution to this equation by inverting matrices that are no larger than the number of degrees of freedom in the unit cell. The resulting algorithm for computing $\boldsymbol{\beta}(t)$ (or its time derivative $\boldsymbol{\theta}(t)$) is concise, easy to program, and gives results that can be reused for any problem involving the same crystal lattice.

Recently two of the current authors, Wagner and Liu, have developed a multiple scale simulation technique that utilizes a projection operation to bridge MD and FEM representations in a coupled simulation [8]. The technique for applying the MD boundary condition derived in that approach is slightly different from the single-scale approach of Adelman and Doll [1]. However, the boundary condition of Wagner and Liu for multiple scales still involves a time history integral including a damping kernel $\boldsymbol{\beta}(t)$, and that damping kernel is the same as the one used in Ref. [1]. Thus, the damping kernel derived in the current work, although it does not make direct reference to coupling with a continuum scale, can be used without change in the multiple scale boundary condition derived by Wagner and Liu.

2. Problem statement

Consider an atomic lattice that, when at equilibrium, consists of a spatially regular pattern in which unit cells are repeated periodically in one, two or three directions. The periodically repeated cell has n_{a} atoms, each of which may be free to move in n_{SD} directions, where n_{SD} is the number of spatial dimensions. Thus the total number of degrees of freedom in each unit cell is $n_{\mathrm{dof}} = n_{\mathrm{a}} \times n_{\mathrm{SD}}$. Each unit cell can be labeled with three indices—l, m, and n—indicating position along axes in the directions of the three primitive vectors of

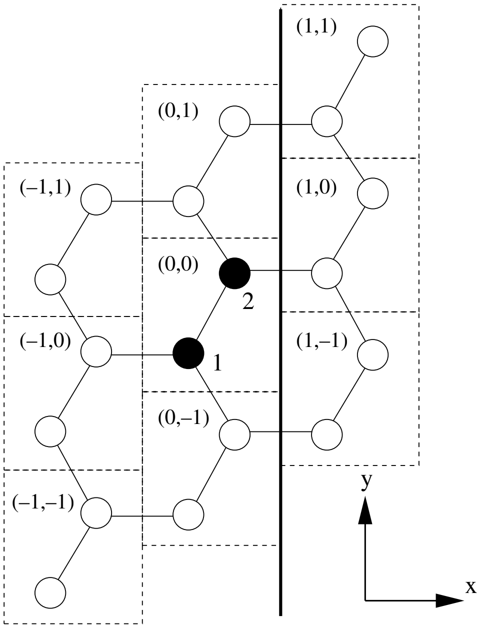

Fig. 1. Periodic structure of the graphene lattice in the zigzag orientation. The thick solid line represents the boundary between the molecular dynamics simulation (to the left) and the surrounding medium (to the right).

the crystal structure. Note that these axes need not be orthogonal. However, for reasons that will be made clear, in this work we will make the assumption that the m and n primitive vectors are coplanar with the y–z plane, so that a collection of unit cells with constant l form a layer that is bounded within a small range of x. An example is shown in Fig. 1, depicting a 2D graphene lattice with a "zigzag" facet lying along a y–z plane (the integer pair in each cell gives the (l, m) coordinates, and $n = 0$ in this 2D structure). Each value of l describes a layer of atoms bounded by y–z planes.

For small displacements of atoms, the interatomic forces can be assumed to be linear in the displacements. The forces on a given unit cell depend on the displacements within that cell, as well as the displacements in the neighboring cells. For the analysis to follow, we require that atoms in a given slab of constant l are coupled only to each other and to atoms in the slabs $l - 1$ and $l + 1$. For longer-range forces, the size of the unit cell must be increased so that this requirement is satisfied. Note that the coupling in the m and n coordinate directions is not limited to immediate neighbors. The force on unit cell (0,0,0) can then be written [7]:

$$\mathbf{f}_{0,0,0}(t) = \sum_{l'=-1}^{1} \sum_{m'=-\mu}^{\mu} \sum_{n'=-\nu}^{\nu} \mathbf{K}_{-l',-m',-n'} \mathbf{u}_{l'm'n'}(t), \tag{2.1}$$

where the matrices $\mathbf{K}_{l,m,n}$ relate the displacements in cell $(-l, -m, -n)$ to the forces in cell (0,0,0), and μ and ν represent the ranges of the forces in the m and n coordinate directions. More generally, because of the translational symmetry of the crystal:

$$\mathbf{f}_{l,m,n}(t) = \sum_{l'=l-1}^{l+1} \sum_{m'=m-\mu}^{m+\mu} \sum_{n'=n-\nu}^{n+\nu} \mathbf{K}_{l-l',m-m',n-n'} \mathbf{u}_{l',m',n'}(t). \tag{2.2}$$

In both of the above expressions, vectors $\mathbf{f}_{l,m,n}$ and $\mathbf{u}_{l,m,n}$ contain all $n_{\mathrm{a}} \times n_{\mathrm{SD}}$ forces and displacements (respectively) for unit cell (l, m, n).

The equation of motion for a unit cell using this linear approximation is then

$$\ddot{\mathbf{u}}_{l,m,n}(t) = \sum_{l'=l-1}^{l+1} \sum_{m'=m-\mu}^{m+\mu} \sum_{n'=n-v}^{n+v} \mathbf{M}^{-1}\mathbf{K}_{l-l',m-m',n-n'}\mathbf{u}_{l',m',n'}(t) + \mathbf{M}^{-1}\mathbf{f}^{\text{ext}}_{l,m,n}(t), \tag{2.3}$$

where $\mathbf{M}$ is the diagonal matrix of atomic masses of a unit cell and $\mathbf{f}^{\text{ext}}_{l,m,n}$ is the external force acting on cell (l, m, n).

Eq. (2.3) gives a system of equations for all of the atoms in the crystal lattice. In general, this is a very large number of atoms, and we will wish to restrict the computation to some subset of the system, say $l \leqslant 0$. However, the effects of the atoms with $l > 0$ cannot be completely ignored, as their presence will affect the dynamics of the atoms near the $l = 0$ boundary. Our goal therefore is to eliminate the $l > 0$ atoms from Eq. (2.3) analytically, leaving a modified equation of motion for the remaining atoms; this modification will take the form of a boundary condition at $l = 0$ that exactly accounts for the effects of the eliminated atoms. It is very important to note that this boundary condition can then be used in a standard molecular dynamics computation in which nonlinear interatomic forces are used for all atoms with $l \leqslant 0$; the assumption of linearity is necessary here only for $l > 0$.

In order to eliminate the $l > 0$ atoms from (2.3), we must first define several transform operation that will be used. This is done in the following section.

3. Transform definitions

3.1. Discrete Fourier transform

A discrete Fourier (DFT) transform will be used to transform spatially varying functions from real space to wavenumber space. Assume that a function f is defined at all lattice positions l; the value of f at lattice position l will be denoted f_l. We assume that the lattice consists of L sites, so that $-(L/2) + 1 < n < L/2$. The discrete Fourier transform of f, which we will denote using a hatted notation, is then given by

$$\hat{f}(p) = \mathscr{F}_{l\to p}\{f_l\} \equiv \sum_{l=-\frac{L}{2}+1}^{\frac{L}{2}} f_l \mathrm{e}^{-\mathrm{i}2\pi pl/L}, \tag{3.1}$$

where p can take any integer value between $-(L/2) + 1$ and $L/2$. The inverse Fourier transform (IFT) can be computed via

$$f_l = \mathscr{F}^{-1}_{p\to l}\{\hat{f}(p)\} \equiv \frac{1}{L} \sum_{p=-\frac{L}{2}+1}^{\frac{L}{2}} \hat{f}(p)\mathrm{e}^{\mathrm{i}2\pi pl/L}. \tag{3.2}$$

In some cases, it will be useful to consider the limiting case of an infinitely long chain. In this case, $-\infty < l < +\infty$, but rather than taking integer values the wavenumber $\bar{p}$ is mapped to the real numbers between $-\pi$ and π. The DFT and IFT are then given by

$$\hat{f}(\bar{p}) = \sum_{l=-\infty}^{\infty} f_l \mathrm{e}^{-\mathrm{i}\bar{p}l}, \tag{3.3a}$$

$$f_l = \frac{1}{2\pi} \int_{-\pi}^{\pi} \hat{f}(\bar{p})\mathrm{e}^{\mathrm{i}\bar{p}l}\mathrm{d}\bar{p}. \tag{3.3b}$$

This limiting case is often convenient because it allows the IFT to be computed analytically.

