A Unified Framework and Performance Benchmark of Fourteen Multiscale Atomistic/Continuum Coupling Methods*

Ronald E. Miller[1] and E. B. Tadmor[2]

[1]Department of Mechanical and Aerospace Engineering,
Carleton University, Ottawa, ON, Canada, K1S 5B6

[2]Department of Aerospace Engineering and Mechanics,
University of Minnesota, 107 Akerman Hall, 110 Union St SE,
Minneapolis, MN 55455, USA

May 8, 2009

Abstract

A partitioned-domain multiscale method is a computational framework in which certain key regions are modeled atomistically while most of the domain is treated with an approximate continuum model (such as finite elements). The goal of such methods is to be able to reproduce the results of a fully-atomistic simulation at a reduced computational cost. In recent years, a large number of partitioned-domain methods have been proposed. Theoretically, these methods appear very different to each other making comparison difficult. Surprisingly, it turns out that at the implementation level these methods are in fact very similar. In this paper, we present a unified framework in which fourteen leading multiscale methods can be represented as special cases.

We use this common framework as a platform to test the accuracy and efficiency of the fourteen methods on a test problem; the structure and motion of a Lomer dislocation dipole in face-centered cubic aluminum. This problem was carefully selected to be sufficiently simple to be quick to simulate and straightforward to analyze, but not so simple to unwittingly hide differences between methods. The analysis enables us to identify generic features in multiscale methods that correlate with either high or low accuracy and either fast or slow performance.

All tests were performed using a single unified computer code in which all fourteen methods are implemented. This code is being made available to the public along with this paper.

*To appear in *Modeling and Simulation in Materials Science and Engineering*, 2009. This article will also appear in a modified form as a chapter in the upcoming book: E. B. Tadmor and R. E. Miller, *Modeling Materials: Continuum, Atomistic and Multiscale Techniques*, Cambridge University Press.

1 Introduction

In recent years, there has been great interest in developing a class of concurrent multiscale modeling methods that are able to obtain the same results as classical molecular statics (MS) or molecular dynamics (MD) simulations at a fraction of the computational cost. This is achieved by atomistically[1]modeling only small, strategically chosen sub-domains of the problem, with most of the body modeled using a numerical approximation to continuum mechanics such as the finite element method. For certain classes of problems, of which fracture and nanoindentation are quintessential examples, deformation is naturally localized in a manner that lends itself to this sort of atomistic/continuum partitioning of the model. We refer to multiscale formulations of this type as *partitioned-domain* methods.

There is a vast literature surrounding these methods, as summarized in several recent reviews [9, 26, 34, 13, 31]. However, it is extremely difficult to digest the similarities, differences, and relative strengths and weaknesses of the various methods. This is due, in part, to the disparate backgrounds, target applications, and conceptual frameworks each research group brings to bear on the problem. In some instances, models that appear starkly different are in fact very similar or even identical in certain specific cases. It is also hard to quantify the relative accuracy and efficiency of these methods. Each method is implemented in a different computer program, perhaps optimized for specific atomistic systems or specific types of problem. While most groups are diligent in presenting "proof-of-principle" simulations or simple benchmarks of their models against direct atomistic simulation, it is not possible to establish how other similar methods would fare against the same tests.

This paper presents a comprehensive study of fourteen methods from the literature, all implemented within a single computer code [30] to facilitate the comparison between them. The details of these methods are reviewed in the following sections, culminating with the summary presented in table 1. The study sheds light on three important aspects of multiscale methods. First, it elucidates the key features of the models that uniquely define each method (or conversely the features that reveal when two apparently different methods are quite similar). Second, it allows us to define a benchmark test problem and run all the methods against it, determining the relative accuracy and efficiency of the different approaches. Finally, the study identifies trends in the model features that suggest ways to improve existing models; we are able to correlate certain features in the models with their effects on accuracy and efficiency. Some of our more significant observations, most notably the poor performance of cluster-based multiscale methods relative to those that use a continuum constitutive law, have recently been explained theoretically by Luskin and Ortner [29]. They have shown how the cluster-based methods introduce new types of errors that are especially pronounced in non-uniform meshes like the one we will use in our comparison.

The intended audience of the paper is mainly developers of these and related methods; we hope that it will help guide future improvements. It will also be somewhat useful to potential *users* of these methods, but less so. By users, we imagine researchers that are already experienced with molecular dynamics and other atomistic methods, and who are trying to determine whether implementing a multiscale method will be worth the trouble;

[1]We will use the term "atomistic" to refer generically to classical molecular statics and molecular dynamics techniques.

to understand if multiscale methods have a chance of solving their particular problem of interest. Such a user should be less interested in our comparison of the performance of the methods, focussing instead on the discussion of the details of the various methods presented and on the actual code used to perform these tests (which is available at [51]). This will give a sense of the effort required. Finally, in assessing the overall capabilities of these methods, the user should keep in mind that all of the methods discussed herein can be implemented in 3D, and some of them address the question of finite temperature and dynamics. For these details, the user should consult the original references and recent reviews such as [31]. The user should also remember that our goal here was not to create an especially efficient implementation, but rather to create one that fairly compares the methods. Much can be done to improve the speed of our implemenation for all of the methods.

The list of methods implemented is not exhaustive. Indeed, there are so many variants of these methods, with new ones still appearing, that a truly exhaustive study is beyond our finite capabilities. However, we believe that the study is strongly representative of the entire body of literature to date, in the sense that we believe that each of the major methods can be shown to be very similar to one of the methods on our list. This will become more evident in the following sections where we define and discuss the key features that define a multiscale model. An important exception to this is the coarse-grain molecular dynamics (CGMD) method of [43]. This method is notably different from the methods presented here in the way that it constructs the continuum model from the atomistics. Unfortunately, we have not been able to include CGMD in our study.

Although many of the methods discussed here were developed with dynamics in mind, we focus only on the static limit. As such, we are putting aside questions of wave reflections, thermostats and others that remain open challenges to the development of dynamic multiscale models [31]. Our goal is to focus on the accuracy and efficiency of just the *coupling method itself*. Any errors or inefficiencies present in the static implementation of a coupling method will remain in the dynamic setting. Dynamics may introduce new challenges, and indeed some of these methods may be better suited than others to address these dynamic issues, but on this matter our study will be unable to opine.

2 Defining a Multiscale Model

A generic multiscale model is presented in fig. 1. In fig. 1(a), we schematically illustrate an idealized partitioning, where the body is unambiguously divided into two regions. Region B^A is treated atomistically, while region B^C is modeled as a continuum. Most methods use a finite element method to treat the continuum, we will therefore regard the continuum region as a discretized approximation represented by a finite element (FE) mesh.[2] An interface between the two regions is identified, ∂B^I, across which compatibility and equilibrium are enforced. However, the finite range of interactions of the atoms typically requires a more elaborate interfacial region with finite width, B^I, as illustrated in fig. 1(b). Boundary conditions, in the form of prescribed tractions on ∂B_t^C, forces on ∂B_t^A or displacements on

[2]There are also multiscale methods that use a "meshless method" instead of finite elements to approximate the continuum region. See for example the recent review by Park and Liu [34]. We do not discuss such methods here.

