



A parallel algorithm for the concurrent atomistic-continuum methodology

Adrian Diaz^{a,*}, Boyang Gu^a, Yang Li^a, Steven J. Plimpton^b, David L. McDowell^{c,d}, Youping Chen^a

^a University of Florida, Department of Mechanical and Aerospace Engineering, Gainesville, FL 32611, USA

^b Sandia National Laboratories, Albuquerque, NM 87185, USA

^c Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

^d School of Materials Science and Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA



ARTICLE INFO

Article history:

Received 27 March 2020

Received in revised form 14 February 2022

Accepted 6 March 2022

Available online 15 April 2022

Keywords:

Concurrent atomistic-continuum method

Molecular dynamics

Nonequilibrium processes

Parallel algorithms

LAMMPS

ABSTRACT

In this work we present a parallel algorithm for the Concurrent Atomistic Continuum (CAC) formulation that can be integrated into existing molecular dynamics codes. The CAC methodology is briefly introduced and its parallel implementation in LAMMPS is detailed and then demonstrated through benchmarks that compare CAC simulation results with corresponding all-MD (molecular dynamics) results. The parallel efficiency of the algorithm is demonstrated when simulating systems represented by both atoms and finite elements. The verification benchmarks include dynamic crack propagation and branching in a Si single crystal, wave propagation and scattering in a Si phononic crystal, and phonon transport through the phase interface in a PbTe/PbSe heteroepitaxial system. In each of these benchmarks the CAC algorithm is shown to be in good agreement with MD-only models. This parallel CAC algorithm thus offers one of the first scalable multiscale material simulation methodologies that relies solely on atomic-interaction models.

© 2022 Published by Elsevier Inc.

1. Introduction

Nanoscale computational methods, such as Density Functional Theory [3,4] (DFT) calculations and Molecular Dynamics [5] (MD) simulations, have enabled our understanding of materials behavior at the nanoscale. They have also been successfully applied to studies of individual defects, interfaces, or simple periodic structures. Once the microstructural feature size exceeds the nanoscale, establishing the link between the material microstructure and the non-equilibrium thermo-mechanical behavior is no longer feasible using these atomistic methods [6,7]. A typical example is a polycrystal that contains micron-sized grains and nanometer-thick grain boundaries, such as those seen in Fig. 1, in which the structure and the connectivity of the grain boundaries may play a significant role on the mechanical and thermal transport properties. The desire to predict the properties and behavior of these material microstructures has incentivized the development of concurrent multiscale methodologies that aim to extend the reach of materials modeling using only potentials for interatomic interaction. The successes and limitations of dynamic simulation with such multiscale methods is reviewed in a previous work [8].

This work focuses on the Concurrent Atomistic Continuum (CAC) method [9], which has demonstrated broad predictive capability [10–20] by supplying a truly concurrent governing equation, but lacks an efficient parallel algorithm for the

* Corresponding author.

E-mail address: adiatz@lanl.gov (A. Diaz).

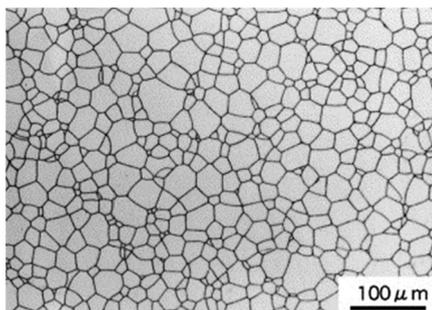


Fig. 1. Ceramic polycrystal exhibiting micrometer scale grains and intricate grain boundaries [1].

simulation of systems involving large disparate morphological length scales. Systems at the mesoscale, such as those in Fig. 1, will still require several million, or even billions, of degrees of freedom (pure atoms and coarse-grained degrees of freedom) to model; thus, such mesoscale endeavors still demand large-scale parallel computation, even when efficiently describing crystalline regions.

The CAC method is numerically implemented using the finite element method; this work specifically employs the Galerkin Method to derive the FEM integral equations of the numerical problem. This leads to a material system, interacting nonlocally, described in terms of both finite elements and atoms to provide variable resolution as needed. At every resolution, the CAC method utilizes an interatomic potential to define the internal forces experienced by the material. Additionally, the numerical solution can be computed without element interconnectivity that is typically required in displacement-based FE solutions. Hence, discontinuities to arise naturally as a solution of the governing equation. This approach also facilitates the application of spatial decomposition strategies employed in MD codes, since the finite elements can be represented as extended particles.

Modern MD algorithms have achieved reasonable parallel efficiency on large supercomputers with up to 100,000 nodes (CPUs or GPUs) or more than a million CPU cores, an achievement resulting from efforts to optimize spatial decomposition algorithms that decompose large systems of particles interacting through nonlocal forces. Enabling comparable performance of the CAC method to MD requires addressing the following additional challenges not present in MD algorithms:

- (1) devise an accurate integration scheme to numerically solve the governing equation, using FEM, for a system with non-local forces and multiple resolutions;
- (2) decompose the collection of particles across processes (i.e. MPI processes) when there are disparate resolutions, meaning large finite elements of multiple sizes as well as atoms (point particles), and efficiently compute the forces between them;
- (3) extend the halo communication strategies of the MD spatial decomposition algorithm to account for extended particles of different sizes and differing interaction distances, with a minimal communication volume.

This work will address these challenges and develop a parallel algorithm to implement the CAC formulation in the widely-used LAMMPS MD code [21]. In general, the motivation to implement CAC in an MD code is that they are already designed to compute non-local force calculations in the context of spatial decomposition methods. The specific choice of LAMMPS is due to its object orientation for aspects of the MD algorithm that required modification to accommodate the CAC method, so that CAC-specific changes could be implemented with minimal effort and impact on the rest of the code. Additionally, LAMMPS has a non-uniform load balancing scheme that can be used to decompose a multiresolution CAC problem efficiently.

Thus, this paper details the following: a description of the CAC spatial decomposition algorithm, the parallel performance of the CAC algorithm on LAMMPS when simulating a multiscale model, and the verification of the implementation through the simulation of several well-known phenomena using both CAC and their equivalent MD-only representations. The material processes that will be simulated as verifications will cater to the CAC method's strengths relative to previous multiscale methodologies. These include:

- (1) Propagation and scattering of a heat pulse in a silicon phononic crystal,
- (2) The structure and density of misfit dislocations in a PbTe/PbSe (001) heterostructure and the interaction between phonons and misfit dislocations in the heterostructure,
- (3) Crack initiation, propagation, and branching in silicon.

In each of these cases, it will be demonstrated that the coarse-grained description has the capability to reproduce essential solutions of these phenomena obtained via MD at full atomic resolution, including phonon scattering and defect dynamics.



Fig. 2. (a) crystal structure = lattice + basis, (b) continuous lattice deformation; (c) discontinuities caused by slip.

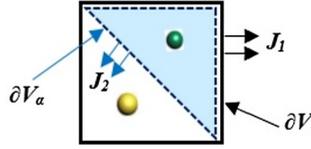


Fig. 3. A unit cell V of 2 atoms, and the two types of fluxes.

This paper is organized by section as follows: Section 2 introduces the CAC formulation, Section 3 presents the major design aspects of the parallel algorithm established on LAMMPS to numerically implement the CAC formulation and discusses the performance of the algorithm, Section 4 discusses and displays the set of verification simulations comparing the CAC method's predictive features to that of MD simulations. Section 5 demonstrates the algorithm with a model containing disparate numerical resolutions. Section 6 concludes our findings and provides a brief outlook on the algorithm's impact and future.

2. The CAC methodology

The CAC method is an atomic-interaction-based multiscale method. The formulation is an extension of the classical statistical mechanical theory of transport processes by Irving and Kirkwood [22,23] to a two-level description of materials deformation: the continuous lattice level deformation and the discrete internal motion of atoms embedded within each lattice point [9,24–26]; see Fig. 2 for a depiction. This leads to a concurrent atomistic-continuum (CAC) methodology [14]. Since CAC is built on the foundation of the classical mechanics of particles, it is fully compatible with the method of MD. It shares MD's power to simultaneously simulate the dynamics of defects and thermal transport without the need of a priori assumptions for mechanisms or empirical parameters other than an interatomic potential. CAC's advantage over MD is the ability to simulate larger-scale processes, via coarse graining the continuous lattice deformation, while retaining the ability to concurrently model at full atomistic resolution in critical regions, such as regions with atomic-scale structural disorder, materials interfaces, crack tips, dislocation cores, etc.

In the CAC formalism, a crystalline material is viewed as a union of several continuous domains of lattice cells, each domain separated from another via a discontinuous transition, with a group of discrete atoms embedded within each lattice cell; additionally, the union contains particles in areas where a continuous description of densities is impractical or unnecessary. Each atom in the discrete basis of a given continuous domain of lattice cells experiences a gradual deformation from cell to cell, with respect to the same discrete atom in the basis of nearby unit cells, in that same continuous domain. Similar two-level descriptions are also employed in Micromorphic and other generalized continuum mechanics (GCM) theories [27–33], but CAC contrasts with these GCM theories in that the subscale description consists of discrete atoms. As with the Irving-Kirkwood formalism, the continuum balance laws may be derived, using instead a concurrent two-level description of the material, to provide a concurrent atomistic-continuum representation of the conservation laws [9]; the mathematical tool by which such equations are formulated is the theory of distributions [34]. The resulting conservation equations hold instantaneously and are valid at the atomic scale without fields needing to be continuous. Consequently, the time rate of change of any density, such as mass, momentum, and energy, of the α -th species per unit cell volume, $\mathbf{a}_\alpha(\mathbf{x}, t)$, at a point \mathbf{x} in space can be expressed in terms of two fluxes in the following form:

$$\frac{\partial}{\partial t} \mathbf{a}_\alpha(\mathbf{x}, t) = -\frac{1}{V} \oint_{\partial V} \mathbf{J}_1(\mathbf{x} + \mathbf{x}', t) \cdot \mathbf{n} d^2 x' - \frac{1}{V} \oint_{\partial V_\alpha} \mathbf{J}_2(\mathbf{x}, \mathbf{y}', t) \cdot \mathbf{n} d^2 y', \quad (1)$$

where \mathbf{J}_1 , is the flux across the bounding surface of the unit cell at \mathbf{x} , ∂V , and \mathbf{J}_2 the internal flux across ∂V_α (the bounding surface of atom α) within the unit cell V , cf. Fig. 3. This means the rate of change of the density at a given point in the physical space is due to all surrounding matter in all surrounding unit cells in addition to the other matter contained within the respective unit cell.

Using the mass conservation equation and the relationship between internal force and surface traction, the conservation law for linear momentum can be expressed in terms of the internal force density $\mathbf{f}_\alpha^{\text{int}}(\mathbf{x}, t)$ and the kinetic parts of momentum flux, which are related to kinetic temperature T , to grant:

$$\rho_\alpha \ddot{\mathbf{u}}_\alpha = \frac{1}{V} \oint_{\partial V} (\boldsymbol{\tau}_\alpha^{pot} + \boldsymbol{\tau}_\alpha^{kin}) dS + \frac{1}{V} \oint_{\partial V_\alpha} (\boldsymbol{\tau}_\alpha^{pot} + \boldsymbol{\tau}_\alpha^{kin}) dS_\alpha = \mathbf{f}_\alpha^{int}(\mathbf{x}, t) + \mathbf{f}_\alpha^T(\mathbf{x}, t), \quad \text{with } \frac{1}{t} \int_0^t \mathbf{f}_\alpha^T(\mathbf{x}, \tau) d\tau = \lambda \nabla_{\mathbf{x}} T(\mathbf{x}, t). \quad (2)$$

The conservation equations expressed in Eq. (2), supplemented by the underlying interatomic potential, can solve for either the exact or approximate atomic trajectories, depending on the elected resolution, since it solves for both the lattice deformation and individual atomic motion. At this stage, the chosen resolution may coarse-grain the smooth lattice deformation to reduce the volume of information. Thus, this method constitutes a Concurrent-Atomistic-Continuum (CAC) solution. Notable features of CAC include the following.