One important property of the DFT is that the transform of a convolution summation of two functions in space is equal to the product of the transforms of the individual functions:

$$\mathscr{F}_{l\to p}\left\{\sum_{l'=-\frac{L}{2}+1}^{\frac{L}{2}} f_{l-l'}g_{l'}\right\} = \hat{f}(p)\hat{g}(p). \tag{3.4}$$

3.2. Laplace transform

The Laplace transform (LT) will be used to transform functions of time t into the transformed variable s. The LT of a function $f(t)$ is defined as

$$F(s) = \mathscr{L}\{f(t)\} \equiv \int_0^\infty f(t)\mathrm{e}^{-st}\,\mathrm{d}t. \tag{3.5}$$

The inverse Laplace transform (ILT) is given by

$$f(t) = \mathscr{L}^{-1}\{F(s)\} \equiv \frac{1}{2\pi \mathrm{i}}\int_{c-\mathrm{i}\infty}^{c+\mathrm{i}\infty} F(s)\mathrm{e}^{st}\,\mathrm{d}s, \tag{3.6}$$

where c is a real constant greater than the real parts of all singularities of $F(s)$.

An important property of the Laplace transform is that, similar to the DFT, the transform of a convolution integral of two functions is equal to the product of the transforms of the individual functions:

$$\mathscr{L}\left\{\int_0^t f(t-t')g(t')\,\mathrm{d}t'\right\} = F(s)G(s). \tag{3.7}$$

Another important property of Laplace transforms is that the transform of a time derivative is equivalent to multiplication of a function by s. More generally:

$$\mathscr{L}\left\{\frac{\mathrm{d}^n f(t)}{\mathrm{d}t^n}\right\} = s^n F(s) - s^{n-1}f(0) - s^{n-2}\frac{\mathrm{d}f}{\mathrm{d}t}(0) - \cdots - \frac{\mathrm{d}^{n-1}f}{\mathrm{d}t^{n-1}}(0). \tag{3.8}$$

4. Boundary condition derivation

By using the Fourier and Laplace transforms defined in the previous section we can eliminate the atoms in the $l > 0$ unit cells from Eq. (2.3) by solving for those degrees of freedom in term of the $l \leqslant 0$ degrees of freedom, and resubstituting. Since forces on a cell depend on displacements in that cell and its immediately neighboring cells, this amounts to solving for the $l = 1$ displacements in terms of those at $l = 0$, which we can treat as “given.” The key to the problem is the realization that as far as the displacements at $l = 1$ are concerned, it does not matter if the $l = 0$ displacements arise due to the motion of the $l < 0$ atoms or to some external force. Thus we are free to stipulate that our known $l = 0$ displacements are enforced in Eq. (2.3) by an external constraint force, to be determined, that acts only at $l = 0$:

$$\mathbf{f}^{\mathrm{ext}}_{l,m,n}(t) = \delta_{l,0}\mathbf{f}^{\mathrm{ext}}_{0,m,n}(t). \tag{4.1}$$

Taking the LT and DFT of (2.3) eliminates spatial coupling and time derivatives. Denoting p, q, and r as the wavenumbers corresponding to spatial indices l, m, and n, respectively, and assuming all initial velocities and displacements to be zero:

$$s^2\widehat{\mathbf{U}}(p,q,r,s) = \widehat{\mathbf{A}}(p,q,r)\widehat{\mathbf{U}}(p,q,r,s) + \mathbf{M}^{-1}\widehat{\mathbf{F}}^{\mathrm{ext}}_0(q,r,s), \tag{4.2}$$

where

$$\widehat{\mathbf{U}}(p,q,r,s) = \mathscr{L}\{\mathscr{F}_{l,m,n\to p,q,r}\{\mathbf{u}_{l,m,n}(t)\}\}, \tag{4.3a}$$

$$\widehat{\mathbf{A}}(p,q,r) = \mathscr{F}_{l,m,n\to p,q,r}\{\mathbf{M}^{-1}\mathbf{K}_{l,m,n}\}, \tag{4.3b}$$

$$\widehat{\mathbf{F}}_0^{\mathrm{ext}}(q,r,s) = \mathscr{L}\left\{\mathscr{F}_{m,n\to q,r}\left\{\mathbf{f}_{0,m,n}^{\mathrm{ext}}(t)\right\}\right\}. \tag{4.3c}$$

Note that because the matrices $\mathbf{K}_{l,m,n}$ are nonzero only for a small range of values of l, m and n (dependent on the range of the interatomic forces), the DFT $\widehat{\mathbf{A}}(p,q,r)$ is very easy to compute.

Eq. (4.2) can be solved to give the LT/DFT of the displacements in terms of the external force:

$$\widehat{\mathbf{U}}(p,q,r,s) = [s^2\mathbf{I} - \widehat{\mathbf{A}}(p,q,r)]^{-1}\mathbf{M}^{-1}\widehat{\mathbf{F}}_0^{\mathrm{ext}}(q,r,s). \tag{4.4}$$

The IFT in the x-direction of this expression gives the displacement at x-position l:

$$\widetilde{\mathbf{U}}_l(q,r,s) = \widetilde{\mathbf{T}}_l(q,r,s)\mathbf{M}^{-1}\widehat{\mathbf{F}}_0^{\mathrm{ext}}(q,r,s), \tag{4.5}$$

where

$$\widetilde{\mathbf{U}}_l(q,r,s) = \mathscr{F}_{p\to l}^{-1}\{\widehat{\mathbf{U}}(p,q,r,s)\}, \tag{4.6a}$$

$$\widetilde{\mathbf{T}}_l(q,r,s) = \mathscr{F}_{p\to l}^{-1}\{\widehat{\mathbf{T}}(p,q,r,s)\}, \tag{4.6b}$$

$$\widehat{\mathbf{T}}(p,q,r,s) = [s^2\mathbf{I} - \widehat{\mathbf{A}}(p,q,r)]^{-1}. \tag{4.6c}$$

We use the tilde notation to denote mixed space–wavenumber functions, i.e., functions of space in the x-direction (through index l) and wavenumber in the y- and z-directions (through q and r, respectively).

To obtain $\widetilde{\mathbf{U}}_1$ in terms of $\widetilde{\mathbf{U}}_0$, we can write (4.5) for both $l = 0$ and $l = 1$ and eliminate $\widehat{\mathbf{F}}_0^{\mathrm{ext}}$:

$$\widetilde{\mathbf{U}}_1(q,r,s) = \widetilde{\mathbf{T}}_1(q,r,s)\widetilde{\mathbf{T}}_0^{-1}(q,r,s,)\widetilde{\mathbf{U}}_0(q,r,s). \tag{4.7}$$

We have succeeded in expressing the desired interdependence of the displacements, but in order to make use of this expression we must invert the Fourier and Laplace transforms. For convenience, we introduce the intermediate quantity $\mathbf{Q}_{m,n}(s)$ and its Fourier transform:

$$\widetilde{\mathbf{Q}}(q,r,s) = \widetilde{\mathbf{T}}_1(q,r,s)\widetilde{\mathbf{T}}_0^{-1}(q,r,s,), \tag{4.8a}$$

$$\mathbf{Q}_{m,n}(s) = \mathscr{F}_{q,r\to m,n}^{-1}\{\widetilde{\mathbf{Q}}(q,r,s)\}. \tag{4.8b}$$

Now, by (4.7) and the convolution property of the DFT (Eq. (3.4)):

$$\mathbf{U}_{1,m,n}(s) = \sum_{m'=-\frac{M}{2}+1}^{\frac{M}{2}} \sum_{n'=-\frac{N}{2}+1}^{\frac{N}{2}} \mathbf{Q}_{m-m',n-n'}(s)\mathbf{U}_{0,m',n'}(s). \tag{4.9}$$

At this stage, the ILT of Eq. (4.9) can be taken, resulting in a displacement boundary condition on the layer $l = 1$. In fact, this may be the most straightforward approach for boundary conditions on a stand-alone MD simulation. For a multiscale simulation, however, it is often more convenient to have a force boundary condition for atoms in layer $l = 0$. For example, in the coarse-fine scale decomposition of Ref. [8], this force is interpreted as that due to the fine scale degrees of freedom that are unaccounted for in the coarse-scale simulation, and is added to the coarse-scale force. Furthermore, previous work by other authors [1,2,4,6] is framed in terms of a force boundary condition. For these reasons, we continue in Laplace space from Eq. (4.9) and seek a force boundary condition on layer $l = 0$.