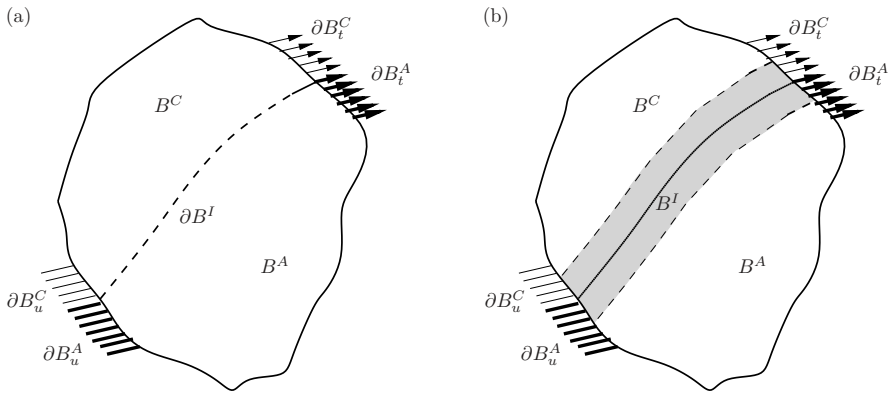


Figure 1: (a) A general partitioned domain problem. B^C is modeled as continuum while all atoms in B^A are explicitly treated as discrete degrees of freedom. (b) Some methods require a finite interfacial region, B^I, where the atomistic and continuum features must coexist.

$\partial B_u = \partial B_u^A \cup \partial B_u^C$ are applied to induce deformation.

The key to any multiscale method lies in the way that it handles the interface region shown in fig. 1(b). Most methods adopt one of the strategies depicted in fig. 2. On the left is perhaps the most generic interface, which in various limiting cases leads to all the models discussed here. The interface region B^I, shown by the dashed lines, has been further subdivided into two parts. We refer to these as the "handshake region", B^H (filled circles), and the "padding region", B^P (open squares). The size and nature of these regions depend on the specifics of multiscale model, as discussed below.

The handshake region is neither fully-atomistic nor fully-continuum. It is a region where there is some degree of mixing between the two descriptions of the material as we explain later. The padding region is continuum in nature, however it is used to generate atoms that provide the boundary conditions to the atoms in B^A and B^H. This is a necessity born out of the nonlocal nature of atomic bonds, and therefore the thickness of the padding depends on the range of the atomistic interactions, r_{cut}. The motion of atoms in B^P is determined from the continuum displacement fields at the position of the padding atoms, in different ways for different methods.

Several models use less general versions of the interface in fig. 2(a). The most common variation is to eliminate the handshake region (or more precisely, reduce it to a surface of zero thickness). This is illustrated in fig. 2(b).

For many models, the imposition of displacement compatibility across the interface does not require a clear correspondence between atoms and nodes at the interface; as illustrated in fig. 2(a) the nodes (vertices of the triangular elements) need not coincide with atom positions. On the other hand, most models that do not include a handshake region require a direct atom-node correspondence along the edge of the finite element region (the heavy jagged line in fig. 2(b)). This provides a way to impose displacement boundary conditions on the finite elements, as the nodes along this edge move in lock step with the corresponding atoms. The cost of this is a restriction on the finite element mesh, which must be refined down to the

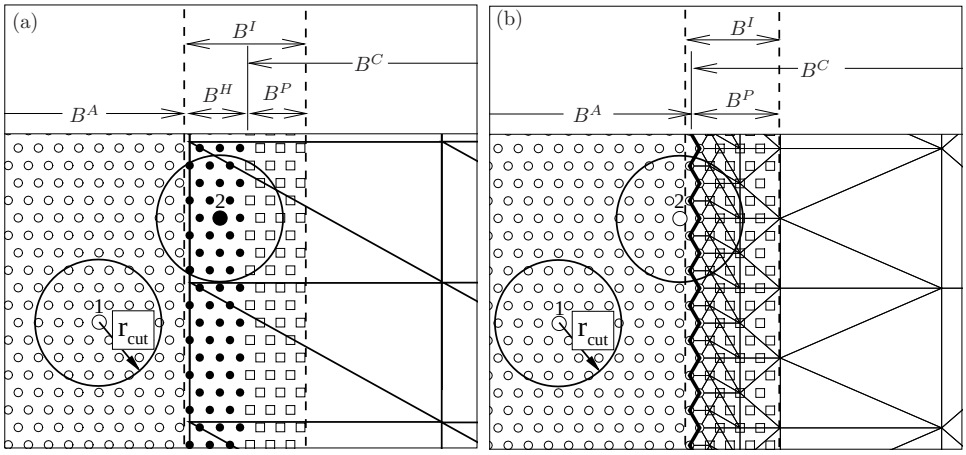


Figure 2: A generic interface in a coupled atomistic/continuum problem. The finite cut-off radius of the atoms mean that an atom like 1 cannot "see" into the continuum, while atom 2 can. Thus the need for a "padding" region as discussed in the text. The model on the left includes includes a handshake region, B^H, while the model on the right does not. Padding atoms are shown as open squares, handshake atoms as black circles and regular atoms as white circles. See the text for the discussion of frames (a) and (b).

atomic scale on the continuum side of the interface (we refer to this as a "fully-refined" mesh), but normally the mesh coarsens as it extends deeper into the continuum region.

There are then four primary ways in which various multiscale methods differ. We list these differences here, roughly in order of decreasing importance (to the extent that they define the fundamental differences between various models):

1 *The governing formulation.* For static methods, there are two fundamentally different approaches to finding equilibrium. The first (which we call the "energy-based" approach) is to develop a well-defined energy functional for the problem and rigorously minimize this energy. The second ("force-based" approach) is to develop a physically motivated set of forces on all degrees of freedom, and reach equilibrium by driving these forces to zero. While the two seem as though they might be equivalent, they are not. Derivatives of the total energy in the energy-based approach lead to forces on each degree of freedom which are necessarily zeroed when the energy is minimized, however these are not the same as the forces used in the force-based methods. The force-based methods have *no well-defined total energy.* Instead, they start from a definition of the coupling in terms of a physically-motivated prescription of the forces on unequilibrated atoms or nodes.

   As we shall discuss later, the disadvantage of the energy-based approach is that it is extremely difficult to eliminate the non-physical side effects of the coupled energy functional, dubbed "ghost forces" in the literature[3]. On the other hand, force-based

[3]The existence of ghost forces was first discussed in the original derivation of the QC method [50]. Several authors have since then devoted considerable time to discussing the source and nature of these ghost forces [9, 49, 14, 5, 12]. The name "ghost forces" was coined by David Rodney and appears for the first time in

methods can be slow to equilibrate, can converge to unstable equilibrium states (such as saddle points or maxima), are non-conservative, and can be numerically unstable. Later in this paper we will quantitatively compare the energy-based and force-based approaches.

2 *The Coupling Boundary Conditions.* The coupling between the continuum and atomistic regions, requires compatibility conditions in each direction. This means providing some prescription for finding the displacements of atoms in the padding region, B^P, from the nodal displacement information in B^C, as well as a way to determine the displacement boundary conditions for the FE nodes along the edge of the mesh closest to the atomistic region. As we describe the various methods, we will see that compatibility can be imposed in a "strong" or "weak" sense, as we elaborate here.

"Strong compatibility" imposed by the continuum on the atoms implies that the padding atoms are constrained to move as though they are "glued" to the finite elements in which they reside. That is to say that the displacement, $\tilde{\boldsymbol{u}}^\alpha$, of a padding atom α is set to

$$\tilde{\boldsymbol{u}}^\alpha = \sum_{I=1}^{n_{\text{nodes}}} N^I(\boldsymbol{X}^\alpha)\mathbb{U}^I, \tag{1}$$

where $\mathbb{U}^I$ is the displacement vector of node I, $\boldsymbol{X}^\alpha$ is the reference coordinate position of node α, $N^I(\boldsymbol{X})$ is the finite element shape function of node I and there are n_{nodes} nodes.[4]

Strong compatibility is imposed *by the atomistic region* on the continuum by defining a subset of the nodes (typically a row like the jagged, heavy line in fig. 2(b)) that are coincident with some of the atoms in B^A. These atoms are formally part of B^A. Their contribution to the total energy or their force (depending on whether the method is energy-based or forced-based) is computed as appropriate for the atomistic region. However, their positions define the displacements of a set of nodes whose reference coordinates coincide with the reference coordinates of the black atoms. The motion of the black atoms therefore imposes a displacement boundary condition on the finite elements of B^C.