- (1) The same single set of governing equations govern both the atomistic and continuum regions; in the two limiting cases, namely the atomic and the macroscopic scales, the atomistic and continuum descriptions of transport processes are recovered.
- (2) There is no need for an artificial interface between atomistic-continuum descriptions that limits many multiscale methods to static phenomena.
- (3) The CAC formulation can be solved efficiently using continuum simulation approaches, such as the Finite Element Method (FEM) [35], with the only constitutive relation being the interatomic potential. In addition, due to its use of a nonlocal force field, continuity of the solution is not required; consequently, nucleation and propagation of dislocations or cracks can be simulated via the sliding and separation of finite elements as a direct consequence of the solution of the governing equations.
- (4) CAC can simulate complex crystalline materials and reproduce both acoustic and optical phonon branches, due to its incorporation of internal degrees of freedom, in the coarse-grained regions modeled using finite elements. With a trilinear shape function typically employed in FE simulations, CAC can reproduce accurate phonon dynamics for phonons whose wavelengths are longer than the element size; this feature is present with either a continuous or a discontinuous interpolation.
- (5) Although the use of CAC may result in error as a result of coarse graining, the convergence of the solution can be rigorously quantified by increasing the numerical resolution, since CAC utilizes no additional constitutive laws. The formalism guarantees that any answer must inevitably converge at the atomic resolution. Thus, the numerical procedure enables the exploration of emergent mesoscopic behaviors that would prove intractable to simulate with full atomistic resolution, whose description may converge with a coarse-grained representation. The error of this coarse-grained description can then be quantified.

In contrast with other continuum-based methods, in CAC it is only necessary to solve the balance equation of linear momentum numerically since the conservation equations of energy and momentum produce, in the theoretical sense, the same time evolution of the field variables. Thus, the weak form solution of the CAC momentum balance equation through the Galerkin method is expressed as:

$$\int_{\Omega} \varphi_n(\mathbf{X}) (\rho^\alpha \ddot{\mathbf{u}}^\alpha - \mathbf{f}_{int}^\alpha - \mathbf{f}_T^\alpha + \mathbf{f}_{ext}^\alpha) d\Omega_{\mathbf{X}} = 0, \quad (3)$$

where integration is carried out over the span, Ω , of the material coordinate, \mathbf{X} , in the chosen domain without any requirement of continuity on the deformation or its derivatives for evaluation [36]; note that this presentation utilizes a Lagrangian description of the integral for convenience. Additionally, $\varphi_n(\mathbf{x})$ represents the chosen basis of interpolation functions; these are also referred to as shape functions.

Such finite element solutions have been implemented in previous CAC algorithms: e.g., PyCAC [37] and other in-house versions found in previous publications [38,39]. However, these previous CAC parallel algorithms were not designed for the parallel solution of problems with nonuniform scales. Consequently, these predecessors are unsuitable for systems that involve large disparate morphological length scales. However, due to the structure of the discrete equations, Eq. (3), the numerical solution of a multiscale system can be decomposed and efficiently implemented in parallel.

Next, we explain why the CAC numerical solution is amenable to large-scale parallelization. Firstly, the matrix equations of the FEA problem can be formulated to be both banded and block diagonal, with the only coupling in the solution between finite elements (each of which spans its own piecewise subdomain $V_e \in \Omega$) arising through the nodal force vector; this is due to the use of a non-local force field defined by the interatomic potential. Since the solution is block diagonal, every finite element's nodal variables can be solved for in terms of its respective mass matrix \mathbf{M}_α , nodal accelerations $\ddot{\mathbf{U}}_\alpha$, and nodal force vector \mathbf{F}_α , where α represents the internal degree of freedom in the CAC two-level description. The N , number of elements and atoms in the system, discretized equations can thus be written as:

$$\mathbf{M}_\alpha \ddot{\mathbf{U}}_\alpha = \mathbf{F}_\alpha, \quad (4)$$

where each of the matrices \mathbf{M}_α and \mathbf{F}_α are the result of integration over one element's respective volume. Thus, these matrices take the form:

$$\mathbf{M}_\alpha = \iiint_{V_e} \rho_\alpha(\mathbf{x}) \begin{pmatrix} \varphi_1(\mathbf{x})\varphi_1(\mathbf{x}) & \dots & \varphi(\mathbf{x})_1\varphi_n(\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \varphi(\mathbf{x})_n\varphi_1(\mathbf{x}) & \dots & \varphi_n(\mathbf{x})\varphi_n(\mathbf{x}) \end{pmatrix} dV \quad (5)$$

$$\mathbf{F}_\alpha = \iiint_{V_e} \mathbf{f}_{\text{int}_\alpha}(\mathbf{x}) \begin{pmatrix} \varphi_1(\mathbf{x}) \\ \vdots \\ \varphi_n(\mathbf{x}) \end{pmatrix} dV. \quad (6)$$

Here, the mass and force densities are expressed as unit cell volume averages of the Irving Kirkwood point densities. For a crystalline material these unit cell averages obey the following simple result:

$$\rho_\alpha(\mathbf{x}) = \frac{m_\alpha}{V} \quad \forall \mathbf{x} \in V_e \quad (7)$$

$$\mathbf{f}_{\text{int}_\alpha}(\mathbf{x}) = \frac{1}{V} \sum_{i=1}^{N_e} \mathbf{F}_{\alpha_i} B_i(\mathbf{x}) \quad \forall \mathbf{x} \in V_e, \quad (8)$$

where m_α is the mass of the particle associated with the internal degree of freedom α for that respective finite element and V is the undeformed unit cell volume. The integration domain is mapped from a crystalline representation of the finite element, Lagrangian description, so that the unit cell volume remains a constant in the equation throughout that finite element. While the distribution of mass over a finite element is simple to describe with the mapped coordinates, the volume averaged internal force density, $\mathbf{f}_{\text{int}_\alpha}(\mathbf{x})$, is more elaborate. In general, it obeys Eq. (8), where N_e is the number of unit cells in the element and \mathbf{F}_{α_i} is the particle force, computed via the interatomic potential, for particle α in the i th unit cell. The box function, $B_i(\mathbf{x})$, results from the unit cell averaging of the singular Irving Kirkwood point function and is defined as:

$$B_i(\mathbf{x}) = \begin{cases} 1 & \forall \mathbf{x} \in V_i \\ 0 & \text{otherwise} \end{cases}, \quad (9)$$

with V_i being the domain of the i th unit cell.

The variation of $\mathbf{f}_{\text{int}_\alpha}(\mathbf{x})$ is expected to be smooth across a single finite element; as a result we may sample Eq. (8) at several choice points using a quadrature scheme to approximate Eq. (6). The numerical solution in the case of atoms is identical to the process of MD; the only coupling between the finite element regions and atoms is through the computation of the non-local particle force. The numerical details of estimating the force vector, through quadrature, for finite elements are explained in Section 3.

With the solution of Eqs. (4), a numerical integrator such as Velocity Verlet is used to update the positions and velocities of all atoms and finite element nodes. The timestep provided to the integrator must satisfy a stability criterion in order to avoid a failed simulation; this stability criteria may be determined via the phonon dispersion curve, or in general an eigenvalue decomposition, of the problem. Recent modal analysis of the CAC numerical solution [40] indicates that for polyatomic (i.e., multicomponent) systems, the timescale may be restricted to that of MD; this is due to the possibility of high frequency content in optical phonon branches, even for large wavelengths, when there is more than one internal degree of freedom. In the case of monoatomic representations, the stable timestep scales with the finite element resolution; this is because the longitudinal branch of phonons behaves linearly for large wavelengths. Accordingly, in the form presented here, CAC should not be viewed as a multi-time scaling method.

It is noted that the surface integrals of the kinetic fluxes in Eq. (2), due to the difference between particle velocities and the velocity field represented by finite elements, are present in the conservation equation, i.e., $\mathbf{f}_\alpha^T(\mathbf{x}, t)$ in Eq. (3). The time average of this contribution is either zero or a constant within a finite element at equilibrium or steady state. However, the fine-scale fluctuations due to this term are not accounted for. For now, the CAC codes are aimed at phenomena and processes where fine-scale thermal fluctuations can be neglected.

To date, the accuracy of CAC has been tested through one-to-one comparisons with MD in space- and time-resolved simulations of crack nucleation and branching [41,42], phase transitions [43], dislocation nucleation and propagation [14,17,44–46], dislocation loop formations [15,16,47,48] dislocation interactions with interfaces and other defects [13,18–20,49–51], as well as phonon-dislocation [11,12], phonon-GB (grain boundary) [10] and phonon-internal surface interactions [52]. Thus, CAC enables the simulation of physical processes that involve interactions among phonons, interfaces, and mesoscale defect structures such as those found in hierarchical metamaterials or multiscale layered materials.

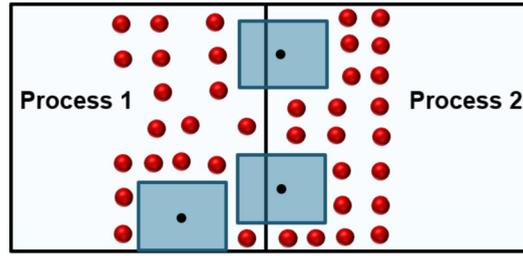


Fig. 4. Schematic Demonstrations (in 2D) of the assignment of atoms and finite elements in a spatially decomposed CAC problem. The red spheres are atoms and the blue boxes with black dots denote finite elements and their respective centroids. Elements are assigned to a process based on the location of their centroid. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

3. CAC in a parallel MD code

3.1. Decomposing the problem

The first step in solving this multiresolution problem in parallel requires us to decompose the problem of obtaining all the force vectors, particle forces, and resulting accelerations, into a set of smaller problems. The chosen methodology to achieve this is the spatial decomposition method, such as the one used in LAMMPS [21]. This method has the benefit of an efficient ratio of force calculation cost to communication cost. In the case of an MD spatial decomposition algorithm, assignment of individual atoms to each computing process is determined by the particle's position; if the particle is in the region (portion of simulation domain) assigned to a process, then that process is responsible for computing the forces, constraints, etc. on that particle.

In the case of CAC, the decomposition must assign both particles and finite elements. Since the finite elements are extended objects, they may not lie uniquely in any of the regions. We assign a finite element to a process if its centroid lies within its respective region; this is depicted in Fig. 4. Once a finite element is assigned to a process, that process will allocate and retain all the relevant information associated with that element: nodal positions, nodal velocities, etc.

Generally, MD simulations are run with each process being assigned equally sized regions. If the particle density is roughly uniform, such simulations will scale as $O(N/P)$, where N is the total number of particles in the model and P is the number of processes. A CAC model, however, will not, in general, have a homogeneous distribution of finite element centroids and particles. As a result, a load balancing strategy is needed whereby the regions assigned to each process are adjusted so that they are not all of equal size; this is discussed later in Section 3.5.

3.2. Computing forces

In section 2, a brief overview of the finite element solution process was provided. It was determined that the FE force vector in Eq. (6) was the remaining aspect of the calculation that required a numerical method. The algorithm to address this is summarized as follows:

Algorithm 1 Computing Forces.