The penultimate step in the derivation of such a boundary condition is the calculation of $\mathbf{F}^{1\to 0}_{m,n}(s)$, the force on the $l = 0$ layer due to the $l = 1$ layer. This calculation requires yet another convolution involving the stiffness matrices; by Eq. (2.2), this force is:

$$\mathbf{F}^{1\to 0}_{m,n}(s) = \sum_{m'=m-\mu}^{m+\mu} \sum_{n'=n-v}^{n+v} \mathbf{K}_{-1,m-m',n-n'} \mathbf{U}_{1,m',n'}(s) = \sum_{m'=-\frac{M}{2}+1}^{\frac{M}{2}} \sum_{n'=-\frac{N}{2}+1}^{\frac{N}{2}} \mathbf{\Theta}_{m-m',n-n'}(s)\mathbf{U}_{0,m',n'}(s), \tag{4.10}$$

where

$$\mathbf{\Theta}_{m-m',n-n'}(s) = \sum_{m'=m-\mu}^{m+v} \sum_{n'=n-v}^{n+v} \mathbf{K}_{-1,m-m',n-n'} \mathbf{Q}_{m',n'}(s). \tag{4.11}$$

Finally, taking the ILT of (4.10) completes the boundary condition. We define the time history kernel $\boldsymbol{\theta}_{m,n}(t)$ as the ILT of $\mathbf{\Theta}_{m,n}(s)$:

$$\boldsymbol{\theta}_{m,n}(t) = \mathscr{L}^{-1}\{\mathbf{\Theta}_{m,n}(s)\}. \tag{4.12}$$

By the convolution rule for the Laplace transform (3.7), the force boundary condition takes the form

$$\mathbf{f}^{1\to 0}_{m,n}(t) = \sum_{m'=-\frac{M}{2}+1}^{\frac{M}{2}} \sum_{n'=-\frac{N}{2}+1}^{\frac{N}{2}} \int_0^t \boldsymbol{\theta}_{m-m',n-n'}(t-\tau)\mathbf{u}_{0,m',n'}(\tau)\,\mathrm{d}\tau. \tag{4.13}$$

In practice, the double summation in the above expression cannot be evaluated as written at every time step, since M and N are expected to be large. Instead, the summations should be truncated to some manageable range around m and n (such as $|m'-m|$ and $|n'-n| \leqslant 1$ or 2), depending on the accuracy desired. As will be seen in the examples in Section 6, the sizes of elements of matrix $\boldsymbol{\theta}_{m,n}(t)$ decrease quickly with increasing m and n.

It is more conventional to express the force boundary condition in terms of an integral over velocity, rather than position as in Eq. (4.13), so that:

$$\mathbf{f}^{1\to 0}_{m,n}(t) = \sum_{m'=-\frac{M}{2}+1}^{\frac{M}{2}} \sum_{n'=-\frac{N}{2}+1}^{\frac{N}{2}} \left[\boldsymbol{\beta}_{m-m',n-n'}(t)\mathbf{u}_{0,m',n'}(0) - \boldsymbol{\beta}_{m-m',n-n'}(0)\dot{\mathbf{u}}_{0,m',n'}(t) - \int_0^t \boldsymbol{\beta}_{m-m',n-n'}(t-\tau)\dot{\mathbf{u}}_{0,m',n'}(\tau)\,\mathrm{d}\tau\right], \tag{4.14}$$

where the damping kernel $\boldsymbol{\beta}_{m,n}(t)$ is

$$\boldsymbol{\beta}_{m,n}(t) = \int_t^{\infty} \boldsymbol{\theta}_{m,n}(\tau)\,\mathrm{d}\tau. \tag{4.15}$$

The Laplace transform of the damping kernel is related to that of $\boldsymbol{\theta}_{m,n}(t)$:

$$\boldsymbol{\beta}_{m,n}(t) = \mathscr{L}^{-1}\left\{\frac{1}{s}(\mathbf{\Theta}_{m,n}(0) - \mathbf{\Theta}_{m,n}(s))\right\}. \tag{4.16}$$

4.1. Summary of algorithm

The procedure developed above for computing the kernel functions $\boldsymbol{\theta}_{m,n}(t)$ and $\boldsymbol{\beta}_{m,n}(t)$ is now summarized. We will assume that there are L, M and N unit cells in the x-, y- and z-directions, respectively; each of these parameters can be taken of order 10–100 to mimic an infinite or very large crystal lattice. In some cases, it may be possible to let one or more of these parameters go to infinity, allowing an integral representation of the IFT (Eq. (3.3)). Taking these parameters to be finite allows use of a numerical Fourier transform; in this case, the kernel functions can be truncated at some value of t to avoid artificial reflections or periodic images from far away boundaries.

In order to take the ILT numerically (Section 5.1), we must evaluate the LT at a number of discrete s values. The number of s values is equal to the number of Laguerre polynomials used in the expansion of the transform; we will denote this number by S.

Now we can write step-by-step the algorithm derived above. We assume that the stiffness matrices $\mathbf{K}$ are known for $|l| \leqslant 1$, $|m| \leqslant \mu$ and $|n| \leqslant v$ for a given crystal structure. Each of these matrices is of size $n_{\mathrm{dof}} \times n_{\mathrm{dof}}$, where $n_{\mathrm{dof}} = n_{\mathrm{a}} \times n_{\mathrm{SD}}$. We also assume that the kernel functions are only needed for a small number of values of m and n, since it will be found that the relative sizes of these functions decay quickly with increasing m and n. We define $m_{\max}$ and $n_{\max}$ as the largest values of these parameters to be computed. Then we can proceed according to the following steps:

1. For each (p,q,r), form $\widehat{\mathbf{A}}(p,q,r)$:

$$\widehat{\mathbf{A}}(p,q,r) = \sum_{l=-1}^{+1} \sum_{m=-\mu}^{+\mu} \sum_{n=-v}^{+v} \mathbf{M}^{-1}\mathbf{K}_{l,m,n}\mathrm{e}^{-\mathrm{i}2\pi(lp/L+mq/M+nr/N)}. \tag{4.17}$$

2. For each (p,q,r,s), compute $\widehat{\mathbf{T}}(p,q,r,s)$:

$$\widehat{\mathbf{T}}(p,q,r,s) = \left[s^2\mathbf{I} - \widehat{\mathbf{A}}(p,q,r,s)\right]^{-1}. \tag{4.18}$$

3. For each (q,r,s), invert the DFT in the x-direction at $l=0$ and $l=1$:

$$\widetilde{\mathbf{T}}_0(q,r,s) = \frac{1}{L} \sum_{p=-\frac{L}{2}+1}^{\frac{L}{2}} \widehat{\mathbf{T}}(p,q,r,s), \tag{4.19a}$$

$$\widetilde{\mathbf{T}}_1(q,r,s) = \frac{1}{L} \sum_{p=-\frac{L}{2}+1}^{\frac{L}{2}} \widehat{\mathbf{T}}(p,q,r,s)\mathrm{e}^{2\pi\mathrm{i}p/L}. \tag{4.19b}$$

4. For each (q,r,s), compute $\widetilde{\mathbf{Q}}(q,r,s)$:

$$\widetilde{\mathbf{Q}}(q,r,s) = \widetilde{\mathbf{T}}_1(q,r,s)\widetilde{\mathbf{T}}_0^{-1}(q,r,s). \tag{4.20}$$

5. For each m, n and s compute the IFT of $\widetilde{\mathbf{Q}}(q,r,s)$:

$$\mathbf{Q}_{m,n}(s) = \frac{1}{MN} \sum_{q=-\frac{M}{2}+1}^{\frac{M}{2}} \sum_{r=-\frac{N}{2}+1}^{\frac{N}{2}} \widetilde{\mathbf{Q}}(q,r,s)\mathrm{e}^{2\pi\mathrm{i}(mq/M+nr/N)}. \tag{4.21}$$

   Note that this need only be computed for $|m| \leqslant m_{\max} + \mu$ and $|n| \leqslant n_{\max} + v$, and so there is no need to use a fast Fourier transform for this IFT (which gives the value for all m and n).

6. For each m, n and s compute the convolution with the stiffness matrices:

$$\mathbf{\Theta}_{m,n}(s) = \sum_{m'=m-\mu}^{m+1} \sum_{n'=n-1}^{n+1} \mathbf{K}_{-1,m-m',n-n'}\mathbf{Q}_{m',n'}(s). \tag{4.22}$$

   This must be computed only for $|m| \leqslant m_{\max}$ and $|n| \leqslant n_{\max}$.

7. For each m and n, compute the ILT to get the time history kernel:

$$\boldsymbol{\theta}_{m,n}(t) = \mathscr{L}^{-1}\{\mathbf{\Theta}_{m,n}(s)\}. \tag{4.23}$$

Table 1
Order of operations for each step of algorithm

Step	Order
1	$LMNn_{\text{dof}}^2$
2	$LMNSn_{\text{dof}}^3$
3	$LMNSn_{\text{dof}}^2$
4	$MNSn_{\text{dof}}^3$
5	$MNS(2(m_{\max}+\mu)+1)(2(n_{\max}+v)+1)n_{\text{dof}}^2$
6	$S(2m_{\max}+1)(2n_{\max}+1)n_{\text{dof}}^3$
7	$S^2(2m_{\max}+1)(2n_{\max}+1)n_{\text{dof}}^2$

The order of numerical operations for each step is summarized in Table 1. Note that the most expensive steps are 2 and 3, which involve, respectively, inversion and addition of matrices of size $n_{\text{dof}} \times n_{\text{dof}}$ for each value of p, q, r and s. If steps 1–3 can be done analytically, then a large time savings can be achieved.