In our view, the main disadvantage with strong compatibility is the added complexity in mesh generation near the interface. The mesh must usually be fully-refined and consistent with the underlying atomic configuration (which is normally a crystalline lattice and therefore requires additional lattice-based algorithms [3]). To avoid these additional requirements, some methods employ what we call "weak compatibility", where displacement boundary conditions are enforced only in some average sense, or

[46].

[4] In some methods, a special case is adopted where the FE mesh is fully-refined to the atomic scale in the entire padding region. That is to say that a node is defined to lie on top of every atom in B^P. This portion of the B^C region is not *atomistic*, since the region is still treated using the continuum finite element formulation. It is merely a continuum region with very small elements. A fully-refined padding region of this type serves to simplify the implementation, since as far at the atoms in B^A are concerned, the nodes in the continuum are effectively atoms as well. As such, the interpolation of displacements to the padding atoms is no longer required, but this does not otherwise significantly alter the model.

with some type of penalty method approach. These methods make mesh generation much easier, but we will show later that they are generally less accurate. Examples of weak compatibility will be presented in the context of specific methods later.

3 *The Handshake Region.* The idea of the handshake region is to provide a gradual transition from the atomistic to continuum model. The handshake region is treated very differently in different coupled models. In some, the atomistic-to-continuum transition is abrupt as in fig. 2(b), and there is no handshake region (*e.g.*, the QC and CLS methods of section 4.1 and section 4.2, the FEAt and CADD methods of section 6.1). In others, a handshake region exists as in fig. 2(a), but the ways in which it blends the continuum and atomistics descriptions can differ substantially. We discuss these details below.

4 *Treatment of the Continuum.* As we have mentioned, B^C is usually modeled with finite elements, but the details of the finite element formulation and the constitutive law adopted to describe the material response differ amongst the methods. In some cases, a simple small strain finite element formulation is used with a linear elastic model with elastic constants fitted to the properties of the atomistic model used in B^A. In others, a finite strain (nonlinear) formulation is used together with the Cauchy-Born rule [17] (or some more sophisticated appeal to the atomistic model) to describe the constitutive response in the nonlinear range.

We now turn to a description of the different methods considered in this study. In doing so, we divide the methods into either energy-based or force-based and explore the essential differences that this division creates.

3 The Energy-Based Formulation

3.1 Total Energy Functional

In an energy-based formulation, it is assumed that the total potential energy Π^{tot} of a body B can be written as the sum of the potential energies of the three sub-bodies B^C, B^A and B^H from which it is composed. (Recall that the padding region, B^P is really considered a part of the continuum.) Thus

$$\Pi^{tot} = \Pi^A + \widehat{\Pi}^C + \widehat{\Pi}^H, \tag{2}$$

where the hat on Π^C and Π^H indicates that the potential energy of the continuum is approximated by a finite element formulation. For this approach to work, a prescription must be provided for computing the energy of each of these regions. This is straightforward in the continuum, where an energy density is postulated to exist at every point. In the atomistic region, the analogy of an energy density is a well-defined energy per atom, which most

empirical atomistic models permit.[5] Thus the contributions from B^A and B^C are

$$\Pi^A = \sum_{\alpha \in B^A} E^\alpha - \sum_{\alpha \in B^A} \boldsymbol{f}^\alpha_{\text{ext}} \cdot \tilde{\boldsymbol{u}}^\alpha, \tag{3}$$

$$\Pi^C = \int_{B^C} W(\boldsymbol{F})\, dV - \int_{\partial B^C_t} \bar{\boldsymbol{t}} \cdot \boldsymbol{u}\, dA, \tag{4}$$

where $\tilde{\boldsymbol{u}}^\alpha$ is the displacement of atom α, $\boldsymbol{u}(\boldsymbol{X})$ is the continuum displacement field, the energy of atom α is represented by E^α, and applied forces and tractions are denoted by $\boldsymbol{f}_{\text{ext}}$ and $\bar{\boldsymbol{t}}$, respectively. The strain energy density, W, is a function of the deformation gradient, $\boldsymbol{F} = \boldsymbol{I} + \nabla \boldsymbol{u}$. Once discretized by finite elements, the continuum contribution is approximated by Gauss quadrature (numerical integration) as [61]

$$\widehat{\Pi}^C = \sum_{e=1}^{n_{\text{elem}}} \sum_{q=1}^{n_{\text{q}}} w_q V^e W(\boldsymbol{F}(\boldsymbol{X}^e_q)) - \bar{\mathsf{F}}^T \mathbb{U}, \tag{5}$$

where n_{elem} is the number of elements, V^e is the volume of element e, n_{q} is the number of quadrature points, $\boldsymbol{X}^e_q$ is the position of quadrature point q of element e in the reference configuration, w_q are the associated Gauss quadrature weights, and $\bar{\mathsf{F}}$ and $\mathbb{U}$ represent the vector of applied forces and nodal displacements, respectively, in the FE region.

The energy of the handshake region is a partition-of-unity blending of the atomistic and continuum energy descriptions. For example, a well-known energy-based method which includes a handshake region is the bridging domain (BD) method described in [60]. Within the BD handshake region, both the continuum and atomistic energies are used, but their contributions are weighted according to a function Θ that varies linearly from 1 at the edge of B^H closest to B^C to 0 at the edge closest to B^A. Assuming no externally applied forces on this region to simplify the notation, the energy of the handshake region is

$$\Pi^H = \sum_{\alpha \in B^H} (1 - \Theta(\boldsymbol{X}^\alpha)) E^\alpha + \int_{B^H} \Theta(\boldsymbol{X}) W(\boldsymbol{F}(\boldsymbol{X}))\, dV. \tag{6}$$

If we take the specific case of constant-strain triangular elements, and make the relatively minor restriction that the handshake region starts and ends along a contiguous line of element edges (in other words, every element is either entirely "in" or entirely "out" of region B^H), it is simple to evaluate the integrals above. Since W is constant within each constant-strain element in this case, the energy becomes

$$\widehat{\Pi}^H = \sum_{\alpha \in B^H} (1 - \Theta(\boldsymbol{X}^\alpha)) E^\alpha + \sum_{e \in B^H} \Theta(\boldsymbol{X}^e_{\text{cent}}) W(\boldsymbol{F}(\boldsymbol{X}^e_{\text{cent}})) V^e, \tag{7}$$

where $\boldsymbol{X}^e_{\text{cent}}$ is the coordinate of the Gauss point in element e (the centroid of the triangle in this specific case).

We emphasize that the padding atoms in region B^P, described previously, are distinct from the atoms contained in region B^H; B^H and B^P do not overlap. The padding region

[5] It is worth noting, however, that a quantum mechanical density functional theory (DFT) description cannot be uniquely decomposed into an energy-per-atom description.

corresponds to a set of atoms whose energy is not explicitly included in the energy functional but whose positions are required to provide the full neighbor environment to other atoms. In fact, a padding region is still needed in the BD model (in addition to the atoms in the handshake region). The padding region lies inside B^C along the edge of B^H.

For energy-based methods with no handshake, the energy reduces to

$$\Pi^{tot} = \Pi^A + \widehat{\Pi}^C. \tag{8}$$

Given the positions of the atoms in the atomistic region, the displacements of the nodes in the FE region and a prescription for generating the padding atoms, we can compute the energy of eqn. (2) or (8), and its derivatives as needed. The total potential energy is then minimized subject to the imposed displacement boundary conditions to obtain the equilibrium configuration of the system.

3.2 Ghost Forces

The approximations inherent to an energy-based coupling of this type lead to errors known as "ghost forces", a phrase first coined in [46]. All energy-based methods discussed in this study suffer from these forces to various degrees[6]. The ghost forces are defined as follows. We build a model in which the atoms are on their equilibrium crystal lattice sites and the finite elements are unstressed and undeformed. Physically, this should be an equilibrium configuration where all forces are zero, and therefore any forces on atoms or nodes that arise in this configuration are unphysical and will lead to spurious distortions of the body upon relaxation. These unphysical forces are the ghost forces.