Input: N local finite element and atom data (positions, velocities, etc.)

Output: N local finite element and atom forces.

- 1: **for** $i = 1$ to N **do**
 - 2: Determine if i is an atom or element.
 - 3: If i is an atom get \mathbf{f}_i and go to step 6, otherwise nodal forces $= \int_{\Omega_i(\mathbf{x})} N_j(\mathbf{X}) f_1^\alpha(\mathbf{X}) dV_{\mathbf{x}}$.
 - 4: Decompose integral domain into sectors: $\int_{\Omega_i(\mathbf{x})} N_j(\mathbf{X}) f_1^\alpha(\mathbf{X}) dV_{\mathbf{x}} = \sum_{s=1}^n \int_{\Omega_{i_s}^s(\mathbf{x})} N_j(\mathbf{X}) f_1^\alpha(\mathbf{X}) dV_{\mathbf{x}}$.
 - 5: Apply quadrature rule to each sector $\int_{\Omega_{i_s}^s(\mathbf{x})} N_j(\mathbf{X}) f_{1\alpha}(\mathbf{X}) dV_{\mathbf{x}} = \sum_{q=0}^r w_q f_{1\alpha}(\mathbf{x}_{iq}) N_j(\mathbf{x}_{iq})$.
 - 6: Compute force field at $f_{1\alpha}(\mathbf{x}_{iq})$, or particle force $\mathbf{f}_i = \int_V f_{1\alpha} \delta(\mathbf{x} - \mathbf{x}_{iq}) dV = \int_V f_{1\alpha}(\mathbf{x}_{iq}) dV$ with $\alpha = 1, q = 1$ as convention, by accessing K atom and virtual atom neighbors for each point \mathbf{x}_{iq} . See Fig. 5 for a depiction. See Algorithm 2 for details on getting the K neighbors.
 - 7: **for** $j = 1$ to K **do**
 - 8: If $\|\mathbf{x}_j - \mathbf{x}_{iq}\| < r_{cut}$, the cutoff radius, add the force contribution to $f_{1\alpha}(\mathbf{x}_{iq})$.
 - 9: **end for**
 - 10: **end for**
-

Note that step 4 of Algorithm 1 decomposes the integral over an element into an interior, surface regions, edge regions, and a set of corner regions; where the interior is the only one not expected to interact with other elements and atoms. In the case of surfaces, edges, and corners there is an associated thickness (not actually a surface or edge) that is chosen based on the force cutoff radius; which is used to decide how much of the element's material can interact with other elements and atoms.

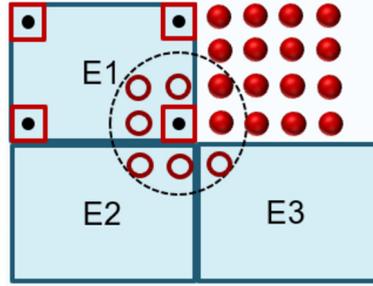


Fig. 5. Depiction of the process by which the non-local force density is computed at quadrature points (small boxes with a point) using real and virtual atoms within the cutoff radius (dashed circle). Real atoms are shown as filled circles while the virtual sampling points for material inside finite elements are shown as unfilled circles.

3.3. The neighbor listing process

Efficient computation of non-local forces in MD relies on some form of neighbor list data structure. This ensures that in $O(N/P)$ time one can identify the set of atoms that are within the force cutoff distance of each particle in the system. Similarly, the CAC algorithm needs an efficient method to find all the elements and atoms within the cutoff distance of each element and atom; this is described in Algorithm 2. Note that this algorithm uses the ASA CG [53] optimization code.

Algorithm 2 Neighbor Listing.

Input: N local finite element and atom data (positions, velocities, etc.)

Output: Neighbor list of elements/atoms for each atom or finite element quadrature abscissae

```

1: for  $i = 1$  to  $N$  do
2:   If  $i$  is an element define the set of  $Q$  quadrature abscissae,  $\mathbf{x}_{iq} \in [1, Q]$ .
3:   If  $i$  is an atom,  $\mathbf{x}_{iq}$  is the atom's position and  $Q = 1$ .
4:   for  $j = 1$  to  $Q$  do
5:     Access the stencil of  $B$  bins that  $\mathbf{x}_{iq}$  corresponds to.
6:     for  $l = 1$  to  $B$  do
7:       Access the  $K$  element and atom contents of bin  $l$ .
8:       for  $k = 1$  to  $K$  do
9:         If  $k$  is an atom, and  $\|\mathbf{x}_k - \mathbf{x}_{iq}\| < r_{cut}$ , add  $k$  to the neighbor list of  $\mathbf{x}_{iq}$  and skip to 16.
10:        If  $k$  is an element, define a bounding box for the element; expand the box edges by  $r_{cut}$ , see Fig. 6 for a depiction.
11:        If  $\mathbf{x}_{qi}$  lies in the expanded bounding box for  $k$ , use ASA CG to find closest point,  $\mathbf{x}_m$ , of element  $k$  to  $\mathbf{x}_{iq}$ .
12:        If  $\|\mathbf{x}_m - \mathbf{x}_{iq}\| < r_{cut}$ , construct a small set of  $V$  virtual atoms, using the finite element basis and  $\mathbf{x}_m$ , that must include all virtual atoms within  $r_{cut}$  of  $\mathbf{x}_{iq}$  and some possible extras.
13:        for  $v = 1$  to  $V$  do
14:          Add  $v$  to the neighbor list if  $\|\mathbf{x}_v - \mathbf{x}_{iq}\| < r_{cut}$ .
15:        end for
16:      end for
17:    If  $i$  is an element, add virtual atoms in its own domain satisfying  $\|\mathbf{x}_v - \mathbf{x}_{iq}\| < r_{cut}$  to the neighbor list of  $\mathbf{x}_{iq}$ .
18:  end for
19: end for
20: end for

```

In LAMMPS, the cutoff includes a small “skin” distance which allows the same list to be re-used for several timesteps until some atom or element has moved half the skin distance. In the case of finite elements, the maximum displacement of the interpolation field determines whether to trigger a rebuild of the neighbor list.

Algorithm 2 relies on a binning technique, similar to that used for MD simulations [21]. To summarize this technique, the simulation box containing all particles is decomposed into an ordered grid of axis-aligned rectangular “bins”; particles are then assigned a bin index depending on their position. A key difference with finite elements is the need to assign to every bin the set of finite elements that overlap with it. An element bounding box is used to produce a safe upper bound on the set of bins an element overlaps with. Determining overlap between a bin and an element bounding box is simple since all bins are themselves boxes within a uniform grid, see Fig. 7. Thus, with an $O(N/P)$ loop through the set of finite elements and atoms every bin is aware of which particles it contains and which finite element bounding boxes overlap with it. Additionally, the desired quadrature integration points may be assigned their bin location just as is done for particle positions.

Due to the disparate length scales that may arise in different parts of the model, each process subdomain in the spatial decomposition receives its own grid of bins. The bin size in each process is selected according to a weighted average of the element sizes, if they are larger than the force cutoff; thus, if there are many large finite elements, locally, the chosen bin size will be of comparable size to the bounding boxes of these finite elements.

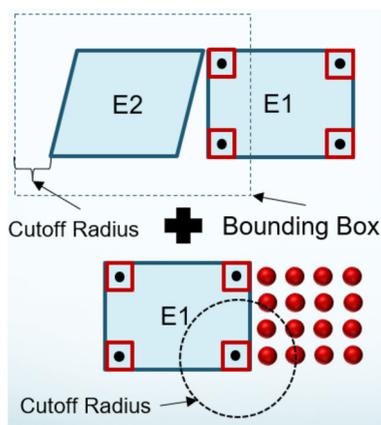


Fig. 6. The procedure for building the neighbor list of a finite element. Element E1 is adjacent to E2 and many atoms. The leftmost quadrature points of E1 include E2 as a neighbor since they lie inside E2's bounded box expanded by the cutoff radius (within the rectangular dotted box). The lower-left quadrature point of E1 includes a few atoms as neighbors (dotted circle).

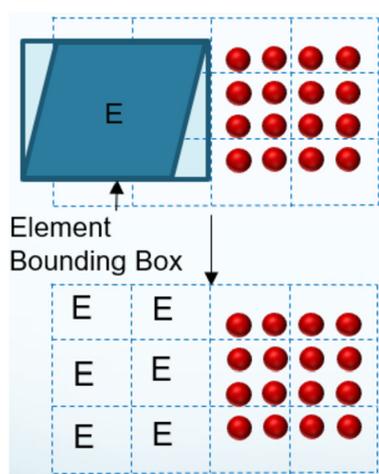


Fig. 7. Simple demonstration of the binning technique used to build neighbor lists for the CAC method. Point particles are assigned to a single bin. Finite elements are assigned to every bin with which their bounding box overlaps (shown with an E).

Once atoms, quadrature points, and elements are assigned to bins, the neighbor list for each atom and quadrature point can be built efficiently. For each point and the bin it resides in, the code loops over a stencil of bins that surrounds that point's bin. The stencil includes the center bin itself and only the nearby bins which are within the cutoff distance of the center bin; this stencil is accessed in step 5 of Algorithm 2.

3.4. Communication

One remaining issue is how to build the neighbor list and compute pairwise interactions in parallel for a spatially decomposed problem. In LAMMPS, each process region is an axis-aligned rectilinear or triclinic box. To compute forces for all the particles and finite elements a process owns, it needs ghost copies of nearby particles and elements (owned by other processes). Specifically, it needs any particles or elements within the force cutoff distance of any of its particles or elements. For an MD-only model this is simply the halo of particles within a cutoff distance of the process sub-box.

For CAC, the communication is more complicated. Since finite elements have a finite size, a single element can overlap with the boxes of several processes; though it is owned by the process whose box the element centroid is in. The method to address this is depicted in Algorithm 3 and in Fig. 8 as well.