5. 1D harmonic lattice

The simplest example of a periodic crystal lattice is the 1D harmonic lattice consisting of a chain of atoms of mass m_a connected by linear springs with force constant k. Analytical solutions for the kernel functions $\boldsymbol{\theta}(t)$ and $\boldsymbol{\beta}(t)$ exist for this lattice [2], so it is ideal for testing the procedure described in Section 4. Note that we can drop subscripts m and n for this 1D problem.

Our derivation of the kernel functions can be carried out analytically for this case, and so we will take the limit of $L \to \infty$ and use the forms of the DFT and IFT given in Eq. (3.3). Our unit cell is a single atom with one degree of freedom, so that stiffnesses are given by scalars:

$$K_{-1} = k, \tag{5.1a}$$

$$K_0 = -2k, \tag{5.1b}$$

$$K_1 = k. \tag{5.1c}$$

Using Eq. (4.3b):

$$\widehat{A}(p) = m_a^{-1}\sum_{l=-1}^{+1} K_l e^{-ipl} = \frac{2k}{m_a}(\cos p - 1). \tag{5.2}$$

Then by (4.6b) and (4.6c):

$$T_0(s) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{1}{s^2 2k(1-\cos p)/m_a}\,dp = \frac{1}{s(s^2+4k/m_a)^{1/2}}, \tag{5.3a}$$

$$T_1(s) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{e^{ip}}{s^2+2k(1-\cos p)/m_a}\,dp = \frac{m_a s^2+2k-m_a s(s^2+4k/m_a)^{1/2}}{2ks(s^2+4k/m_a)^{1/2}}. \tag{5.3b}$$

Note that the tilde notation for mixed space–wavenumber functions is unnecessary in one dimension. Continuing the procedure of Section 4:

$$Q(s) = \frac{T_1(s)}{T_0(s)} = \frac{1}{2k}\left(m_a s^2 + 2k - m_a s(s^2+4k/m_a)^{1/2}\right), \tag{5.4a}$$

$$\Theta(s) = kQ(s) = \frac{1}{2}\left(m_a s^2 + 2k - m_a s(s^2 + 4k/m_a)^{1/2}\right), \tag{5.4b}$$

$$\theta(t) = \mathscr{L}^{-1}\{\Theta(s)\} = \frac{2k}{t} J_2(2(k/m_a)^{1/2} t), \tag{5.4c}$$

$$\beta(t) = \mathscr{L}^{-1}\left\{\frac{1}{s}(\Theta(0) - \Theta(s))\right\} = \frac{(km_a)^{1/2}}{t} J_1(2(k/m_a)^{1/2} t), \tag{5.4d}$$

where J_1 and J_2 are first- and second-order Bessel functions, respectively. The expressions found here for $\theta(t)$ and $\beta(t)$ are the same as those found using a continued fraction representation of the 1D lattice [2], validating our analysis.

5.1. Numerical inverse Laplace transforms

The inverse Laplace transforms in Eqs. (5.4c) and (5.4d) were evaluated using standard transform tables. However, in more complicated cases this may not be possible, and the usefulness of the method hinges on our ability to evaluate the ILT numerically. We have implemented a method introduced by Weeks [9] for the numerical inversion of the transform that utilizes an expansion of the inverse in terms of orthonormal Laguerre functions. This method has been found [5] to give excellent accuracy for the inversion of a wide range of functions. Given a function $F(s)$, we expand its inverse transform using a summation:

$$f(t) \approx e^{ct} \sum_{j=0}^{S} a_j L_j\left(\frac{t}{T}\right), \tag{5.5}$$

where c is a positive number that is greater that the real part of the largest singularity of $F(s)$, T is a parameter chosen to optimize convergence, and $L_j(t)$ are Laguerre polynomials. The coefficients an are

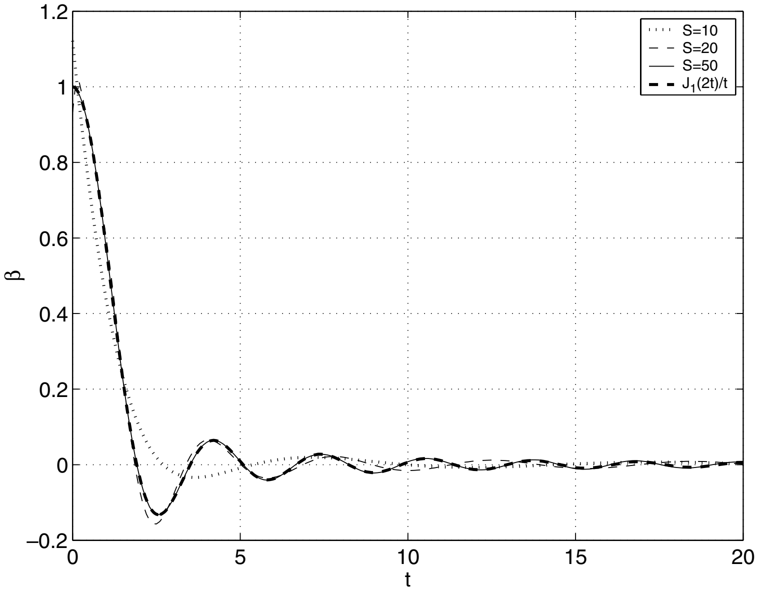

Fig. 2. Numerical ILT for the 1D harmonic chain solution for various values of S, compared with the exact solution. Here, $k = m = 1$.

computed from $F(s)$ using a procedure given by Weeks [9]. We have applied this method for the inverse transform in Eq. (5.4d), using $k = m = 1$, and plotted the results against the analytical solution (5.4d) for various values of S (Fig. 2).

The numerical solution converges nicely to the exact solution for $S = 50$. Because the functions to be inverted, even for very complicated cases, will have forms similar to that in Eq. (5.4c), we can expect that this method of numerical inversion will work well for our problem.

6. Damping matrices for carbon lattices

The damping matrices for various carbon lattices have been computed using the method outlined in the previous sections. For all cases, the interatomic potential used for carbon–carbon interactions is the Tersoff–Brenner potential, with parameters identical to those given as Potential I in Ref. [3]. For each geometry, stiffness matrices were determined by systematically perturbing the positions of atoms in the basic unit cell in each spatial direction, and computing the resulting change in the interatomic forces on the atoms in the system; the elements of the stiffness matrices can then be computed through Eq. (2.2). Because the Tersoff–Brenner potential includes a term dependent on the bond angles in the lattice, the stiffness matrices couple an atom to its nearest neighbors and all neighbors-of-neighbors. For example, in a single hexagon of a graphene lattice, an individual atom is coupled to all other atoms in the hexagon except the one directly opposite it.

The mass used for each carbon atom is 12 amu. The stiffness matrices for the various orientations are given in the Appendix A. For each matrix, the rows and columns are ordered first by atoms in the unit cell, then by spatial degree of freedom. For example, the six rows and columns in the matrices for graphene in the zigzag orientation, which has two atoms per unit cell, are associated in order with the degrees of freedom $(1x, 1y, 1z, 2x, 2y, 2z)$.

6.1. Graphene sheet

The periodic structure of the graphene lattice in the armchair orientation is shown in Fig. 3. In this structure, the basic unit cell contains four atoms, numbered in the figure. In general, a graphene sheet can be reproduced using just two atoms; however, in the armchair orientation such a choice violates the stipulation of our method that the atoms in layer l are coupled only to atoms in layers l and $l \pm 1$. A unit cell with four atoms satisfies this stipulation. Since each atom has a degree of freedom in all three spatial directions, the stiffness matrices given in the Appendix A have size 12×12. Forces on atoms in a cell depend only on displacement in immediately neighboring cells, so $\mu = v = 1$.

The components of the damping matrices were computed using $L = 36$, $M = 36$, $N = 1$, and $S = 150$. Results for various components are given in Figs. 4–7. By far the largest (and therefore most important) component is β_{2x2x} for $(m, n) = (0, 0)$. Note that in our notation here, the subscripts on the scalar quantity β indicate an individual component of the matrix $\boldsymbol{\beta}$, so that β_{2x2x} gives the component governing self-coupling of the velocity of atom 2 in the x-direction. Other components are much smaller. Fig. 7 depicts β_{2x2x} for various values of m, showing how a unit cell is coupled to other unit cells along the boundary (note that n always equals 0 for this 2D geometry). The peak value for $m = 2$ is less than 10% that of $m = 0$, indicating that the unit cell is only weakly coupled to cells far away from it. This is the justification for the truncation of summations in Eqs. (4.13) and (4.14) at small values of $|m' - m|$ and $|n' - n|$. Also interesting is the increasing time delay apparent in Fig. 7 for increasing values of m, which is due to the time required for information to propagate along the boundary.