For a little insight into the origin of the ghost forces, let's consider an energy-based method without a handshake as in eqn. (8). This is an approximation to some full atomistic description of the body, whose energy we will denote as Π^{atom}. Assume that we can write this energy as a contribution from each region, B^A and B^C, respectively

$$\Pi^{\text{atom}} = \Pi^{\text{atom,A}} + \Pi^{\text{atom,C}}. \tag{9}$$

Usually, Π^A is defined such that

$$\Pi^A = \Pi^{\text{atom,A}}, \tag{10}$$

but $\widehat{\Pi}^C$ is only an approximation to $\Pi^{\text{atom,C}}$. Now consider an atom, α, inside B^A, but near the interface, such that it interacts with atoms in B^C. In the fully-atomistic description, a force on this atom is

$$\boldsymbol{f}^\alpha = -\frac{\partial \Pi^{\text{atom,A}}}{\partial \tilde{\boldsymbol{u}}^\alpha} - \frac{\partial \Pi^{\text{atom,C}}}{\partial \tilde{\boldsymbol{u}}^\alpha} = -\frac{\partial \Pi^{\text{A}}}{\partial \tilde{\boldsymbol{u}}^\alpha} - \frac{\partial \Pi^{\text{atom,C}}}{\partial \tilde{\boldsymbol{u}}^\alpha}, \tag{11}$$

where the second equality comes from eqn. (10). In equilibrium, this force must be zero, so that

$$\frac{\partial \Pi^{\text{A}}}{\partial \tilde{\boldsymbol{u}}^\alpha} = -\frac{\partial \Pi^{\text{atom,C}}}{\partial \tilde{\boldsymbol{u}}^\alpha}. \tag{12}$$

[6]This statement may seem controversial, but we believe it to be accurate, as we have not yet seen a method that avoids the problem. It should be noted that there are special cases of *all the methods* for which ghost forces disappear. Usually, this happens for a particular choice of interatomic potential. For example, the original implementation of the CLS method used the Stillinger-Weber potential for Si, for which there are no ghost forces. However, an implementation of CLS that uses an EAM potential will have ghost forces.

For the model system of eqn. (8) to be in equilibrium, we likewise require that

$$\frac{\partial \Pi^{\mathrm{A}}}{\partial \tilde{\boldsymbol{u}}^{\alpha}} = -\frac{\partial \widehat{\Pi}^{\mathrm{C}}}{\partial \tilde{\boldsymbol{u}}^{\alpha}}, \tag{13}$$

however eqn. (12) means that this can only be satisfied if

$$\frac{\partial \widehat{\Pi}^{\mathrm{C}}}{\partial \tilde{\boldsymbol{u}}^{\alpha}} = \frac{\partial \Pi^{\mathrm{atom,C}}}{\partial \tilde{\boldsymbol{u}}^{\alpha}}. \tag{14}$$

If this equality does not hold, there will be residual ghost forces even though the atoms are in their equilibrium positions. However, since $\widehat{\Pi}^{C}$ is only an approximation to $\Pi^{\mathrm{atom,C}}$, it is not possible to satisfy this expression in general. There has been work done to satisfy this expression for certain crystal types and interfacial structures [49, 14], but ghost forces persist in general.

Much more elaborate discussion of the nature and origin of these spurious effects have been presented at length in references such as [46, 9, 14, 11, 12]. Later in section 7.4, we will see their effect in each method for a specific example problem.

4 A Review of Energy-Based Coupling Methods

We now turn to a brief review of several coupled methods from the literature, focusing on their static limit. Our starting point will be the Quasicontinuum (QC) method because this is the method with which we are most familiar. Where a model is similar to QC or some other model already described, we will simply state this similarity without re-hashing the details. This is not meant to imply, necessarily, that one method was a direct off-shoot from another. It shows only that different methods, developed independently, often share common traits.

4.1 The QC method

The QC method is an energy-based method with no handshake regions, so the energy functional of eqn. (8) is used. However, there is a conceptual advantage in developing this equation from a point of view that makes no distinction between atoms and nodes, for reasons that will become clear later when we introduce cluster-based methods. In the QC literature, any node or atom that is retained in the model as part of the set of degrees of freedom is identified as a "representative atom" or simply "repatom". The terminology "nonlocal repatom" and "local repatom" is used to distinguish between repatoms in the atomistic region where the atomic bonding is nonlocal and repatoms in the continuum region where a local constitutive relations is applied.

The idea is that we seek to compute a good approximation to the total energy of the fully-atomistic system, which is a function of the positions of all the atoms in the body, $\boldsymbol{x}^{\alpha}$ ($\alpha = 1, \ldots, N_{\mathrm{A}}$). As usual we can assume, without loss of generality, that we know some reference configuration of the atoms, $\boldsymbol{X}^{\alpha}$ and then work in terms of displacements,

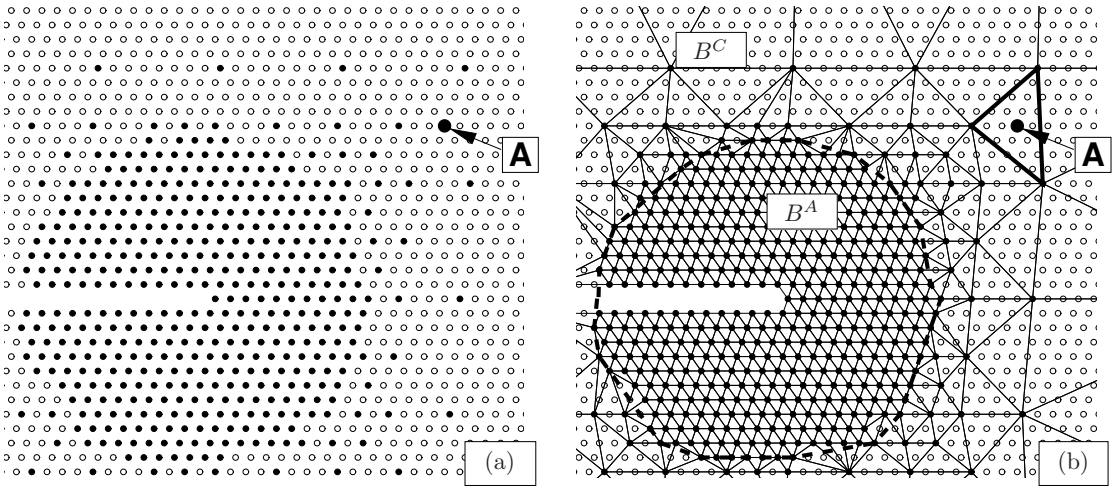


Figure 3: Repatoms selected from all the atoms near a crack tip are shown in (a), which are then meshed by linear triangular elements in (b). The density of repatoms varies according to the severity of the variation in the deformation gradient; high density near the singularity at the crack tip and low elsewhere. The dashed line indicates the atomistic/continuum interface.

$\tilde{\boldsymbol{u}}^\alpha = \boldsymbol{x}^\alpha - \boldsymbol{X}^\alpha$. The energy of the atomistic system is then

$$\Pi = \mathcal{E}(\tilde{\boldsymbol{u}}^1, \ldots, \tilde{\boldsymbol{u}}^{N_\mathrm{A}}) - \sum_{\alpha=1}^{N_\mathrm{A}} \boldsymbol{f}^\alpha_{\mathrm{ext}} \cdot \tilde{\boldsymbol{u}}^\alpha, \tag{15}$$

where $\mathcal{E}$ is the interaction energy of the atoms, and we have included the possibility of external forces applied to the atoms.