3.5. Load balancing

All stages of the calculation have now been described for the spatially decomposed CAC algorithm. However, for systems with a mixture of atoms and finite elements or varying sizes, the spatial decomposition of the domain should ideally be chosen so that the computational load across processes is balanced. The existing spatial decomposition algorithms in

Algorithm 3 Communication.**Input:** N local finite element and atom data (positions, velocities, etc.)**Output:** finite element and atom data, belonging to neighboring processes, needed for forces.

```

1: for  $i = 1$  to N do
2:   If  $i$  is an element define a bounding box for the element.
3:   If the bounding box for  $i$  overlaps with the bounding box of a neighboring process,  $p \in [1, P_{neigh}]$ , send the bounding box data corresponding to  $i$  as a message to  $p$ .
4: end for
5: for  $p = 1$  to  $P_{neigh}$  do
6:   Receive messages for all  $M_p \in M$  overlapping element bounding boxes from neighboring proc  $p$ 
7: end for
8: for  $m = 1$  to M do
9:   Expand the  $m$ th ghost bounding box by  $r_{cut}$  in all directions.
10:  Assign the expanded element bounding box  $m$  to the bins it overlaps with.
11: end for
12: for  $i = 1$  to N do
13:  Assign element/atom  $i$  to its respective bins/bin.
14: end for
15: for  $i = 1$  to N do
16:  for  $p = 1$  to  $P_{neigh}$  do
17:    Test if the atom position, or element bounding box, for  $i$  overlaps with a halo, of depth  $r_{cut}$ , around the box of process  $p$ . If yes send the element/atom data for  $i$  to  $p$ .
18:  end for
19: end for
20: for  $m = 1$  to M do
21:  Access the K bins that the expanded ghost box  $m$ , sent by a neighboring proc  $p$ , overlaps with.
22:  for  $k = 1$  to K do
23:    Access the L atoms and elements contained in bin  $k$ .
24:    for  $l = 1$  to L do
25:      If atom/element  $l$  overlaps with the expanded ghost box  $m$ , send  $l$  to the process  $p$ , that sent the  $m$ th bounding box, to become a ghost of process  $p$ . This is depicted in Fig. 8.
26:    end for
27:  end for
28: end for

```

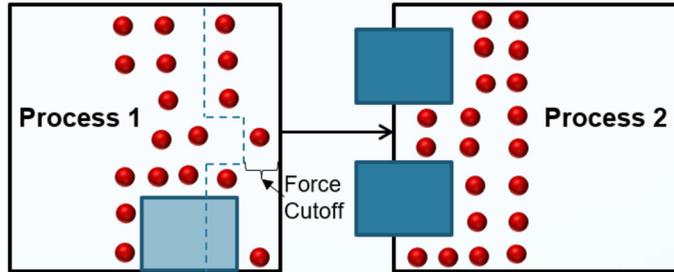


Fig. 8. A simplified example of the communication algorithm in two dimensions. The domain of Process 2 is first expanded by the cutoff radius, so Process 1 can identify particles or elements it owns which must be sent as ghost entities to Process 2. Process 1 has two such particles (rightmost) and no elements that meet this criterion. Next, the two finite elements owned by Process 2 are communicated to Process 1 because they overlap Process 1's region. Any additional atoms and elements owned by Process 1 which are less than the cutoff distance from a received element also need to be sent to Process 2. This constitutes four additional particles and one element. The effective boundary for sending particles and elements from Process 1 to Process 2 is denoted by the dashed line. Process 2 follows the same algorithm to determine what particles and elements to send as ghost entities to Process 1.

LAMMPS, particularly the RCB (recursive coordinate bisection) method [54], illustrated in the rightmost frame of Fig. 9, can assign atoms and elements in a load balanced manner so long as appropriate computational weights are assigned to both elements and atoms. These weights are chosen based on the expected expense of the force calculation, which is the majority of the computational expense for typical simulations. The resulting simulation will then, ideally, scale as $O(N/P)$, N being the total computational weight of the model and P being the number of processes. Initially, the computational weight of each finite element is the number of quadrature points associated with it. As the simulation proceeds, the computational weight of each element and atom is computed according to the expected number of virtual and real neighbors that its set of quadrature points must loop through; the number of neighbors is estimated using the previous neighbor list builds.

3.6. Performance

The first test of the CAC method's performance is its computational efficiency for an all-finite-element simulation compared to equivalent all-MD simulations. To that end, simple cubic blocks, using Cu and Si, were run for 5000 timesteps with both CAC and MD. A cubic block of PbTe was also run for 1000 timesteps using MD and CAC. For the CAC simulations, each

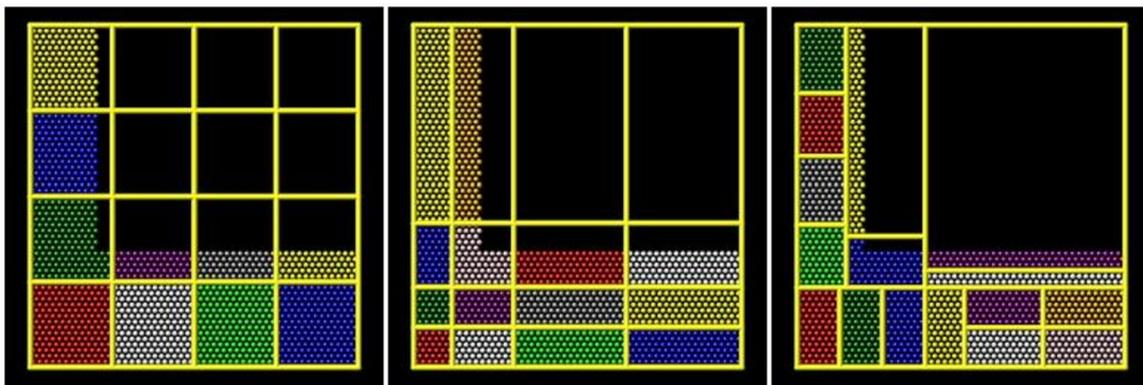


Fig. 9. Schematic demonstrations (in 2D) of the available LAMMPS spatial decomposition techniques [55]. The method best utilized for the multiple resolutions of the CAC method is the one shown furthest to the right, namely, the rcb (recursive coordinate bisection) method. Note that every subdomain in the decomposition scheme is a rectangle (rectangular prism in 3D).

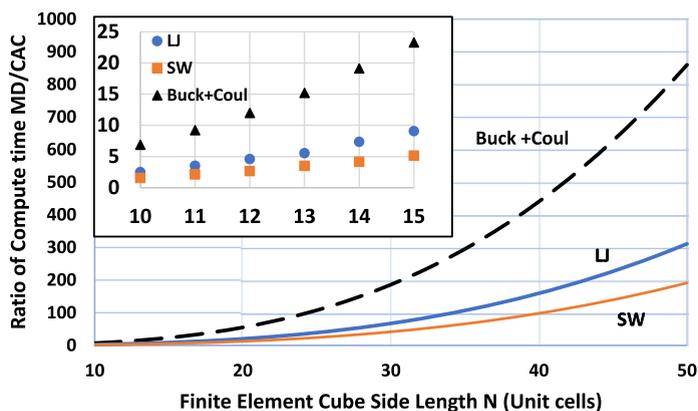


Fig. 10. Ratio of computational time between MD and CAC for the same blocks of crystalline Cu, Si, or PbTe described with their respective potentials; the efficiency increases with M^3 (the element size M), which indicates CAC can be very efficient for mesoscales with a low defect density, e.g., with large dislocation spacing. The first six points of each curve are the result of simulation data, shown on the inset graph, whereas the rest are a theoretical prediction, which is a cubic polynomial, predicting the efficiency after fitting to the first six data points. For all scales, CAC run time for 5000 steps of LJ and SW is approximately 180 s and 400 s; the Buck+Coul run time for 1000 steps is 610 s.

finite element was comprised of M by M by M unit cells; the total number of finite elements was held constant at 1000, but M was increased. Each element used 64 Gaussian quadrature points for the numerical integration. The resulting cubic-shaped finite elements had side lengths ranging from 3.6 nm to 38 nm. The corresponding MD models used the requisite number of atoms to obtain the same set of system sizes for the comparison; the number of atoms corresponding to these models ranges from four million to one billion.

The block of material has free surfaces in all three dimensions (no boundary conditions) and is allowed to equilibrate dynamically. The Stillinger-Weber (SW) potential and the Buckingham/Coulomb hybrid potential, used to model Si and PbTe respectively, are the same potentials used in the verification benchmarks of Section 4. Each simulation is run on 64 Intel Xeon E5-2698 processors with 32 cores per node (1 MPI process per core). The performance results for the simulations are plotted in the Fig. 10 inset plot as a ratio of the CPU time for the MD-only run divided by the CAC model time. For the largest physical problem, this ratio ranged from 200x for Si to 900x for PbTe.

The CPU time for the MD simulations in each set scaled as N/P , where N is the number of atoms. Conversely, the CPU time for the CAC model scales with the number of quadrature points used in the calculation, which remains constant across all cases; note that this is the case when re-neighboring occurs infrequently. However, the number of atoms represented by the CAC models scales with M [3]. Thus, the ratio of computational efficiency theoretically grows as M [3] for this comparison; this is reflected in both the Fig. 10 simulation data (inset plot) and the theoretical extrapolation.

As seen in Fig. 10, CAC simulations using the LJ potential do not speed up as much as those using Buck+Coul. This is due to how cheap the LJ pairwise calculations are compared to the cost of finite element interpolation required per quadrature point. Note that the theoretical limit of the computational speedup due to CAC for pair potentials, when re-neighboring is not very frequent, is the total number of pairwise calculations in the equivalent MD model divided by the total number of pairwise calculations in the CAC model. When compared to MD this is approximately the number of atoms divided by half the number of quadrature points; the factor of two results from the use of Newton's second law in MD algorithms. This

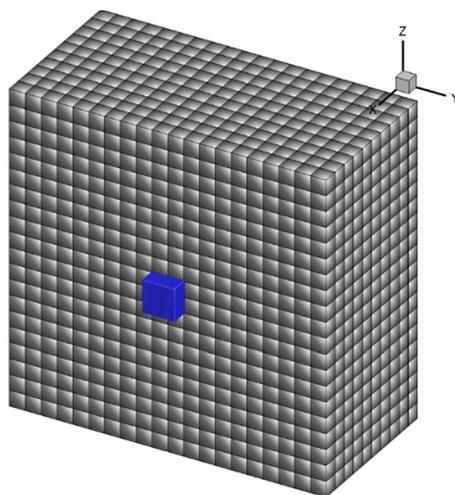


Fig. 11. The Cu cube model (72 nm by 72 nm by 72 nm) used in the sequence of performance runs. The smaller cube (shown blue) (7.2 nm by 7.2 nm by 7.2 nm) in the center of the entire cube is shown with a cross section through the middle of the cube about the x axis. Finite elements are rendered in gray.

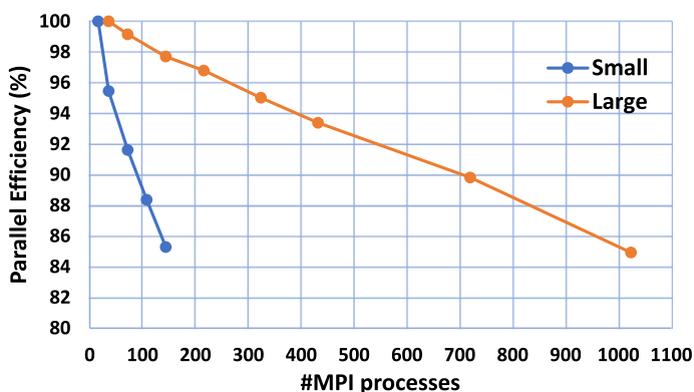


Fig. 12. Parallel efficiency as a function of the number of MPI processes (CPU cores) used to run the sample problem of relaxing a multiresolution cube of Copper. Both the small and large models have 30,000 atoms; the small model has 992 finite elements, the large has 7992. The large problem thus has 7.25 times the computational weight (quadrature points + atoms) of the small problem. Simulation efficiency degrades as more processors are used due to strong scaling effects for the two fixed-size problems. The last right point of the “small” and “large” curves corresponds, respectively, to an average of 1972 and 2027 atoms and quadrature points per process. The 100% efficient value is based on the run time on 16 processors for the small model and 36 for the large model. Simulations were run using an Intel Broadwell cluster at Sandia, with dual-CPU nodes and 36 cores/node.

theoretical limit is approached as the pairwise potential becomes more expensive relative to FE interpolation. For example, the result of the Buckingham/Coulomb hybrid potential shown in Fig. 10 is about 88% efficient compared to this theoretical metric; meanwhile, the LJ potential is 34% efficient.

It is worth mentioning that the in-house CAC codes developed by Chen et al. [39] show similar performance with uniformly meshed models. For example, if comparing their CAC code’s LJ speedup on two nodes (34 cores per node) consisting of Intel Xeon Phi 7250 processors, their speedup result is 2% greater than the results of Fig. 10. This was determined by running an LJ block with MD and CAC, using elements containing 10 by 10 by 10 unit cells, for 1000 steps on their code; the resulting MD simulation ran for 259 s compared to a 102 s CAC run. Given that the complexity of their force calculation loop scales with the same parameters as CAC, the number of quadrature points, this relative performance is expected for all element scales when neighboring is infrequent.