The periodic structure of the graphene lattice in the zigzag orientation is shown in Fig. 1. In this orientation, unlike the armchair lattice, a unit cell of two atoms can be used while still meeting the requirement

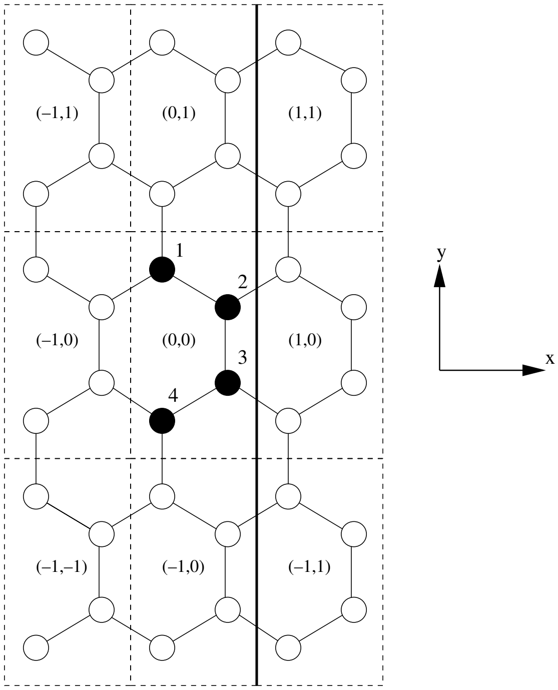

Fig. 3. Periodic structure of the graphene lattice in the armchair orientation. The thick solid line represents the boundary between the molecular dynamics simulation (to the left) and the surrounding medium (to the right).

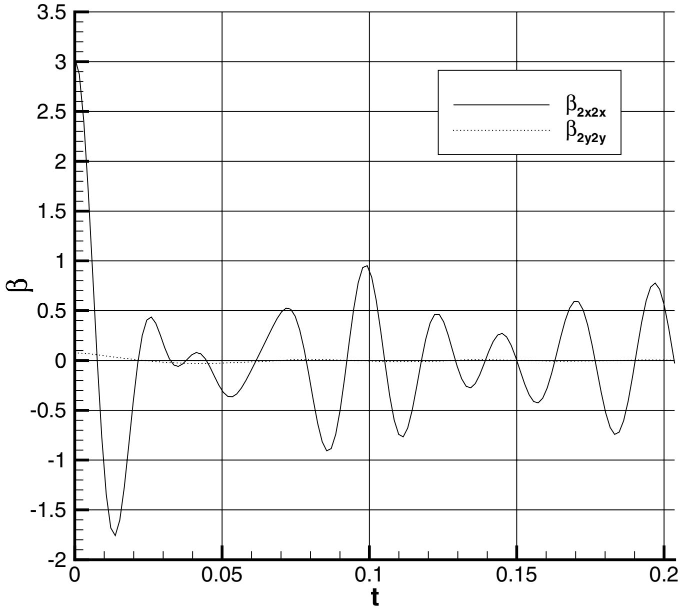

Fig. 4. Components of the damping kernel matrix for armchair graphene at $m = 0$. For this and all following plots, $\beta(t)$ has units of $eV/\mathring{A}^2$, and t has units of *ps*.

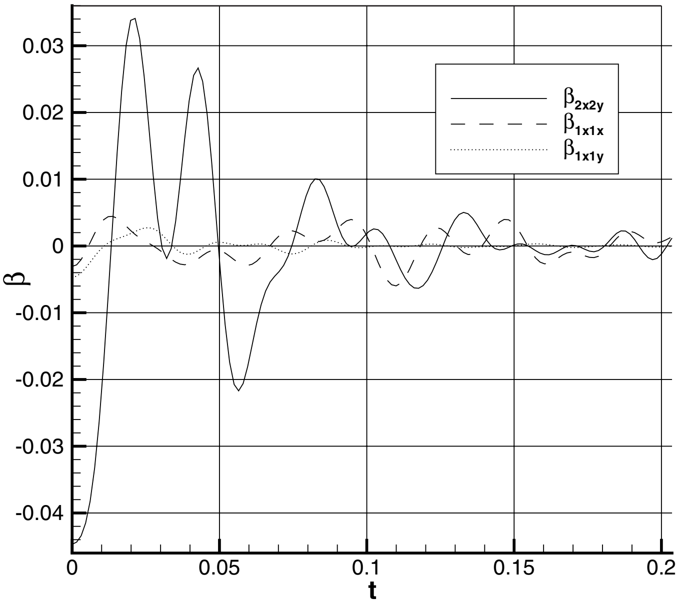

Fig. 5. Components of the damping kernel matrix for armchair graphene at $m = 0$.

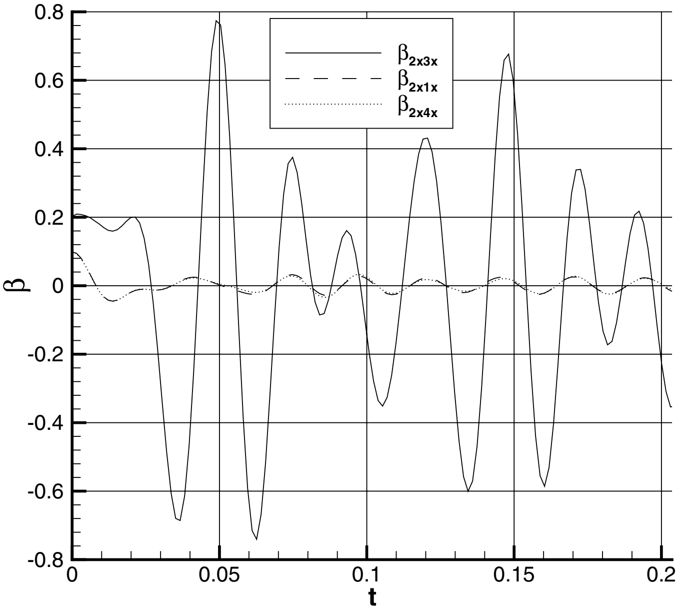

Fig. 6. Components of the damping kernel matrix for armchair graphene at $m = 0$.

that atoms in a given cell are coupled only to atoms in the immediately neighboring cells. Each atom has three spatial degrees of freedom, resulting in stiffness matrices (given in the Appendix A) with size 6×6. Forces on atoms in a cell depend only on displacement in immediately neighboring cells, so $\mu = \nu = 1$.

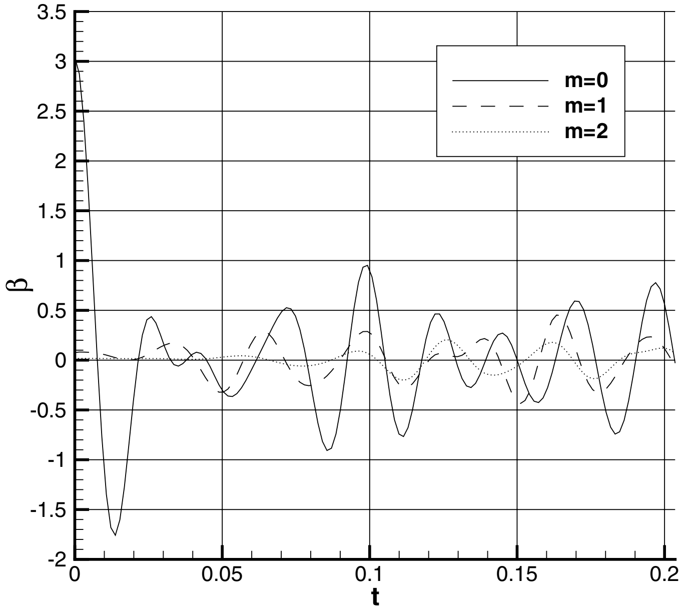

Fig. 7. Functions $\beta_{2x2x}(t)$ for armchair graphene for increasing values of m.

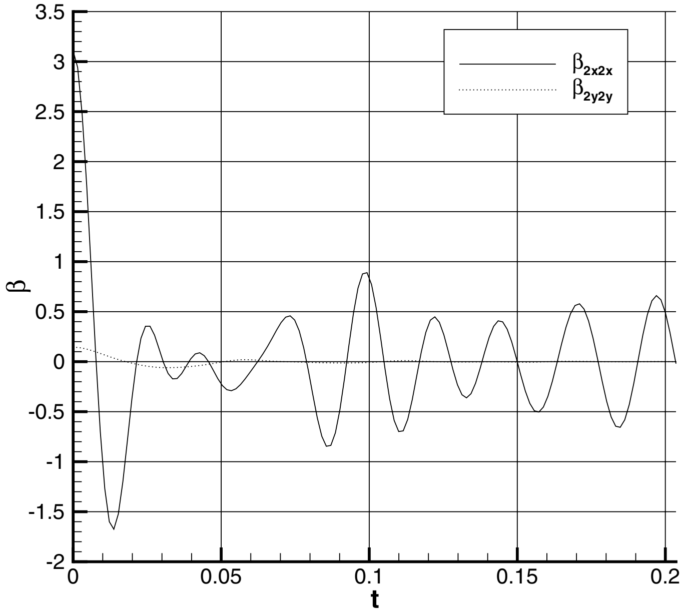

Fig. 8. Components of the damping kernel matrix for zigzag graphene at $m = 0$.