Once again, we reduce generality somewhat by assuming a class of atomistic models that permits the identification of a site energy, E^α, for each atom, and thus

$$\mathcal{E}(\tilde{\boldsymbol{u}}^1, \ldots, \tilde{\boldsymbol{u}}^{N_\mathrm{A}}) = \sum_{\alpha=1}^{N_\mathrm{A}} E^\alpha(\tilde{\boldsymbol{u}}^1, \ldots, \tilde{\boldsymbol{u}}^{N_\mathrm{A}}). \tag{16}$$

For a more concise notation, let $\tilde{\boldsymbol{u}}$ represent the entire set of N_A displacements, $\tilde{\boldsymbol{u}} = \{\tilde{\boldsymbol{u}}^1, \ldots, \tilde{\boldsymbol{u}}^{N_\mathrm{A}}\}$, so that

$$\mathcal{E}(\tilde{\boldsymbol{u}}) = \sum_{\alpha=1}^{N_\mathrm{A}} E^\alpha(\tilde{\boldsymbol{u}}). \tag{17}$$

We start by reducing the number of degrees of freedom in the problem. To achieve this, we constrain the motion of most of the atoms to follow the motion of a representative handful of the atoms. How we choose these repatoms is not our concern here[7], but we can

[7]The original papers describing the QC method [46] and the CACM method [10] provide methodologies for automatically choosing the density of repatoms in various regions. More recently, goal-oriented error estimation techniques have also been developed for this purpose [38].

imagine something similar to the illustration in fig. 3, where the density of repatoms varies according to the expected severity of the deformation. In some critical regions (as around a crack tip in this example) all atoms are selected as repatoms, while very few are selected in regions that are more or less linear elastic. For now, we assume that the displacements of some n_R repatoms, $n_R \ll N_{\mathrm{A}}$, are chosen as the degrees of freedom. A finite element mesh with constant-strain triangles (in 2D) or constant-strain tetrahedra (in 3D) is defined by the repatoms as shown in fig. 3(b), and the repatoms behave as the finite element nodes. The displacements of all other atoms can then be determined from the displacement of these node/repatoms through interpolation. In this way, the potential energy of the atoms becomes a function of *only* the node/repatom displacement vector, $\mathbb{U}$:

$$\mathcal{E}(\mathbb{U}) = \sum_{\alpha=1}^{N_{\mathrm{A}}} E^{\alpha}(\tilde{\boldsymbol{u}}(\mathbb{U})), \tag{18}$$

where the dependence of the atom displacements on the repatom displacements is through the finite element shape functions. By the compact support of these functions, we know, for example, that the displacement of atom A in fig. 3(b) is a linear interpolation of the displacements of the three repatoms forming the nodes of the highlighted element.

Clearly, the constraints introduced by the interpolation of the displacements will mean some level of approximation to the model, but we can presumably control the error it introduces by recognizing regions were the element size needs to be refined. In the fully-refined limit where every atom is chosen as a repatom we recover ordinary lattice statics, since in this limit there are no atoms "between" the repatoms whose motion will be constrained.

This has reduced the number of degrees of freedom in the model, but has not significantly lowered the computational burden associated with computing the energy of the system or the forces on the repatoms. This is because we still need to compute the energy of every atom in the sum over E^{α} in eqn. (18).[8] At this stage the division of the body into the atomistic and continuum sub-bodies becomes necessary, and we rewrite the energy as

$$\mathcal{E}(\boldsymbol{u}) = \sum_{\alpha \in B^A} E^{\alpha}(\tilde{\boldsymbol{u}}(\mathbb{U})) + \sum_{\alpha \in B^C} E^{\alpha}(\tilde{\boldsymbol{u}}(\mathbb{U})). \tag{19}$$

In the original papers [53, 45, 46], the QC developers describe a systematic way to make this division, but here we will merely illustrate the idea by treating all of the fully-refined region (see fig. 3) as B^A and the coarse regions as B^C[9]. We then propose to compute the first sum in the above equation exactly as we would in a fully-atomistic model; the only difference from the exact result comes from the fact that the padding atoms in neighboring regions are undergoing a constrained deformation. The second sum, on the other hand, we replace by finite elements and the Cauchy-Born constitutive rule. Thus for an atom in B^C we assume that it is in a region of approximately uniform deformation, characterized by $\boldsymbol{F}$, so that

$$E^{\alpha} \approx \Omega_0 W[\boldsymbol{F}(\boldsymbol{X}^{\alpha})], \tag{20}$$

[8] It is worth pointing out that up to this point there are still no ghost forces in the model. These are introduced next as the continuum approximation is imposed.

[9] This does not mean that the QC method precludes the possibility of a fully-refined continuum region. Making the entire fully-refined region atomistic is one possible prescription of the model.

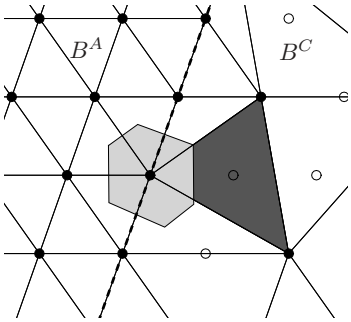


Figure 4: Computing the effective volume of the darkly shaded element requires subtraction of the part of its volume overlapping the Voronoi cells of atoms inside B^A (one is lightly shaded).

where Ω_0 is the Wigner-Seitz volume of a single atom in the reference configuration. The second sum over atoms in eqn. (19) can now be replaced by a sum over *elements* as

$$\mathcal{E}(\boldsymbol{u}) = \sum_{\alpha \in B^A} E^\alpha(\tilde{\boldsymbol{u}}(\mathbb{U})) + \sum_{e \in B^C} n^e \Omega_0 W(\boldsymbol{F}^e), \tag{21}$$

where n^e is the number of atoms contained in element e and $\boldsymbol{F}^e$ is the deformation gradient there.

A practical point concerns the determination of n^e, as it would seem difficult to unambiguously assign atoms to specific elements if they lie at element corners or edges. In fact, $n^e\Omega_0$ can simply be taken as the total volume of the element, eliminating the need for explicitly counting atoms. This leaves one subtlety at the atomistic/continuum interface, as shown for example in fig. 4, where the darkly shaded element is adjacent to an atom that is inside B^A and therefore already accounted for in the atomistic contributions. For this element, the volume used should be the total volume less the volume overlapping the Voronoi cell of the atom in B^A. In this way, $n^e\Omega_0$ values for elements touching B^A are reduced to avoid the double-counting of energy. In practice, approximations to the Voronoi cell can be used to expedite calculations without too much consequence for accuracy [46].

Note the formal equivalence between eqn. (21) and eqn. (8), when constant-strain triangles with a single Gauss point are used. The only difference is the modified weight associated with finite elements near the atomistic/continuum interface to avoid double counting. We have assumed constant-strain elements to emphasize the relation to the CB rule, but it would be straightforward to replace the sum over elements with a sum over Gauss points if higher-order elements were used.

In the energy-based cluster-based QC (CQC(m)-E) method that will be discussed in section 4.6, a different approach to approximating the second sum in eqn. (19) will be introduced.

4.2 The Coupling of Length Scales (CLS) method

The Coupling of Length Scales (CLS) method was developed in a series a papers [1, 8, 2, 42] focused initially on the problem of fracture in silicon. It was later applied to the vibrational response of nanoscale resonators [41].

In addition to the atomistic/continuum coupling that is our focus here, CLS also includes a coupling between molecular dynamics and the tight-binding approximation to quantum mechanics (QM). As such, there are three concurrent regimes in the CLS approach: continuum coupled to classical atomistic coupled to quantum mechanics.[10] For this particular discussion our focus will be only on the atomistic/continuum coupling.

CLS is an energy-based method like QC. Compared with the energy functional in eqn. (21) for the QC method, the energy of the CLS method takes the form[11]

$$\mathcal{E}(\boldsymbol{u}) = \mathcal{E}_A + \sum_{e=1}^{n_{\text{elem}}} W(\boldsymbol{F}^e) w^e, \tag{22}$$

where $\mathcal{E}_A$ is the energy contained in region B^A, and the second sum is over all the elements in the continuum region. For most elements, the weight w^e is simply the volume of the element. The exceptions are elements directly adjacent to the atomistic/continuum interface, where the weights are reduced to avoid double counting.