For many-body forces, the relative cost depends on the loop optimizations present in MD versus CAC. With the Stillinger-Weber (SW) potential, for example, there is a substantial loss due to the calculation of force density for finite elements requiring twice the cutoff radius relative to MD. The impact of this is reflected in the reduced coarse graining efficiency of SW relative to a potential such as LJ.

The next benchmarks demonstrate the CAC algorithm’s parallel efficiency. Two single crystal Copper (Cu) cubes, modeled with finite elements and an inner core of atoms, were created. One model (small) contains approximately 30,000 atoms and 1,000 finite elements; the geometry is depicted in Fig. 11. The other (large) contains 30,000 atoms and 7,000 finite elements. The purpose of using a small and a large model is to explore the relationship between efficiency and the average

computational weight, in atoms and quadrature points, per process. The interatomic potential used to model Cu interaction is the Embedded Atom Method (EAM) [56]. The simulations consist of dynamically simulating each cube towards equilibrium with no boundary conditions (free surfaces) for 5000 timesteps.

These models were run on different numbers of processors to measure the strong-scaling parallel efficiency. The results for both models are shown in Fig. 12. Both models are about 85% efficient when there are roughly 2000 atoms and quadrature points per MPI process. Thus, the dependence of the parallel efficiency on the average computational weight per process is evident.

4. Verification

4.1. Crack propagation and branching

A phenomenon that has been found to occur in nano-scale simulations of fracture is the branching of the crack tip when it interacts with phonons reflected by the model boundaries [41,57]. This result demonstrates the complicated nature of the coupling between fracture and reflected phonons and the capability of the CAC code to allow defects from an atomically resolved region to propagate into a coarsely discretized finite element region. The phenomenon of phonon-defect interaction is likely to occur when the boundary is a material interface, such as a grain boundary or an artificially constructed surface that may scatter and reflect phonons. This contrasts with reflection by the model boundary in numerical simulation. Since this phenomenon can be observed in MD simulation and can be an artifact of small period model size, it provides a convenient benchmark for the CAC method.

On that note, a simulation of crack growth in Si was conducted using both a CAC and MD model. The dimensions of this model are 280 nm by 550 nm by 2 nm; the CAC model is composed of 794 K atoms and 16 K triclinic elements (each of which spans 8, 8, and 4 unit cells respectively for each of the three triclinic directions). The MD model is composed of 9.2 million atoms. The potential used to model the fracture behavior is the Stillinger-Weber potential developed by Holland and Marder [58,59]. The results of the fracture process in both models are shown in Fig. 13. We note that the crack features are nearly identical in both representations. Observe that the crack branches in both the CAC and MD simulations. Additionally, the discontinuous finite element interpolation enables the CAC simulation to continue the crack propagation into the coarse-scale FE region, as shown in the rightmost time frame of Fig. 13.

The averaged stress of the entire system was then measured using the virial stress formula in both simulations to provide a quantitative comparison for benchmark purposes, shown in Fig. 14. It is worth noting that the virial stress formula is valid for quantifying the total or averaged stress for the entire system; since the fracture process is a transient process, it offers an approximate description of the local stress, which amounts to assuming a homogeneous stress distribution in the system [60–62]. Nevertheless, the virial stress can still be used to compare between MD and CAC since it is a quantity that is straightforward to calculate in MD and can also be calculated in CAC. The last point of the stress measurements corresponds to the last time frame in Fig. 13.

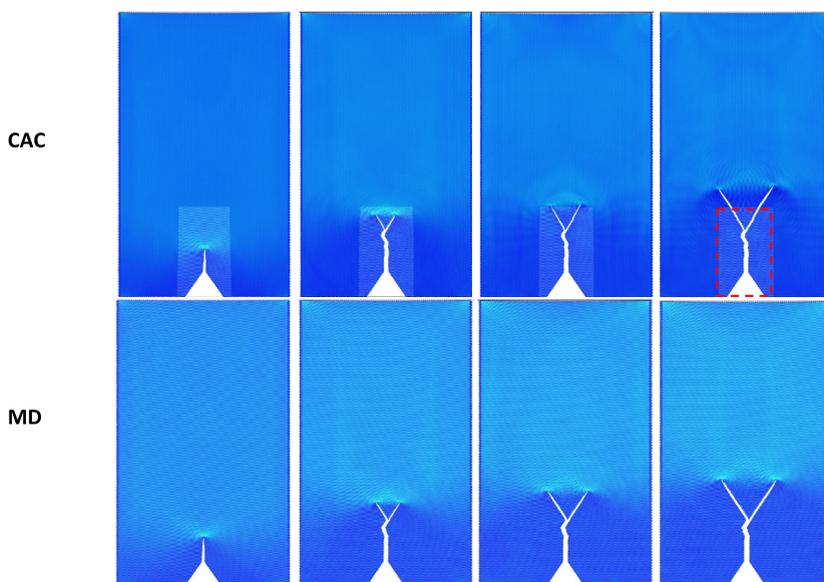


Fig. 13. Snapshots in time, 120 ps, 160 ps, 170 ps, and 180 ps in order, of the crack propagation simulation using both CAC (top row) and MD (bottom row) simulation models. The atoms in the CAC model are shown with the lighter shaded region; the last time frame explicitly shows the atomic region with the red dashed line. The contour plots proceed in time towards the right and the color displays the normal virial stress calculation along the loading direction, x , ranging from -1.5 GPa (blue) to 30 GPa (greenest).

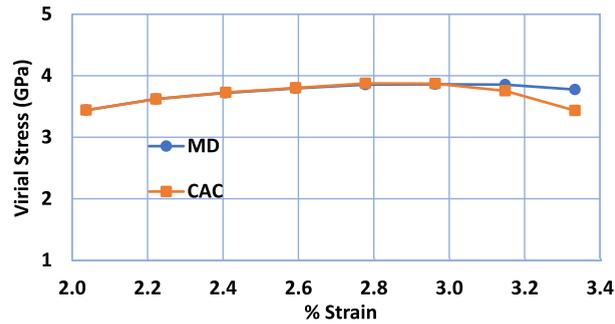


Fig. 14. Averaged stress of the entire crack simulation model for both CAC and MD throughout the crack propagation process. The difference in stress measurements begins to occur once the crack propagates across the numerical interface.



Fig. 15. Rendering of the periodic array of square holes used to define the Si phonic crystal used in the verification benchmark; the rendering is performed by Paraview [2].

4.2. Phonon Transport in a Si phonic crystal

A long standing challenge faced by multiscale methods is spurious wave reflection at the numerical interfaces (transitioning between atomic and coarse-grained resolution) of multi-scale models [8]. Even for phonons with wavelengths several times greater than the coarse-grained resolution scale, this effect can be observed due to the different governing equations present in the atomic and coarse-grained domains [8]. However, the formulation of the CAC method enables the propagation of long wavelength phonons, for all branches of the dispersion curve, even within the coarse-grained description. Thus, one verification of the methodology is to simulate the propagation of a wave packet chosen to excite phonons predominantly considered coarse with respect to the mesh resolution, i.e. excited modes correspond to a wavelength larger than 8 element lengths [38,40].

In order to verify the accuracy of wave propagation in the LAMMPS CAC code, a Si metamaterial with a periodic array of square holes, using a common Stillinger-Weber potential for Silicon [63], is constructed with dimensions of $1.4 \mu\text{m}$ by $1.4 \mu\text{m}$ by 1 nm in the (100) plane of Si and is discretized with 26000 finite elements (each of which is large enough to represent 4000 atoms in the fully resolved atomic model); the same geometry is then constructed for an MD simulation, with 107 million atoms. Such metamaterials have been observed to dramatically alter the thermal properties of the bulk material [64–66]. The square holes forming the periodic porous film of the model are each 17.4 nm by 17.4 nm ; the array is depicted in Fig. 15.

The model is loaded for 16 ps using an applied force in the center of the model, the shape of the force field is a cylindrically symmetric Morlet wavelet. Such local heating strategies have been used successfully in previous works for displacement and velocity loading [67,68]; this approach uses force loading with a similar function in order to reduce the discontinuity in the system response that can result from position or velocity loading. The dominant wavelength parameter of the Morlet wavelet is set to 139 nm , which is four times the artificial period length, i.e., $4p$. Once the loading has been completed the applied force is removed, and the system is simulated further to observe the kinetic energy propagation. The resulting kinetic energy distribution generates a pattern in both the CAC and MD models as depicted in Fig. 16. Note that the vast majority of the features present in the MD propagation pattern, which involves the scattering of phonons, are captured by the CAC model. Some differences emerge in the magnitude and distribution of kinetic energy; predominantly, this is due to the loading scheme exciting some fine scale properties, as discussed next.

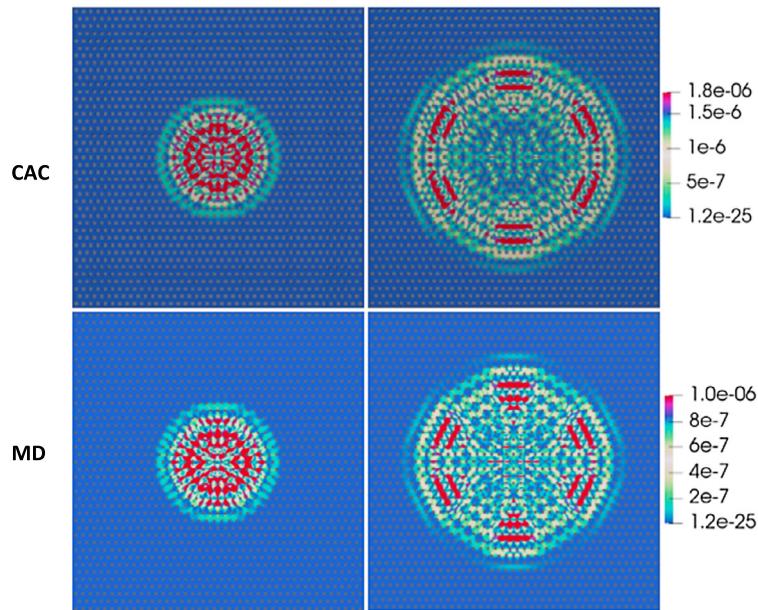


Fig. 16. Wave propagation on the (001) plane of a Silicon Metamaterial. The model is a thin plate with periodic boundary conditions along $\langle 001 \rangle$. CAC results are presented on the top row while MD results appear on the bottom row; left column images represent 20 ps of propagation while the right column represents the propagation at 60 ps. Color bars do not change with time.