The components of the damping matrices were computed using $L = 36$, $M = 36$, $N = 1$, and $S = 150$. Results for various components are given in Figs. 8–10. Again, by far the largest and most important

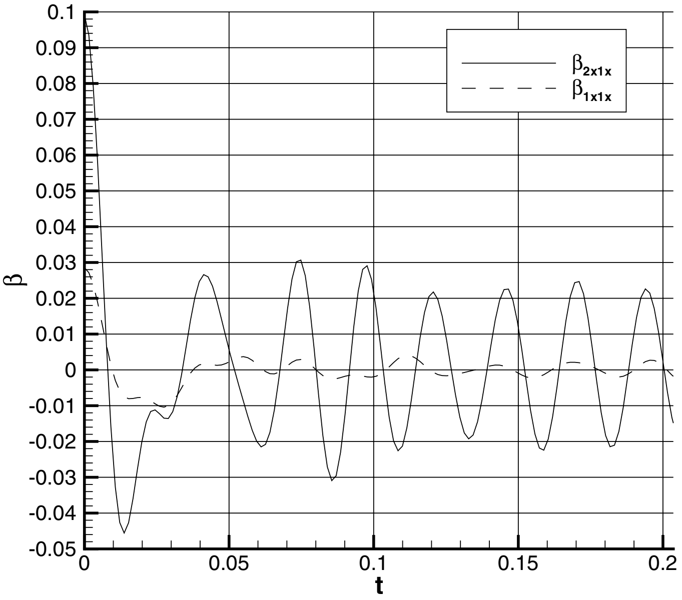

Fig. 9. Components of the damping kernel matrix for zigzag graphene at $m = 0$.

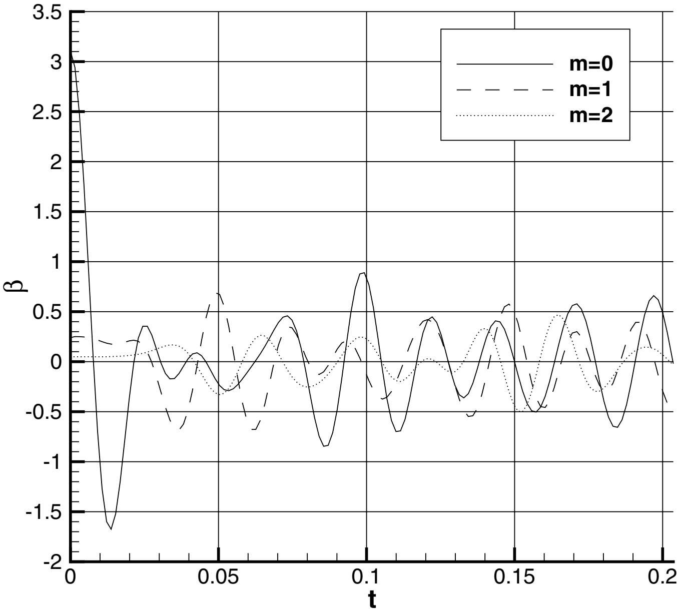

Fig. 10. Functions $\beta_{2x2x}(t)$ for armchair graphene for increasing values of m.

component is β_{2x2x} for $(m, n) = (0, 0)$. In fact, this function for the zigzag orientation is very similar to that for the armchair orientation. Fig. 10 shows that the peak value of β_{2x2x} again decreases with

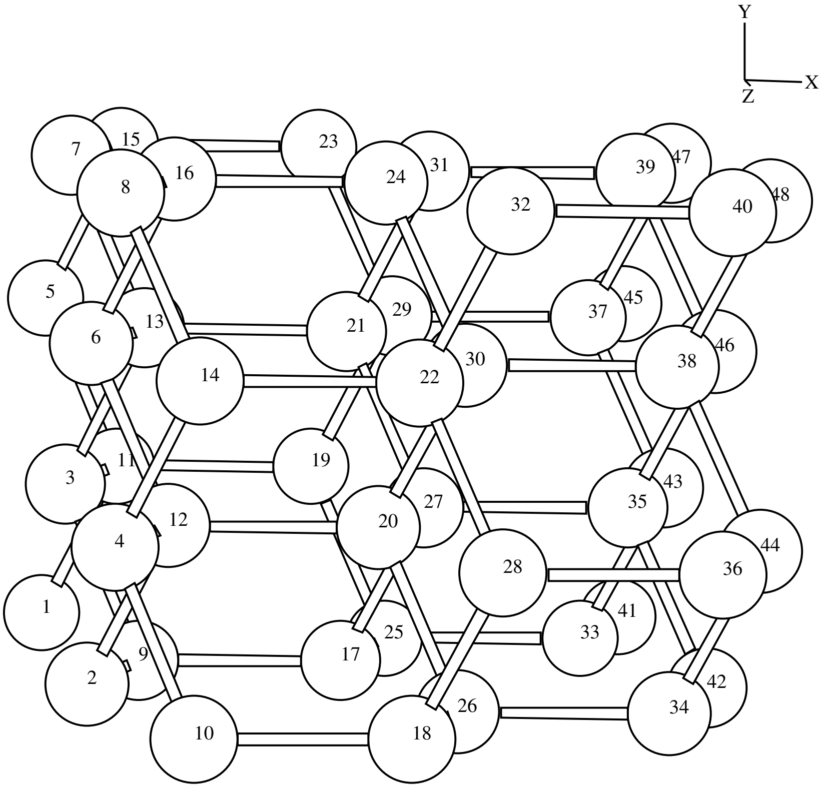

Fig. 11. Periodic structure of the diamond lattice. The $\langle 1\,1\,1\rangle$ direction is parallel to the x-axis.

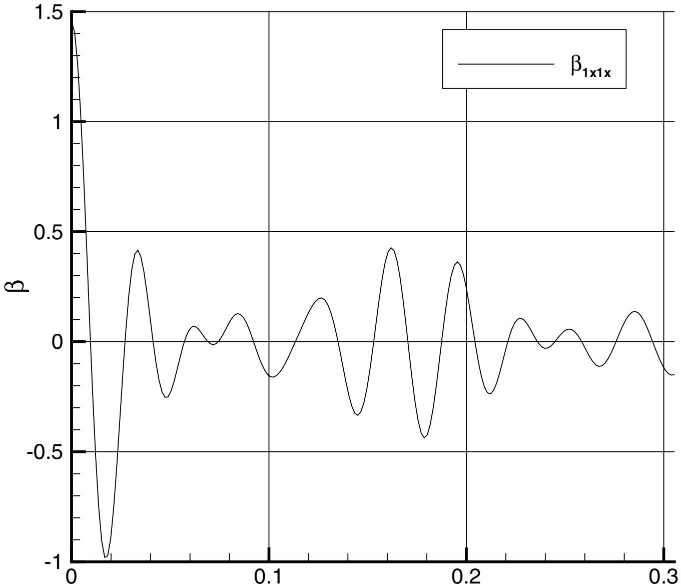

Fig. 12. Components of the damping kernel matrix for diamond at $m = n = 0$.

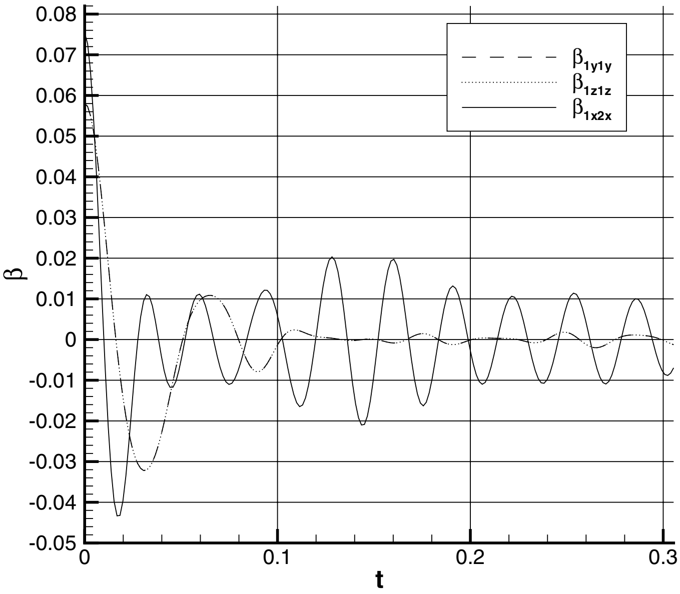

Fig. 13. Components of the damping kernel matrix for diamond at $m = n = 0$.