Because CLS was originally developed for silicon[12], it was crafted in terms of the Stillinger-Weber empirical model, whereas the QC method focused on applying embedded-atom method (EAM) potentials more suitable for metals. The similarities between the models are largely disguised by this different starting point, and the two methods nearly converge for the same underlying atomistic model.

There are small differences between QC and CLS in the treatment right at the interface. In short, there is some ambiguity regarding the parceling of energy near the interface. The QC and CLS methods use slightly different methods of weighting the atomistic and continuum contributions in efforts to avoid double counting of energy. The effect is reflected in differences between the values of w^e for CLS and the analogous $n^e\Omega_0$ in eqn. (21) for QC. These differences are relatively minor in that they lead to only slight changes in the error and the rate of convergence of the models. It is not clear that either the QC or the CLS weighting scheme is superior; it seems that the relative error will be problem-dependent.

4.3 The Bridging Domain (BD) Method

One way to try to mitigate the ghost force effect is to make the transition from nonlocal atomistics to local continuum mechanics less abrupt. Although somewhat *ad hoc*, this can

[10] More recently, MD/QM coupling (or more precisely MS/QM coupling, since the focus has been on the static case) of this type has been developed by a number of researchers [18, 27].

[11] The original CLS method used a small strain approximation to describe the continuum region rather than the CB rule used in QC. However, conceptually the methods are similar. To facilitate the comparison, we generalize the original CLS approach here by adopting a nonlinear CB model in the continuum region. We discuss the effect of the linear elastic approximation in section 7.6.

[12] Note that for a multilattice crystal, such as silicon, the strain energy density in eqn. (22) should also depend on the "shifts" of the sub-lattices. See for example [55]. We do not include this since in this paper we focus on simple lattice crystals.

be achieved by the simple blending of energy in the handshake region between the two descriptions as described earlier in section 3. The approach is related to the so-called Arlequin method [6], which is a general approach to coupling different models based on a blended energy functional. The Bridging Domain (BD) method is perhaps the first application of the blending region within an energy-based scheme [60], and is certainly representative of such models. Briefly, the BD model makes use of eqn. (2) to define the energy, including a handshake region with a specifically-defined blend of the atomistic and continuum energy functionals via eqn. (6).

In the limit where the thickness of the handshake region goes to zero and the size of the finite elements at the interface approaches the atomic spacing, we recover, more or less, the QC or CLS formulation. This is not precisely so, since the weights of the atomistic and continuum energy contributions in QC, CLS and BD would still be slightly different right at the interface. However, these differences will not substantially change the strength of the ghost forces that will occur in the interfacial region.

An important practical difference between the methods discussed previously and the BD method is the way in which compatibility is imposed between the atomistic and continuum regions. In the BD method, displacement constraints are introduced into the handshake region to force the displacement $\tilde{\boldsymbol{u}}^\alpha$ for every atom α in B^H to follow the displacements dictated by the interpolated displacement field at the atoms reference position, $\boldsymbol{u}(\boldsymbol{X}^\alpha)$. This is done using Lagrange multipliers by introducing the vector

$$\boldsymbol{h}^\alpha \equiv \boldsymbol{u}(\boldsymbol{X}^\alpha) - \tilde{\boldsymbol{u}}^\alpha = \sum_{I=1}^{n_{\text{nodes}}} N^I(\boldsymbol{X}^\alpha)\mathbb{U}^I - \tilde{\boldsymbol{u}}^\alpha, \tag{23}$$

which we want to constrain to be equal to zero. This can be achieved by redefining the handshake region energy in eqn. (7) to be

$$\begin{aligned}\widehat{\Pi}^H = &\sum_{\alpha\in B^H} (1-\Theta(\boldsymbol{X}^\alpha))E^\alpha(\tilde{\boldsymbol{u}}) + \sum_{e\in B^H} \Theta(\boldsymbol{X}^e_{\text{cent}})W(\boldsymbol{F}(\boldsymbol{X}^e_{\text{cent}}))V^e \\ &+ \sum_{\alpha\in B^H}\left[\beta_1\boldsymbol{\lambda}^\alpha\cdot\boldsymbol{h}^\alpha + \frac{\beta_2}{2}\boldsymbol{h}^\alpha\cdot\boldsymbol{h}^\alpha\right],\end{aligned} \tag{24}$$

where $\boldsymbol{\lambda}^\alpha = \{\lambda_1^\alpha, \lambda_2^\alpha, \lambda_3^\alpha\}$ are Lagrange multipliers for the degrees of freedom of atom α, and β_1 and β_2 are penalty functions that can be chosen to optimize computational efficiency. Note that the original BD paper was presented as a dynamic method. As such, forces on atoms and nodes come from the differentiation of this energy functional and include effects due to the constraint term. Since this term constitutes no more than a mathematical "trick" without clear physical significance (for example, what is the right choice of the penalties β_1 and β_2?), it is not clear how to interpret the resulting dynamics near the interface. In the static case, on the other hand, this is a convenient tool for finding a constrained minimum energy configuration, and the magnitude of the ghost forces can be examined. The use of a finite handshake region has the effect of smearing out the ghost forces, making the ghost force on a given atom or node smaller, but introduces ghost forces on more atoms and nodes. This will be more clearly evident in the discussion of section 7 to follow.

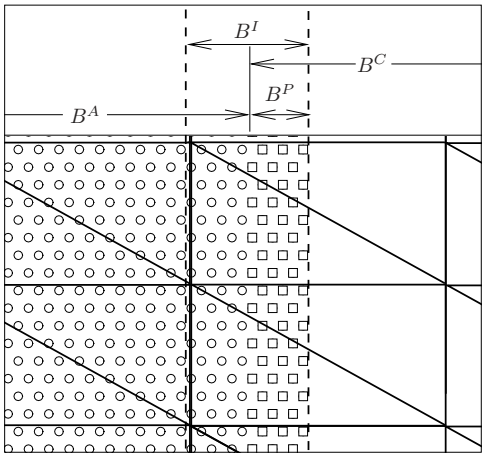


Figure 5: The BSM interface region. There is no handshake region and the finite elements exist throughout the body in order to store the coarse-scale displacement field.

4.3.1 Relaxing the condition of "Strong Compatibility"

In the QC and CLS methods, the "last row" of atoms is identified and made coincident with the "first row" of finite element nodes. The positions of these atoms provide the displacement boundary condition for the finite element region, which in turn imposes its displacement field on the "padding" atoms. This is what we have called the "strong compatibility" approach.

The Lagrange multipliers used in the BD method, on the other hand, are not a strong compatibility approach. Although they do, in the static limit, lead to the atoms in the handshake region identically conforming to the continuum displacement fields, they relax the compatibility constraint in two important ways. The first is that there is no longer a need to identify a one-to-one correspondence between atoms and nodes along an interface. The second is that in dynamic systems, departures from strong compatibility are possible and will lead to additional forces on the atoms and nodes in the handshake region.

The most useful feature of coupling approaches that relax strong compatibility is that this eliminates the need for one-to-one correspondence between atoms and nodes at the interface. This makes mesh generation easier, since elements can be made in any convenient way without regard for compatibility with the underlying atomic lattice.[13] On the other hand, we will show later that the trade-off is a reduced accuracy in the solution. An additional downside of weak compatibility is that extending the atomistic region at the expense of the continuum region, or vice versa, through automatic mesh adaption, becomes more difficult.

4.4 The Bridging Scale Method (BSM)

The groundwork for the bridging scale method (BSM) was laid first in reference [59] and later in reference [39]. The original formulation focused on dynamic simulations, while the later paper presented a static version of the approach. It is the latter that we focus on here.