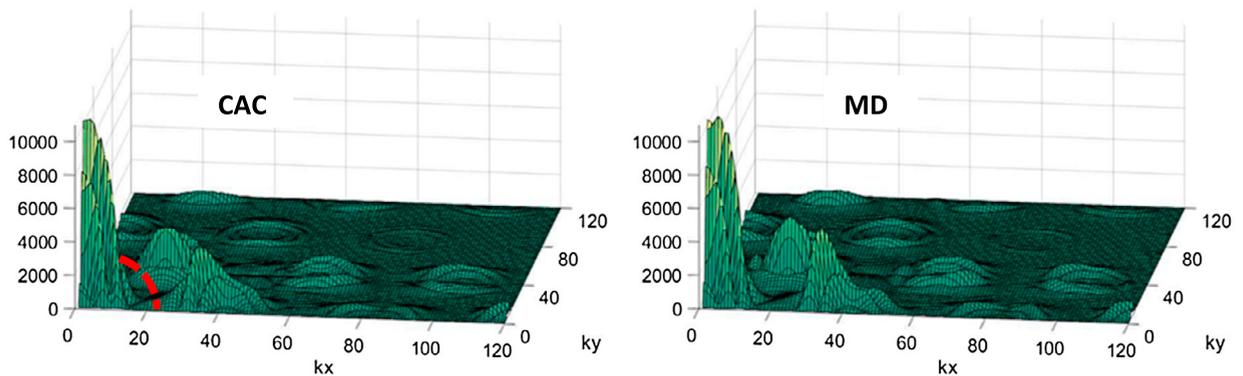


Fig. 17. The NFFT of the CAC (left) and MD (right) model displacements at 20 ps. The amplitudes have not been normalized. The results have almost exact ($<1\%$ error) agreement in the long wavelength limit; however, the two noticeable peaks outside of the long wavelength limit, denoted by the dashed arc on the CAC plot, are substantially taller (approximately 20% error) than the same peaks in the corresponding CAC result. Ranges of k_x and k_y are truncated for visibility of the prominent wave vectors.

To quantitatively compare the accuracy of the wave propagation in CAC with respect to MD, the Non-Uniform Fourier Transform (NFFT) of the displacement data is performed, with respect to the modes of the corresponding single crystal silicon block with the same dimensions as the metamaterial model, in order to observe the system's response in reciprocal space. Note that although these are not the eigenmodes of the metamaterial system, they provide a basis to quantify the phonon content of each simulation. In order to match NFFT samples the CAC data is interpolated from the nodal variables. The results shown in Fig. 17 demonstrate that the CAC model accurately captures the long wavelength phonons present in the MD simulation; differences begin to emerge as the wave vector increases past a typical rule of thumb limit for resolving waves with finite elements; typically, eight element lengths is the smallest accurate wavelength [38,40]. Past this critical limit, the error in the phonon content begins to grow; for example, the two peaks in Fig. 17 with wave vectors beyond the red dashed line, denoting the critical cutoff, exhibit approximately 20-30% error in amplitude between the two model representations. Naturally, this difference in representing the excitation can be ameliorated by simulating a larger metamaterial with a larger Morlet wavelet, i.e. a larger dominant wavelength. However, for the purpose of this verification, simulating a commensurate MD model for such a large system is impractical. Note that the information in the NFFT plot is very close between the two representations because it is computed from displacement data whereas the propagation pattern denotes kinetic energy; the errors in the fine scale modal information are thus amplified since kinetic energy scales with the square of wave frequency.

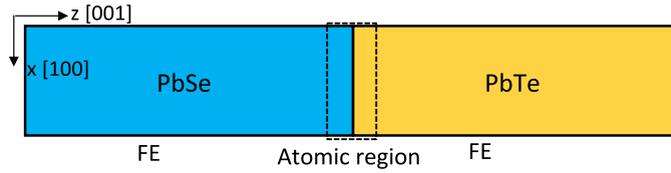


Fig. 18. A 2D schematic of the CAC model of a PbTe/PbSe heterostructure with a semi-coherent interface. The computer model has the dimensions of $23 \text{ nm} \times 23 \text{ nm} \times 600 \text{ nm}$ and is represented by 438400 atoms and 2328 elements. The corresponding MD model contains 10.6 million atoms.

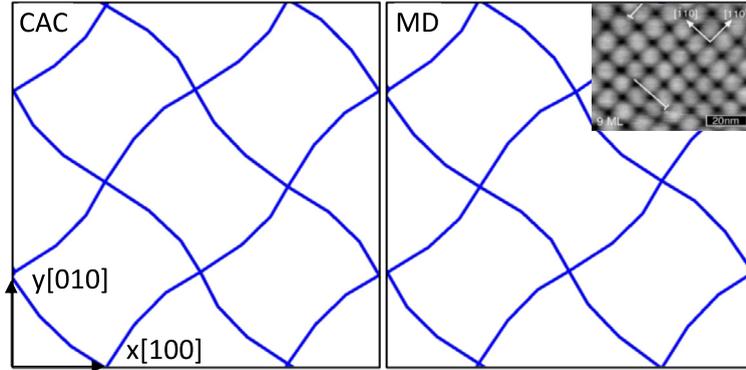


Fig. 19. The atomic structures of the PbTe/PbSe (001) interface in CAC (left) and MD (right) models. The interface structure is analyzed using DXA [77] to plot dislocation lines. The blue lines denote $1/2\langle 110 \rangle$ edge dislocations. Inset: STM imagery of a misfit dislocation network at a PbTe/PbSe (001) interface [73].

4.3. Phonon transport and interaction with misfit dislocations

The interaction between phonons and defects, such as dislocations, is a challenging problem that must be addressed in order to predict transient processes in complex materials [69,70]. MD is one of the few tools that has shed light on the mechanisms of defects interacting with phonons at the nanoscale [71,72]; however, it is currently incapable of addressing mesoscale processes that involve many defects interacting with phonons. Enabling the ability to simulate mesoscale transient processes involving defects and phonons is a core objective of the CAC method [11,12,17].

A verification of this ability is performed through the simulation of a phonon pulse propagating across the phase interface in a PbTe/PbSe heterostructure, for which misfit dislocation networks have been experimentally observed [73,74] and are theorized as the mechanism responsible for the reduced thermal conductivity of superlattices [75]. A schematic of the CAC model for this simulation is depicted in Fig. 18 A 2D schematic of the CAC model of a PbTe/PbSe heterostructure with a semi-coherent interface. The computer model has the dimensions of $23 \text{ nm} \times 23 \text{ nm} \times 600 \text{ nm}$ and is represented by 438400 atoms and 2328 elements. The corresponding MD model contains 10.6 million atoms. The computer model has the dimensions of $23 \text{ nm} \times 23 \text{ nm} \times 600 \text{ nm}$. The interface regions are atomically resolved with 438400 atoms and the rest of the model is discretized into 2328 elements. The corresponding MD model contains 10.6 million atoms. Thus, the CAC model reduces 94.4% of the degrees of freedom of the underlying atomistic model. In order to obtain the misfit dislocation structure at the interface, the direct wafer bonding process is simulated to bond the PbTe and PbSe single crystals, which is then followed by a series of simulations of the annealing process of the bonded structure. A Coulomb/Buckingham hybrid potential is used to model the interaction between atoms [76].

In Fig. 19, we present CAC and MD simulation results of the misfit dislocation network at the PbTe/PbSe (001) interface and compare them with an experimental image of the dislocation network of this interface. As can be seen from Fig. 19, the atomic structures of the interfaces in the CAC and MD models are almost identical. Both of them reproduce the structure of the dislocation network with the average dislocation spacing being 8.2 nm, in good agreement with the experimental measurements of 8.8 nm at room temperature [73].

To quantify and visualize the interaction between phonons and the misfit dislocation network at the PbTe/PbSe (001) interface, a transverse acoustic phonon wave packet that centers at a wavelength of 12.1 nm and a frequency of 0.115 THz is constructed according to the method of ultrafast phonon pulses [67,68]. Snapshots of the displacement responses of the heterostructure to the phonon pulse simulated by both MD and CAC are presented in Fig. 20. It is seen from Fig. 20 that, at $t=120 \text{ ps}$, the dislocation network vibrates strongly with a vibration amplitude of $7.2U$ and $7.5U$ in CAC and MD simulations, respectively, where $U=0.001\text{\AA}$ is the amplitude of the input wave packet. This is a demonstration that a resonance interaction between the phonon pulse and the misfit dislocation network has taken place. The waves that are propagating away from the interface at $t=240 \text{ ps}$ are waves reflected by the interface. The energy transmission coefficients obtained from the

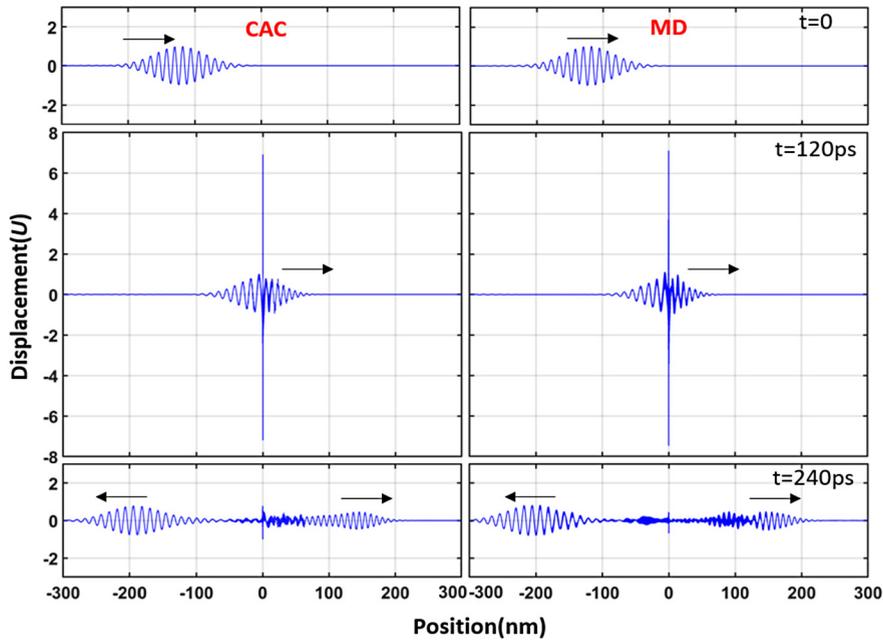


Fig. 20. Snapshots of the displacement response of a PbTe/PbSe heterostructure to a transverse phonon pulse that centers at a wavelength of 12.1 nm to compare CAC and MD simulation results, showing partial transmission and partial reflection of the wave packet at the interface (position 0) at $t=240$ ps, as well as the resonance interaction between phonons and misfit dislocations at $t=120$ ps. The amplitude of the input wave packet is $U=0.001\text{\AA}$.

CAC and MD simulations are 29.4% and 28%, respectively. These comparisons demonstrate a good agreement between CAC and MD in simulating phonon scattering by interface misfit dislocations both qualitatively and quantitatively.

5. Demonstration

In order to demonstrate the parallel algorithm in a situation where multiple resolutions are used, we simulate a PbTe/PbSe (001) heterostructure that is considerable larger than the one in Section 4.3. The computer model is presented in Fig. 21. It has dimensions of 116 nm by 116 nm by 160 nm and is discretized into 1100 8-node finite elements, with each node having four internal degrees of freedom, and approximately 545,000 atoms. The cartesian coordinate system in Fig. 21 is rotated such that the dislocation lines will span along the x and y axes. A MD model of this structure would contain approximately 70 million atoms. The system is periodic along the x and y axes and leaves the z-axis boundary free. The equilibrium structure of the interface is obtained through simulations of the direct wafer bonding process that bond PbTe and PbSe crystals. The misfit dislocation network obtained in the simulations is shown in Fig. 21.