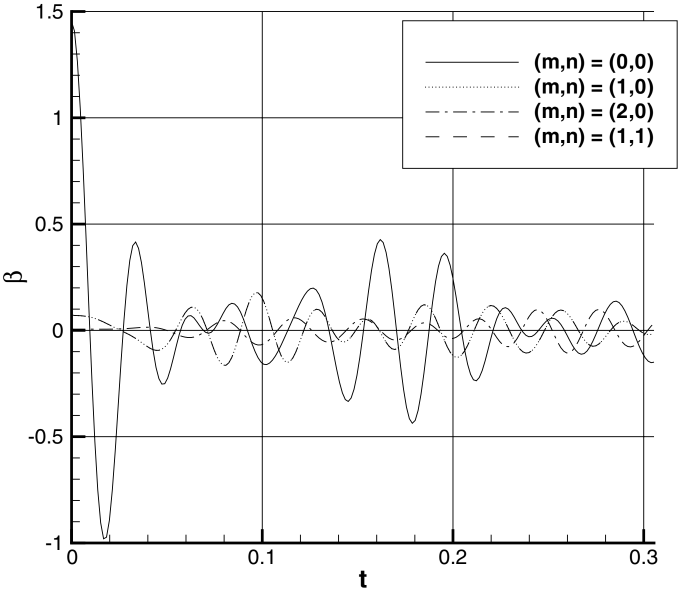

Fig. 14. Functions $\beta_{1x1x}(t)$ for diamond for various values of m and n. The curves for $(m, n) = (1, 0)$ and $(m, n) = (1, 1)$ overlay each other due to the geometry of the lattice.

increasing m; however, the decrease is not as large due to the smaller size of the unit cell in the zigzag orientation.

6.2. Diamond lattice

The periodic structure of the 3D diamond lattice is more difficult to visualize than that of 2D graphene. Fig. 11 shows a small periodic section of a diamond lattice, with the $\langle 1\,1\,1 \rangle$ direction parallel to the x-axis; this is the orientation for which we will derive the damping matrices. In the figure, if the pair $(30, 22)$ is taken to be the base $(0, 0, 0)$ unit cell (with atom #30 as cell atom 1, and #22 as cell atom 2), then the pair $(46, 38)$ is the $(1, 0, 0)$ cell, $(31, 24)$ is the $(0, 1, 0)$ cell, and $(29, 21)$ is the $(0, 0, -1)$ cell. Each atom has three spatial degrees of freedom, resulting in stiffness matrices (given in the Appendix A) of size 6×6. Unlike the other lattices examined so far, forces on an atom depend on displacements two unit cells away in the m coordinate direction; for example, in Fig. 11 atom #30 is coupled through angular dependence at atom #20 with atom #26, which is in the cell $(0, -2, -1)$. Therefore we must take $\mu = 2$, $\nu = 1$.

The components of the damping matrices were computed using $L = 24$, $M = 24$, $N = 24$, and $S = 150$. Results for various components are given in Figs. 12–14. The largest and most important component is β_{1x1x} for $(m, n) = (0, 0)$ (Fig. 12). Fig. 14 shows the decrease in the peak value of this function for increasing m and n. Note that in the figure, the curves for $(m, n) = (1, 0)$ and $(m, n) = (1, 1)$ overlay each other due to the geometry of the lattice.

7. Conclusions

We have presented a method for deriving the time history kernel matrices that can be used for applying nonreflecting boundary conditions for a molecular dynamics simulation. The method is based on the assumptions that the atomic lattice has a regular periodic structure near the boundary, that the boundary is planar, and that the interatomic force can be approximated as a linear function of atomic displacements (i.e., the harmonic approximation). For certain simple cases, this method can be performed completely analytically, yielding exact expressions for these matrices as functions of time. However, all steps of the method can be carried out numerically with a reasonable number of operations, including the computation of the stiffness matrices, allowing for a simple and fully automated implementation of the method.

The time history kernels are nonzero even for fairly large times and separations in space. In order to make the computation of the final boundary condition ((4.13) or (4.14)) feasible, integrals and summations must both be truncated over suitable ranges. Decisions about where to make these truncations should be based on the relative sizes and decay rates of the time history kernel matrix components, and the method presented allows these functions to be easily computed and compared. Future research into the behavior of these boundary conditions for different choices of truncation will help to determine how far to extend summations and integrals in space and time in order to achieve a given level of accuracy.

The method presented is certainly not applicable to all situations; in particular, it is not expected to give accurate boundary conditions where the boundary is not planar (e.g., near a corner) or where the atomic structure is not periodic (e.g., near a dislocation, or for an amorphous structure). However, in many MD simulations the features of interest are located well away from the boundaries, which are in fact well described by the planar, periodic assumptions we have made; the kernel functions derived here are likely to give all the accuracy that is needed in these cases. More general boundary shapes can be obtained by combining multiple plane boundaries; the effects at corners, which are not described by the current analysis, will be studied and reported on in the future. For more complicated cases, it may be possible to use a more general but more computationally intensive procedure, such as that of Cai et al. [4], for deriving the boundary conditions in localized areas, while applying the functions derived according to the method presented here for large areas of the boundary where the planar and periodic assumptions apply.

Acknowledgements

The work of G.J. Wagner is supported by U.S. DOE Contract AC04-94AL85000. E.G. Karpov and W.K. Liu gratefully acknowledge the support of NSF and NSF IGERT.

Appendix A. Stiffness matrices for carbon structures

All stiffness values are given in units $\mathrm{eV}/\mathring{\mathrm{A}}^2$.

A.1. Graphene sheet, armchair orientation

$$\mathbf{K}_{-1,-1} = \mathbf{K}_{1,1}^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.53 & -0.82 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.37 & 1.42 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.99 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.1a}$$

$$\mathbf{K}_{-1,0} = \mathbf{K}_{1,0}^T$$

$$= \begin{bmatrix} 1.87 & -1.59 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.59 & 0.08 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.99 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 35.7 & 18.3 & 0 & 1.87 & 1.59 & 0 & 0 & 0 & 0 & 0.53 & -2.37 & 0 \\ 18.3 & 14.6 & 0 & -1.59 & 0.08 & 0 & 0 & 0 & 0 & 0.82 & 1.42 & 0 \\ 0 & 0 & 4.92 & 0 & 0 & -0.99 & 0 & 0 & 0 & 0 & 0 & -0.99 \\ 0.52 & 2.37 & 0 & 0 & 0 & 0 & 1.87 & -1.59 & 0 & 35.7 & -18.3 & 0 \\ -0.82 & 1.42 & 0 & 0 & 0 & 0 & 1.59 & 0.08 & 0 & -18.3 & 14.6 & 0 \\ 0 & 0 & -0.99 & 0 & 0 & 0 & 0 & 0 & -0.99 & 0 & 0 & 4.92 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.87 & 1.59 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.59 & 0.08 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.99 \end{bmatrix}, \tag{A.1b}$$

$$\mathbf{K}_{-1,1} = \mathbf{K}_{1,-1}^{T} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.53 & 0.82 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -2.37 & 1.42 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.99 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.1c}$$

$$\mathbf{K}_{0,-1} = \mathbf{K}_{0,1}^{T} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0.53 & 2.37 & 0 & 4.02 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.82 & 1.42 & 0 & 0 & 46.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.99 & 0 & 0 & 4.92 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.53 & 0.82 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2.37 & 1.42 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -0.99 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \tag{A.1d}$$

$$\mathbf{K}_{0,0} = \begin{bmatrix} -81.2 & 0 & 0 & 35.7 & -18.3 & 0 & 0.53 & 0.82 & 0 & 0 & 0 & 0 \\ 0 & -81.2 & 0 & -18.3 & 14.6 & 0 & -2.37 & 1.42 & 0 & 0 & 0 & 0 \\ 0 & 0 & -8.81 & 0 & 0 & 4.92 & 0 & 0 & -0.99 & 0 & 0 & 0 \\ 35.7 & -18.3 & 0 & -81.2 & 0 & 0 & 4.02 & 0 & 0 & 0.53 & 2.37 & 0 \\ -18.3 & 14.6 & 0 & 0 & -81.2 & 0 & 0 & 46.2 & 0 & -0.82 & 1.42 & 0 \\ 0 & 0 & 4.92 & 0 & 0 & -8.81 & 0 & 0 & 4.92 & 0 & 0 & -0.99 \\ 0.53 & -2.37 & 0 & 4.02 & 0 & 0 & -81.2 & 0 & 0 & 35.7 & 18.3 & 0 \\ 0.82 & 1.42 & 0 & 0 & 46.2 & 0 & 0 & -81.2 & 0 & 18.3 & 14.6 & 0 \\ 0 & 0 & -0.99 & 0 & 0 & 4.92 & 0 & 0 & -8.81 & 0 & 0 & 4.92 \\ 0 & 0 & 0 & 0.53 & -0.82 & 0 & 35.7 & 18.3 & 0 & -81.2 & 0 & 0 \\ 0 & 0 & 0 & 2.37 & 1.42 & 0 & 18.3 & 14.6 & 0 & 0 & -81.2 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.99 & 0 & 0 & 4.92 & 0 & 0 & -8.81 \end{bmatrix}. \tag{A.1e}$$