[13]When strong compatibility is enforced, additional numerical algorithms for moving nodes onto lattice sites are required. See for example [3].

The BSM is an energy-based method with no handshake region (like QC and CLS), but the BSM is conceptually very different from other methods. This is because in the BSM there are (in principle) no separate continuum and atomistic regions. Instead, the BSM introduces the notion of *coarse-scale* and *fine-scale* displacement fields that exist *everywhere* in the body. As such, the picture is one where atoms exist throughout the body with a displacement

$$\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{u}}' + \tilde{\boldsymbol{u}}'', \tag{25}$$

where $\tilde{\boldsymbol{u}}'$ is the coarse-scale displacement and $\tilde{\boldsymbol{u}}''$ is the fine-scale. The former is stored on a finite element mesh everywhere in the body (overlapping all of B^C and B^A). The fine-scale displacements are assumed to exist in B^C as the manifestation of heat, but they are only explicitly tracked and stored inside B^A. The interface region for the BSM is illustrated in fig. 5.

The coarse-scale displacement field is stored on the finite element grid, and is defined as a least-squares fit to the underlying atomic displacements. In the atomistic region, this fit is explicitly determined from the atoms, while in the continuum region the inverse thinking is applied: it is imagined that the untracked atoms are vibrating around equilibrium positions that are known from the finite element displacements. In the zero-temperature static limit, the fine-scale displacements are assumed to be zero everywhere in B^C, so the static BSM is similar to the QC or CLS approach with two additional features: the partitioning of atomic displacements using eqn. (25) and a different compatibility condition. This is a mix of weak and strong compatibility, which we can describe next by referring to fig. 5.

Compatibility is first imposed on the atomistic region by constraining the square atoms in the padding region to follow the coarse-scale displacements of the finite elements using eqn. (1) (strong compatibility). Since it is assumed that there is no fine-scale displacement in B^C at zero temperature, this completely determines their positions. On the other hand, compatibility of the continuum region is imposed at all nodes lying inside B^A. Their displacements are constrained to follow the least-squares fit of the atomic displacements in neighboring elements (*i.e.*, $\tilde{\boldsymbol{u}}'$). This fully constrains elements deep inside B^A, whose energy is not included in the model, and constrains some of the nodes of elements straddling B^I. As we will see, this amounts to the same weak compatibility as described in more detail for the CACM model in the next section, with a particular choice of the weighting function g^I in eqn. (26).

In this way, the BSM mixes elements of weak and strong compatibility: strong compatibility for the motion of the padding atoms and weak compatibility for the enforcement of the finite element boundary condition.

The full energy of the atoms shown as open circles in fig. 5 is included in the BSM, while the energy of any finite element overlapping the filled circles is reduced by an amount that reflects the volume of the overlap. So, for instance, the energy of an element completely inside B^A is not included at all, the energy an element overlapping B^I is partially included, and the that of an element deep inside B^C is included as usual. Formally, then, the energy is identical to the QC energy of eqn. (21), but with different weighting factors n^e and a relaxation of the constraint that interfacial elements must be fully-refined.

The introduction of the coarse-scale displacement with eqn. (25) within the atomistic region introduces some complications. Mainly, this approach introduces a larger set of de-

pendent variables for which the equations must be solved; instead of satisfying equations for $\tilde{\boldsymbol{u}}$, we must now solve equations for two fields $\tilde{\boldsymbol{u}}'$ and $\tilde{\boldsymbol{u}}''$. As outlined in [39], this means that the static BSM requires an iterative approach to the solution, alternatively holding fixed the fine-scale variables while solving for the coarse-scale and vice versa. As we will discuss more quantitatively in section 7, iterative approaches like this tend to be quite slow.

In this study, we have implemented a variation of the BSM that eliminates the need for the iterative solver. This is possible in the static case because the fine-scale displacements are identically zero in B^C, while the coarse-scale in B^A is only required as a way to define the boundary condition on B^C. It is likely that the full BSM approach, with the iterative solver, will be slower, which we infer from our analysis of the CACM method discussed next.

4.5 CACM: Iterative minimization of two energy functionals

The problem of ghost forces stems from trying to combine two energy functionals from different models (an atomistic and a continuum model) into a single, coupled energy expression. A way to avoid ghost forces while retaining an energy formulation was proposed by [10] in the so-called Composite Grid Continuum Method (CACM). In CACM, no attempt is made to combine the atomistic and continuum energy functionals. Instead, they work with the two energy functionals separately.

The CACM method can be described using our generic interface pictured in fig. 2(a) without a handshake region[14]. First, the continuum problem is defined by setting the boundary condition on nodes that lie inside the atomistic region. This is done in an average sense, using a form of weak compatibility. In general

$$\bar{\mathbb{U}}^I = \sum_{\alpha=1}^{N_{\mathrm{A}}} g^I(\boldsymbol{X}^\alpha)\tilde{\boldsymbol{u}}^\alpha, \tag{26}$$

where the $\bar{\cdot}$ indicates a prescribed displacement and g^I is some general weighting function for node I. The CACM developers [10] chose to define g^I as a cubic function over a sphere of radius r_{cut} surrounding node I:

$$g^I(\boldsymbol{X}^\alpha) = 1 - 3\left(\frac{r}{r_{\mathrm{cut}}}\right)^2 + 2\left(\frac{r}{r_{\mathrm{cut}}}\right)^3,$$

where $r = \|\boldsymbol{X}^I - \boldsymbol{X}^\alpha\|$. Choosing

$$g^I(\boldsymbol{X}^\alpha) = \delta(\boldsymbol{X}^I - \boldsymbol{X}^\alpha), \tag{27}$$

introduces the special case of constraining the node I to move exactly in step with an atom α at the same position. Other choices will see the displacement of node I take on some weighted-average of the displacements of atoms in its immediate vicinity.

[14]For convenience the CACM developers [10] chose to fill the entire body with a finite element mesh that overlaps the atomistic region similar to fig. 5. Because of the local nature of the finite element energy functional, elements deep inside B^A that do not overlap the padding region do not affect the solution and so the formulation is equivalent to fig. 2(a). The overlapping elements are convenient, however, to simplify the mesh building process, since there is neither a need to mesh a non-convex region containing a "hole" in which the atoms reside, nor the need for a complicated fine-scale mesh around the atomistic/continuum interface.

Given this scheme to find the FE nodal displacements within B^A from the atomic displacements, the nodes within B^A are held fixed and the energy of the continuum region is minimized without any consideration of the atomistic energy. Once a minimum is found, the atoms lying in the padding region are positioned according to the current displacement field in the finite elements (in this instance, CACM uses strong compatibility through eqn. (1), so that CACM combines characteristics of strong and weak compatibility). These padding atoms are then fixed, and the energy of the atomistic system is minimized, again without regard for the continuum energy. This leads to a new set of atomic positions, which are used to find new displacements for the nodes inside B^A, and the process repeats until the change in atomistic energy during an iteration is below a specified tolerance. This type of an approach is often referred to as an alternating Schwartz method in the mathematical literature.

The main advantage of this approach is its modularity. In principle, it can be implemented with a simple interface between two fully independent codes: a lattice statics code and a finite element code. The coupling can be treated as a post-processing step on the solution from each iteration, and since most of the time will be spent in minimizing the two energy functionals, the implementation of the coupling need not even be especially efficient. However, the main disadvantage of this method is that it will be extremely slow to converge for problems that are nonlinear. In section 7, we will clearly illustrate this slow convergence.

The developers of CACM also included the ability to adapt the size and shape of the atomistic region on-the-fly.

4.6 Cluster-based Quasicontinuum (CQC(m)-E)

If we return to the discussion of the QC method from section 4.1, we can follow a different path starting from eqn. (18). Recall that in the QC method, the step following eqn. (18) was to divide the body into two regions that would ultimately be treated differently; one where the energy per atom is computed using the fully-atomistic approach (the "nonlocal region") and one where the energy was found using the Cauchy-Born rule (the "local region"). We will next describe a different approach that was originally posited as a force-based method in reference [24] (see section 6.4, later), and later re-cast as an energy-based method [15].