The performance of the CAC algorithm as it equilibrates the interface model can be determined by analyzing the balance in computational time between processes for each of the code's primary operations. The inefficiency in distributing the computation time is quantified in Table 1 using the following definition:

$$f = 100 \cdot \frac{\sup |t_i - t_{avg}|}{t_{avg}}, \quad (10)$$

where t_i denotes the set of CPU times reported by all processes and t_{avg} is the sum of all the CPU times (minus synchronization time) divided by the number of processes; which approximates the ideal load balanced value. Note that there is a less than 13% imbalance in the force computation, which is the portion of the computation for which the load balancing algorithm assigns weights, for a model where the density of degrees of freedom changes 10000 fold between the coarsest resolution and the atomic resolution. However, the other portions of the algorithm that are incurred somewhat frequently when relaxing this interface model such as communication, reneighboring, and other modifications such as viscous dampening encounter more severe imbalances; due to the lack of assigned weights for these portions of the computation such that they can simultaneously be load balanced whilst balancing the force computation. In this case, the force computation comprised 80% of the computational time; since the equilibration process requires frequent reneighboring due to the temperature at the interface.

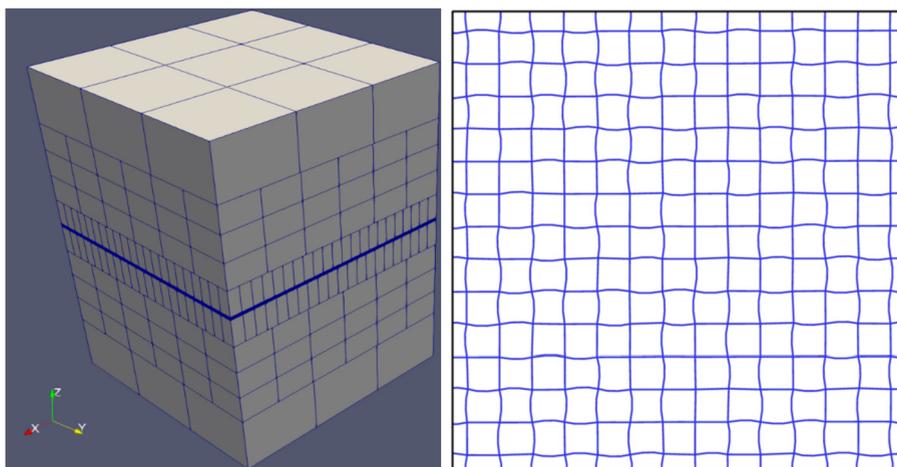


Fig. 21. Left: Depiction of the multi-resolution PbTe/PbSe model on Paraview. The model uses four resolutions; the smallest resolution corresponds to the atoms in the middle where the dislocations are expected to form. Right: DXA analysis result of the PbTe/PbSe interface for the multi-resolution model. The average dislocation spacing obtained in the simulation is 8.2 nm, in close agreement with experimental results [73].

Table 1

Run times (first row) and percent inefficiencies, per simulation algorithm, encountered in the simulation of the multi-resolution misfit dislocation model for 5000 time steps. Percent inefficiency is calculated with Equation (10).

Process Count	32	48	64
Total Loop Time	801 m	640 m	504 m
Force Calculation	6.16%	11.19%	12.35%
Neighbor Listing	87.2%	86.4%	228.8%
Communication	90.9%	84.6%	88.14%
Modification	97.3%	90.1%	92.10%

6. Summary and discussions

This paper has presented a massively parallel algorithm to implement the Concurrent Atomistic Continuum (CAC) method in Molecular Dynamics (MD) software and demonstrated an implementation in LAMMPS. The spatial decomposition algorithm described in this paper is one of the first to exhibit $O(N/P)$ scaling when the solution involves multiple scales of finite elements, discrete particles, and their interaction through non-local forces. The algorithm addresses these aims via the design of a halo communication strategy that accommodates finite, not point-like, objects with disparate resolutions; devising a multi-scale neighbor listing algorithm for non-local systems with extended particles, of multiple length scales, and atoms; creating an integration scheme to numerically solve the nonlocal governing equation, using FEM, for said systems; and efficiently decomposing the numerical solution of the resulting integral equations amongst computational processes using the LAMMPS rcb algorithm. The details of this process are depicted in Section 3 and shown to exhibit $O(N/P)$ scaling in Section 3.6. The computational efficiency of the CAC method in comparison with MD, achieved by coarse graining, has also been exhibited in Section 3.6 as a function of finite element length; as expected, the larger the average scale of the finite elements present in the model the greater the computational efficiency compared to an atomically resolved model.

Furthermore, it was demonstrated through benchmarking simulations of crack branching, phonon propagation in a Si phononic crystal, and phonon-misfit dislocation interaction that the CAC method in LAMMPS is capable of simulating complex nonequilibrium processes. It is recognized that coarse-graining error exists relative to MD for certain details such as kinetic energy density. These examples, both of uniform and multiple resolution, have demonstrated the ability of the CAC method to capture phonon and defect dynamics in both the atomistic and coarse-grained domains. Many previous multi-scale methodologies do not address many of the capabilities demonstrated in this work, especially simultaneously [7]. This provides a pathway for the effective simulation of complex mesoscale structures to better understand emergent dynamic behaviors.

The predictive power of classical interatomic interaction has proven very beneficial given the prohibitive computational cost and size limitations of quantum mechanical solutions [2,3]. However, MD will not be able to elucidate, likely for several decades [6], the phenomena encountered in mesoscale transport processes in complex material microstructures. The computational efficiency offered by CAC presents a viable method for the simulation of dynamic mesoscale transport processes. We expect this capability to enable the prediction of relationships between engineered microstructures and material performance.

We also expect that the algorithm will become more efficient over time as optimizations to both the single-core performance and the reduction in volume of communicated information lead to increases in overall computational efficiency. Regarding single-core efficiency, the current CAC algorithm possesses many loops that dominate the computational time; many of these could be vectorized, after refactoring the loops and memory access as required, by interfacing with an existing API such as OpenMP. There is also the potential for GPU acceleration using an API such as Kokkos [78]; which LAMMPS already exploits in other MD contexts.

In addition, algorithms to improve the accuracy (i.e., reduce coarse-graining error in modeling phenomena) and efficiency of the CAC method will be added in the near future. These include (1) adaptive refinement of the finite element interpolation that accommodates the need for changing resolution in dynamic simulations of defect propagation and multiplications, and (2) an algorithm to model the fine-scale fluctuations of the underlying atomistic information that is removed by the FE shape functions in coarse grained regions, i.e., $f_T^\alpha(x)$ in Eq. (3). Addressing the multiple time scales in $f_T^\alpha(x)$ could enable the efficient simulation of dynamically coupled thermal transport and defect dynamics at various temperatures.

Although in this work, CAC is presented as a method enabling concurrent atomistic and continuum simulation, the CAC formulation also provides consistent formulas for the calculation of continuum quantities, such as stress and heat flux, in atomistic or coarse-grained atomistic simulations of non-equilibrium processes in heterogeneous materials [60–62,79]. Currently, the majority of flux formulas implemented in MD software are only applicable to homogenized single-component materials, according to the restrictions specified by Irving and Kirkwood [22]. Future work will address a parallel implementation of the CAC flux formulas for both atomistic and coarse-grained simulations, thereby enabling the measurement of flux in non-equilibrium processes.

The CAC algorithm in LAMMPS was implemented as the “USER-CAC” package within the LAMMPS source code and will be publicly available in the GitHub version by the time this paper is published. In the eventuality of delays, it may be found at <https://github.com/Adrian-Diaz/lammps/tree/USER-CAC>.

CRediT authorship contribution statement

Adrian Diaz: Conceptualization, Methodology, Software, Validation, Formal Analysis, Writing

Boyang Gu: Methodology, Validation, Formal analysis, Software

Yang Li: Methodology, Validation, Formal analysis

Steven J. Plimpton: Supervision, Resources, Software

Younging Chen: Conceptualization, Resources, Supervision, Project Administration, Funding acquisition

David L. McDowell: Conceptualization, Project Administration, Funding acquisition

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This material is based on research supported by the US National Science Foundation under Award Numbers CMMI-1761512 (AD, YL and YC), CMMI 1761553 (DLM), and CMMI-2054607 (BG and YC). AD was also partially supported by the U.S. Department of Energy (DOE) Office of Science Graduate Student Research (SCGSR) program. The SCGSR program is administered by the Oak Ridge Institute for Science and Education (ORISE) for the DOE. ORISE is managed by ORAU under contract number DE-SC0014664. We thank Phan Thanh of Iowa State University for providing the LJ simulation data using their CAC code.

References

- [1] A. Ikesue, Polycrystalline nd: YAG ceramics lasers, *Opt. Mater.* 19 (2002) 183–187.
- [2] U. Ayachit, *The paraview guide: a parallel visualization application*, Kitware, Inc., 2015.
- [3] R.G. Parr, in: *Horizons of Quantum Chemistry*, Springer, 1980, pp. 5–15.
- [4] R.M. Martin, R.M. Martin, *Electronic Structure: Basic Theory and Practical Methods*, Cambridge University Press, 2004.
- [5] J.M. Haile, *Molecular Dynamics Simulation: Elementary Methods*, John Wiley & Sons, Inc., 1992.
- [6] J. Hemminger, G. Crabtree, J. Sarrao, From quanta to the continuum: Opportunities for Mesoscale Science, A Report from the Basic Energy Sciences Advisory Committee, Tech. Rep., 2012, pp. 1601–1606.
- [7] J. Dolbow, M. Khaleel, J. Mitchell, Multiscale mathematics initiative: a roadmap, US Department of Energy, Report No. PNNL-14966, 2004.
- [8] A. Diaz, D. McDowell, Y. Chen, in: *Generalized Models and Non-classical Approaches in Complex Materials*, vol. 2, Springer, 2018, pp. 55–77.
- [9] Y. Chen, S. Shabanov, D.L. McDowell, Concurrent atomistic-continuum modeling of crystalline materials, *J. Appl. Phys.* 126 (2019) 101101.
- [10] X. Chen, et al., Ballistic-diffusive phonon heat transport across grain boundaries, *Acta Mater.* 136 (2017) 355–365, <https://doi.org/10.1016/j.actamat.2017.06.054>.
- [11] X. Chen, L. Xiong, D.L. McDowell, Y. Chen, Effects of phonons on mobility of dislocations and dislocation arrays, *Scr. Mater.* 137 (2017) 22–26, <https://doi.org/10.1016/j.scriptamat.2017.04.033>.
- [12] L. Xiong, D.L. McDowell, Y. Chen, Sub-THz phonon drag on dislocations by coarse-grained atomistic simulations, *Int. J. Plast.* 55 (2014) 268–278.