A.2. Graphene sheet, zigzag orientation

$$\mathbf{K}_{-1,0} = \mathbf{K}_{1,0}^T = \begin{bmatrix} 1.42 & 2.37 & 0 & 0 & 0 & 0 \\ -0.82 & 0.53 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.99 & 0 & 0 & 0 \\ 46.2 & 0 & 0 & 1.42 & -0.82 & 0 \\ 0 & 4.02 & 0 & 2.37 & 0.53 & 0 \\ 0 & 0 & 4.92 & 0 & 0 & -0.99 \end{bmatrix}, \tag{A.2a}$$

$$\mathbf{K}_{-1,1} = \mathbf{K}_{1,-1}^T = \begin{bmatrix} 1.42 & -2.37 & 0 & 0 & 0 & 0 \\ 0.82 & 0.53 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.99 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.42 & 0.82 & 0 \\ 0 & 0 & 0 & -2.37 & 0.53 & 0 \\ 0 & 0 & 0 & 0 & 0 & -0.99 \end{bmatrix}, \tag{A.2b}$$

$$\mathbf{K}_{0,-1} = \mathbf{K}_{0,1}^T = \begin{bmatrix} 0.08 & -1.59 & 0 & 0 & 0 & 0 \\ 1.59 & 1.87 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.99 & 0 & 0 & 0 \\ 14.6 & -18.3 & 0 & 0.08 & 1.59 & 0 \\ -18.3 & 35.7 & 0 & -1.59 & 1.87 & 0 \\ 0 & 0 & 4.92 & 0 & 0 & -0.99 \end{bmatrix}, \tag{A.2c}$$

$$\mathbf{K}_{0,0} = \begin{bmatrix} -81.2 & 0 & 0 & 14.6 & 18.3 & 0 \\ 0 & -81.2 & 0 & 18.3 & 35.7 & 0 \\ 0 & 0 & -8.81 & 0 & 0 & 4.92 \\ 14.6 & 18.3 & 0 & -81.2 & 0 & 0 \\ 18.3 & 35.7 & 0 & 0 & -81.2 & 0 \\ 0 & 0 & 4.92 & 0 & 0 & -8.81 \end{bmatrix}, \tag{A.2d}$$

$$\mathbf{K}_{-1,-1} = \mathbf{K}_{1,1} = \mathbf{0}. \tag{A.2e}$$

A.3. Diamond lattice

$$\mathbf{K}_{-1,-1,-1} = \mathbf{K}_{1,1,1}^T \begin{bmatrix} 1.24 & -0.38 & -0.22 & 0 & 0 & 0 \\ 1.16 & 0.55 & 0.64 & 0 & 0 & 0 \\ 0.67 & 0.64 & -0.20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.24 & 0.12 & 0.67 \\ 0 & 0 & 0 & -0.38 & 0.55 & 0.64 \\ 0 & 0 & 0 & -0.22 & 0.64 & -0.20 \end{bmatrix}, \tag{A.3a}$$

$$\mathbf{K}_{-1,0,0} = \mathbf{K}^T_{1,0,0}\begin{bmatrix} 1.24 & 0 & 0.44 & 25.0 & 0 & 0 \\ 0 & -0.57 & 0 & 0 & 2.38 & 0 \\ -1.34 & 0 & 0.92 & 0 & 0 & 2.38 \\ 0 & 0 & 0 & 1.24 & 0 & -1.34 \\ 0 & 0 & 0 & 0 & -0.57 & 0 \\ 0 & 0 & 0 & 0.44 & 0 & 0.92 \end{bmatrix}, \tag{A.3b}$$

$$\mathbf{K}_{-1,1,0} = \mathbf{K}^T_{1,-1,0} = \begin{bmatrix} 1.24 & 0.38 & -0.22 & 0 & 0 & 0 \\ -1.16 & 0.55 & -0.64 & 0 & 0 & 0 \\ 0.67 & -0.64 & -0.20 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1.24 & -1.16 & 0.67 \\ 0 & 0 & 0 & 0.38 & 0.55 & -0.64 \\ 0 & 0 & 0 & -0.22 & -0.64 & -0.20 \end{bmatrix}, \tag{A.3c}$$

$$\mathbf{K}_{0,-2,-1} = \mathbf{K}^T_{0,2,1} = \begin{bmatrix} -0.18 & 0.51 & 0.55 & 0 & 0 & 0 \\ -0.51 & 1.56 & -0.73 & 0 & 0 & 0 \\ 0.55 & 0.73 & 0.21 & 0 & 0 & 0 \\ 0 & 0 & 0 & -0.18 & -0.51 & 0.55 \\ 0 & 0 & 0 & 0.51 & 1.56 & 0.73 \\ 0 & 0 & 0 & 0.55 & -0.73 & 0.21 \end{bmatrix}, \tag{A.3d}$$

$$\mathbf{K}_{0,-1,-1} = \mathbf{K}^T_{0,1,1} = \begin{bmatrix} -0.18 & 0.73 & 0.17 & 0 & 0 & 0 \\ 0.22 & 0.55 & 1.31 & 0 & 0 & 0 \\ -0.72 & -0.14 & 1.22 & 0 & 0 & 0 \\ 4.89 & 6.15 & 3.55 & -0.18 & 0.22 & -0.72 \\ 6.15 & 17.4 & 8.70 & 0.73 & 0.55 & -0.14 \\ 3.55 & 8.70 & 7.40 & 0.17 & 1.31 & 1.22 \end{bmatrix}, \tag{A.3e}$$

$$\mathbf{K}_{0,-1,0} = \mathbf{K}^T_{0,1,0} = \begin{bmatrix} -0.18 & -0.22 & -0.72 & 4.89 & -6.15 & 3.55 \\ -0.73 & 0.55 & 0.14 & -6.15 & 17.4 & -8.70 \\ 0.17 & -1.31 & 1.22 & 3.55 & -8.70 & 7.40 \\ 0 & 0 & 0 & -0.18 & -0.73 & 0.17 \\ 0 & 0 & 0 & -0.22 & 0.55 & -1.31 \\ 0 & 0 & 0 & -0.72 & 0.14 & 1.22 \end{bmatrix}, \tag{A.3f}$$

$$\mathbf{K}_{0,0,0} = \begin{bmatrix} -46.0 & 0 & 0 & 4.89 & 0 & -7.10 \\ 0 & -46.0 & 0 & 0 & 2.38 & 0 \\ 0 & 0 & -46.0 & -7.10 & 0 & 22.5 \\ 4.89 & 0 & -7.10 & -46.0 & 0 & 0 \\ 0 & 2.38 & 0 & 0 & -46.0 & 0 \\ -7.10 & 0 & 22.5 & 0 & 0 & -46.0 \end{bmatrix}. \tag{A.3g}$$

All other stiffness matrices are **0**.

References

[1] S.A. Adelman, J.D. Doll, Generalized Langevin equation approach for atom/solid-surface scattering—collinear atom/harmonic chain model, J. Chem. Phys. 61 (10) (1974) 4242–4245.

[2] S.A. Adelman, J.D. Doll, Generalized Langevin equation approach for atom-solid-surface scattering—general formulation for classical scattering off harmonic solids, J. Chem. Phys. 64 (6) (1976) 2375–2388.

[3] D.W. Brenner, Empirical potential for hydrocarbons for use in simulating the chemical vapor deposition of diamond films, Phys. Rev. B 42 (15) (1990) 9458–9471.

[4] W. Cai, M. de Koning, V.V. Bulatov, S. Yip, Minimizing boundary reflections in coupled-domain simulations, Phys. Rev. Lett. 85 (15) (2000) 3213–3216.

[5] B. Davies, B. Martin, Numerical inversion of the Laplace transform—survey and comparison of methods, J. Comput. Phys. 33 (1) (1979) 1–32.

[6] W. E, Z.Y. Huang, Matching conditions in atomistic–continuum modeling of materials, Phys. Rev. Lett. 87 (13) (2001) 135501.

[7] E.G. Karpov, N.G. Stephen, D.L. Dorofeev, On static analysis of finite repetitive structures by discrete Fourier transform, Int. J. Solids Struct. 39 (16) (2002) 4291–4310.

[8] G.J. Wagner, W.K. Liu, Coupling of atomistic and continuum simulations using a bridging scale decomposition, J. Comput. Phys. 190 (2003) 249–274.

[9] W.T. Weeks, Numerical inversion of Laplace transforms using Laguerre functions, J. ACM 13 (3) (1966) 419–429.