- [13] L. Xiong, S. Xu, D.L. McDowell, Y. Chen, Concurrent atomistic–continuum simulations of dislocation–void interactions in fcc crystals, *Int. J. Plast.* 65 (2015) 33–42.
- [14] L. Xiong, G. Tucker, D.L. McDowell, Y. Chen, Coarse-grained atomistic simulation of dislocations, *J. Mech. Phys. Solids* 59 (2011) 160–177.
- [15] L. Xiong, Q. Deng, G.J. Tucker, D.L. McDowell, Y. Chen, Coarse-grained atomistic simulations of dislocations in Al, Ni and Cu crystals, *Int. J. Plast.* 38 (2012) 86–101, <https://doi.org/10.1016/j.ijplas.2012.05.002>.
- [16] L. Xiong, D.L. McDowell, Y. Chen, Nucleation and growth of dislocation loops in Cu, Al and Si by a concurrent atomistic-continuum method, *Scr. Mater.* 67 (2012) 633–636.
- [17] L. Xiong, et al., Coarse-grained elastodynamics of fast moving dislocations, *Acta Mater.* 104 (2016) 143–155.
- [18] S. Yang, Y. Chen, Concurrent atomistic and continuum simulation of bi-crystal strontium titanate with tilt grain boundary, *Proc. R. Soc. A, Math. Phys. Eng. Sci.* 471 (2015).
- [19] S. Yang, N. Zhang, Y. Chen, Concurrent atomistic–continuum simulation of polycrystalline strontium titanate, *Philos. Mag.* 95 (2015) 2697–2716.
- [20] S. Xu, L. Xiong, Y. Chen, D.L. McDowell, Validation of the concurrent atomistic-continuum method on screw dislocation/stacking fault interactions, *Crystals* 7 (2017) 120.
- [21] S. Plimpton, Fast parallel algorithms for short-range molecular dynamics, *J. Comput. Phys.* 117 (1995) 1–19, <https://doi.org/10.1006/jcph.1995.1039>.
- [22] J. Irving, J.G. Kirkwood, The statistical mechanical theory of transport processes. IV. The equations of hydrodynamics, *J. Chem. Phys.* 18 (1950) 817–829.
- [23] J.G. Kirkwood, The statistical mechanical theory of transport processes I. General theory, *J. Chem. Phys.* 14 (1946) 180–201, <https://doi.org/10.1063/1.1724117>.
- [24] Y. Chen, Reformulation of microscopic balance equations for multiscale materials modeling, *J. Chem. Phys.* 130 (2009) 134706.
- [25] Y. Chen, J. Lee, Atomistic formulation of a multiscale field theory for nano/micro solids, *Philos. Mag.* 85 (2005) 4095–4126.
- [26] Y. Chen, J. Zimmerman, A. Krivtsov, D.L. McDowell, Assessment of atomistic coarse-graining methods, *Int. J. Eng. Sci.* 49 (2011) 1337–1349.
- [27] A.C. Eringen, *Microcontinuum Field Theories: Foundations and Solids*, vol. 487, Springer New York, 1999.
- [28] Y. Chen, J.D. Lee, A. Eskandarian, Examining the physical foundation of continuum theories from the viewpoint of phonon dispersion relation, *Int. J. Eng. Sci.* 41 (2003) 61–83.
- [29] A.C. Eringen, *Mechanics of Micromorphic Continua*, Springer, 1968.
- [30] Y. Chen, J.D. Lee, Connecting molecular dynamics to micromorphic theory. (I). Instantaneous and averaged mechanical variables, *Physica A* 322 (2003) 359–376.
- [31] Y. Chen, J.D. Lee, Connecting molecular dynamics to micromorphic theory. (II). Balance laws, *Physica A* 322 (2003) 377–392.
- [32] Y. Chen, J. Lee, A. Eskandarian, Atomistic counterpart of micromorphic theory, *Acta Mech.* 161 (2003) 81–102.
- [33] Y. Chen, J.D. Lee, Determining material constants in micromorphic theory through phonon dispersion relations, *Int. J. Eng. Sci.* 41 (2003) 871–886.
- [34] L. Schwartz, *Mathématique*, I. d. Théorie des Distributions, vol. 2, Hermann, Paris, 1957.
- [35] S.S. Rao, *The Finite Element Method in Engineering*, Butterworth-Heinemann, 2017.
- [36] V.S. Vladimirov, *Equations of Mathematical Physics*, Moscow Izdatel. Nauka, 1976.
- [37] S. Xu, et al., PyCAC: the concurrent atomistic-continuum simulation environment, *J. Mater. Res.* 33 (2018) 857–871.
- [38] X. Chen, A. Diaz, L. Xiong, D.L. McDowell, Y. Chen, Passing waves from atomistic to continuum, *J. Comput. Phys.* 354 (2018) 393–402, <https://doi.org/10.1016/j.jcp.2017.10.038>.
- [39] H. Chen, et al., A spatial decomposition parallel algorithm for a concurrent atomistic-continuum simulator and its preliminary applications, *Comput. Mater. Sci.* 144 (2018) 1–10.
- [40] Y. Li, et al., Phonon spectrum and phonon focusing in coarse-grained atomistic simulations, *Comput. Mater. Sci.* 162 (2019) 21–32.
- [41] Q. Deng, L. Xiong, Y. Chen, Coarse-graining atomistic dynamics of brittle fracture by finite element method, *Int. J. Plast.* 26 (2010) 1402–1414.
- [42] Q. Deng, Y. Chen, A coarse-grained atomistic method for 3D dynamic fracture simulation, *Int. J. Multiscale Comput. Eng.* 11 (2013) 227–237.
- [43] L. Xiong, Y. Chen, Coarse-grained simulations of single-crystal silicon, *Model. Simul. Mater. Sci. Eng.* 17 (2009) 035002.
- [44] L. Xiong, Q. Deng, G. Tucker, D.L. McDowell, Y. Chen, A concurrent scheme for passing dislocations from atomistic to continuum domains, *Acta Mater.* 60 (2012) 899–913.
- [45] S. Yang, L. Xiong, Q. Deng, Y. Chen, Concurrent atomistic and continuum simulation of strontium titanate, *Acta Mater.* 61 (2013) 89–102.
- [46] K. Chu, A. Diaz, Y. Chen, T. Zhu, D.L. McDowell, Multiscale concurrent Atomistic-Continuum (CAC) modeling of multicomponent alloys, *Comput. Mater. Sci.* 201 (2022) 110873.
- [47] S. Xu, L. Xiong, Y. Chen, D.L. McDowell, Edge dislocations bowing out from a row of collinear obstacles in Al, *Scr. Mater.* 123 (2016) 135–139.
- [48] S. Xu, L. Xiong, Y. Chen, D.L. McDowell, An analysis of key characteristics of the Frank-Read source process in FCC metals, *J. Mech. Phys. Solids* 96 (2016) 460–476.
- [49] S. Xu, L. Xiong, Y. Chen, D.L. McDowell, Sequential slip transfer of mixed-character dislocations across $\Sigma 3$ coherent twin boundary in FCC metals: a concurrent atomistic-continuum study, *npj Comput. Mater.* 2 (2016) 15016.
- [50] S. Xu, L. Xiong, Y. Chen, D.L. McDowell, Shear stress- and line length-dependent screw dislocation cross-slip in FCC Ni, *Acta Mater.* 122 (2017) 412–419.
- [51] S. Yang, Y. Chen, in: *Multiscale Materials Modeling for Nanomechanics*, Springer International Publishing, 2016, pp. 261–296.
- [52] Y. Li, A. Diaz, X. Chen, D.L. McDowell, Y. Chen, Interference, scattering, and transmission of acoustic phonons in Si phononic crystals, *Acta Mater.* 224 (2022) 117481.
- [53] W.W. Hager, H. Zhang, A new active set algorithm for box constrained optimization, *SIAM J. Optim.* 17 (2006) 526–557.
- [54] M.J. Berger, S.H. Bokhari, A partitioning strategy for nonuniform problems on multiprocessors, *IEEE Trans. Comput.* 570 (580) (1987).
- [55] S. Plimpton, LAMMPS documentation, 2003.
- [56] M.S. Daw, M.I. Baskes, Embedded-atom method: derivation and application to impurities, surfaces, and other defects in metals, *Phys. Rev. B* 29 (1984) 6443–6453.
- [57] B.L. Holian, R. Ravelo, Fracture simulations using large-scale molecular dynamics, *Phys. Rev. B* 51 (1995) 11275.
- [58] J.A. Hauch, D. Holland, M.P. Marder, H.L. Swinney, Dynamic fracture in single crystal silicon, *Phys. Rev. Lett.* 82 (1999) 3823.
- [59] D. Holland, M. Marder, Cracks and atoms, *Adv. Mater.* 11 (1999) 793–806.
- [60] Y. Chen, A. Diaz, Local momentum and heat fluxes in transient transport processes and inhomogeneous systems, *Phys. Rev. E* 94 (2016) 053309.
- [61] Y. Chen, A. Diaz, Physical foundation and consistent formulation of atomic-level fluxes in transport processes, *Phys. Rev. E* 98 (2018) 052113.
- [62] A. Diaz, D. Davydov, Y. Chen, On the equivalence of the two foundational formulations for atomistic flux in inhomogeneous transport processes, *Proc. R. Soc. A* 475 (2019) 20180688.
- [63] F.H. Stillinger, T.A. Weber, Computer simulation of local order in condensed phases of silicon, *Phys. Rev. B* 31 (1985) 5262.
- [64] M. Maldovan, Sound and heat revolutions in phononics, *Nature* 503 (2013) 209–217, <https://doi.org/10.1038/nature12608>.
- [65] J.-K. Yu, S. Mitrovic, D. Tham, J. Varghese, J.R. Heath, Reduction of thermal conductivity in phononic nanomesh structures, *Nat. Nanotechnol.* 5 (2010) 718.
- [66] M. Maldovan, Phonon wave interference and thermal bandgap materials, *Nat. Mater.* 14 (2015) 667–674.
- [67] X. Chen, A. Chernatynskiy, L. Xiong, Y. Chen, A coherent phonon pulse model for transient phonon thermal transport, *Comput. Phys. Commun.* (2015).
- [68] P. Schelling, S. Phillpot, P. Keblinski, Phonon wave-packet dynamics at semiconductor interfaces by molecular-dynamics simulation, *Appl. Phys. Lett.* 80 (2002) 2484–2486.

- [69] T. Vreeland Jr, K. Jassby, Dislocation interactions with thermal phonons and conduction electrons, California Inst. of Tech., Pasadena. WM Keck Lab. of Engineering Materials, 1972.
- [70] A. Maurel, V. Pagneux, F. Barra, F. Lund, Interaction of a surface wave with a dislocation, *Phys. Rev. B* 75 (2007) 224112, <https://doi.org/10.1103/PhysRevB.75.224112>.
- [71] B. Deng, A. Chernatynskiy, P. Shukla, S.B. Sinnott, S.R. Phillpot, Effects of edge dislocations on thermal transport in UO₂, *J. Nucl. Mater.* 434 (2013) 203–209.
- [72] Z. Zheng, et al., Phonon thermal transport through tilt grain boundaries in strontium titanate, *J. Appl. Phys.* 116 (2014) 073706.
- [73] G. Springholz, K. Wiesauer, Nanoscale dislocation patterning in PbTe/PbSe (001) lattice-mismatched heteroepitaxy, *Phys. Rev. Lett.* 88 (2001) 015507.
- [74] K. Wiesauer, G. Springholz, Critical thickness and strain relaxation in high-misfit heteroepitaxial systems: PbTe 1– x Se x on PbSe (001), *Phys. Rev. B* 69 (2004) 245313.
- [75] S.-M. Lee, D.G. Cahill, R. Venkatasubramanian, Thermal conductivity of Si–Ge superlattices, *Appl. Phys. Lett.* 70 (1997) 2957–2959.
- [76] Y. Li, Z. Fan, W. Li, D.L. McDowell, Y. Chen, A multiscale study of misfit dislocations in PbTe/PbSe (001) heteroepitaxy, *J. Mater. Res.* 34 (2019) 2306–2314.
- [77] A. Stukowski, V.V. Bulatov, A. Arsenlis, Automated identification and indexing of dislocations in crystal interfaces, *Model. Simul. Mater. Sci. Eng.* 20 (2012) 085007.
- [78] H.C. Edwards, C.R. Trott, D. Sunderland, Kokkos: enabling manycore performance portability through polymorphic memory access patterns, *J. Parallel Distrib. Comput.* 74 (2014) 3202–3216.
- [79] Y. Chen, The origin of the distinction between microscopic formulas for stress and Cauchy stress, *Europhys. Lett.* 116 (2016) 34003.