The idea is illustrated in fig. 6. As outlined before, we select a handful of the atoms to also act as nodes, and these are connected by a FE mesh. The atoms between the nodes are constrained to move according to the interpolated finite element displacement field. The CQC(m)-E method[15] now posits that the energy of *all* of the atoms can be estimated by only computing the energy of a handful of atoms around each node. These are the atom "clusters" highlighted in the figure. The number of atoms in the cluster, m, is a parameter of the model, with larger clusters improving accuracy while increasing the computational effort. This is the reason for the notation (m) in our acronym for the cluster methods. Where it is necessary, we will indicate the cluster size for a given model. For example, CQC(13)-E indicates an energy-based cluster method with each cluster containing 13 atoms.

The energy of eqn. (18) can then approximated as follows. The energy of a single

[15] CQC(m)-E refers to the energy-based cluster quasicontinuum method. In section 6.4, we discuss the original force-based CQC(m)-F method.

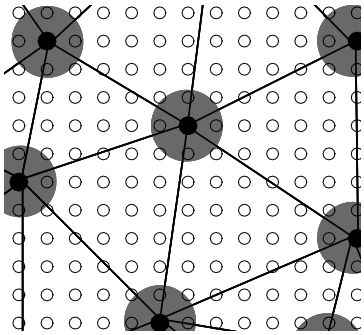


Figure 6: Clusters of repatoms around each node used to compute the energy in CQC(m)-E.

node/repatom I is found by computing the average energy of all the atoms in its cluster, the set of which are denoted $\mathcal{C}^I$:

$$\bar{E}^I(\mathbb{U}) \approx \frac{1}{m^I} \sum_{\alpha \in \mathcal{C}^I} E^\alpha(\tilde{\boldsymbol{u}}[\mathbb{U}]), \tag{28}$$

where m^I is the number of atoms in cluster I. Based on the discussion so far, the superscript on m seems unnecessary since we have suggested that all clusters are the same size. In fact, in highly refined regions (and especially the atomistic region) there will be overlap of the clusters for any $m > 1$. This is resolved by assigning atoms only to their nearest repatom, so that no atom is part of more than one cluster (ambiguous cases that are equidistant from several repatoms are resolved randomly). As such, some repatoms near the atomistic/continuum interface will have $m^I < m$, and all of the atomistic region will have $m^I = 1$ regardless of the nominal choice of m for the model.

The energy of atom α in the cluster, E^α, depends on the positions of all atoms, but these are constrained to move as dictated by the nodal displacements $\mathbb{U}$ as in eqn. (1). The total energy of all the atoms in the body is now approximated by a weighted sum of these repatom energies:

$$\mathcal{E}_{tot}(\mathbb{U}) = \sum_{I \in \mathcal{R}} n^I \bar{E}^I(\mathbb{U}) = \sum_{I \in \mathcal{R}} \frac{n^I}{m^I} \sum_{\alpha \in \mathcal{C}^I} E^\alpha(\tilde{\boldsymbol{u}}[\mathbb{U}]), \tag{29}$$

where $\mathcal{R}$ denotes the set of atoms that are selected to be repatom/nodes and n^I is a weight assigned to each repatom. The weights n^I can be chosen in a variety of ways, and can be thought of as the number of atoms whose energy is *represented* by repatom I. For example, one might draw the Voronoi diagram of the repatoms and then count or otherwise approximate the number of atoms falling within the cell of repatom I. Other schemes for computing n^I are possible, as well as different weighted-averaging techniques to compute $\bar{E}^I$ from the cluster atom energies (eqn. (28)). These differences do not strongly effect the behaviour, speed or accuracy of the method, although a sensible restriction on the selection

of the n^I values is that

$$\sum_{I \in \mathcal{R}} n^I = N_{\mathrm{A}}, \tag{30}$$

where N_{A} is the total number of atoms in the entire body.

In regions where the distance between repatoms becomes small, the method naturally goes over to the fully-atomistic limit if we insist that clusters cannot overlap. Instead, they shrink to $m^I = 1$ in the fully-refined limit, where the repatom weights become $n^I = 1$ as well. In that case, it is clear that eqn. (29) reduces to the lattice statics energy expression.

Although it is no longer as apparent, there is still a division between "atomistic" and "continuum" regions in this model. In essence, repatoms with $n^I = 1$ constitute B^A, whereas B^C is spanned by elements touching repatoms with $n^I > 1$. In B^C, the continuum assumption is not in the form of a new constitutive law, since the energy $\bar{E}^I$ for repatoms in this region is still obtained from a (cluster-averaged) fully-atomistic energy calculation subject to the constrained deformation $\tilde{\boldsymbol{u}}(\mathbb{U})$. But there is still a continuum assumption made here, in that the method assumes a smooth variation of the energy from one atom to the next.

For clusters of size $m^I = 1$ in coarsely meshed regions, the accuracy of CQC(1)-E is quite poor, and it is necessary to use larger clusters or a ghost-force correction technique as shown in [15]. We will discuss this at more length in subsequent sections.

4.7 Ghost Force Correction Methods

Several methods have been proposed to either eliminate or at least mitigate ghost forces. The first method proposed to eliminate them in a fundamental sense from an energy-based formulation was the creation of so-called "quasi-nonlocal atoms" by [49]. These atoms occupy a transition region between B^A and B^C, and are designed to interact differently with the two regions. The approach is limited to certain crystal structures and certain orientations of the atomistic-continuum interface within the crystal. Later, a more general reconstruction scheme was proposed in [14], but it is still not clear how to generalize this approach beyond a planar interface within a crystal. Furthermore, the goal within many coupled methods to include automatic re-meshing and adaption of the model to respond to error measures or changes in the deformation field make these approaches expensive.

Recently, Klein and Zimmerman [23] have proposed a generalization of the coupling used in the static BSM, including a careful accounting of the causes of ghost forces and ways to minimize them (but not completely eliminate them). Currently, that method is limited to pair potential atomistic models.

4.7.1 Deadload Ghost Force Correction

In the original implementation of the QC method where ghost forces are discussed [46], an approximate ghost force correction is proposed. It turns out that this approach is very general; it will work within the framework of any method for which the ghost forces can be analytically derived or computationally estimated. As we shall see in our quantitative example, this correction seems to go a long way in improved accuracy with almost negligible extra computational effort.

The proposed ghost force correction is to explicitly compute the ghost forces in some suitable configuration, and then add the negative of these forces as deadloads on the affected atoms or nodes. Thus, defining $\boldsymbol{g}^I$ as the ghost force experienced by atom or node I (which could be zero if the atom/node is far from the continuum-atomistic interface), then the energy functional is modified from eqn. (21) to

$$\Pi^{tot} = \sum_{\alpha \in B^A} E^\alpha(\boldsymbol{u}[\mathbb{U}]) + \sum_{e \in B^C} n^e \Omega_0 W(\boldsymbol{F}^e[\mathbb{U}]) - \sum_{I=1}^{n_R} \boldsymbol{g}^I \cdot \mathbb{U}^I. \tag{31}$$

For an undeformed model this exactly eliminates the ghost forces by construction. It becomes less clear how to proceed when the model is non-uniformly strained, for two reasons. First, it may not be possible to clearly identify which internal forces are "ghost" and which are "real". Second, the ghost forces will not remain constant as the deformation changes, and so the dead load assumption is only an approximation (with an indeterminate error associated with it).

To deal with the first difficulty, the QC developers *defined* the ghost forces in an intuitive way that ensured that the method will have no ghost forces in the undeformed configuration while still providing a clear prescription for ghost force computation under general deformation. This "ghost force" is defined as follows: For an atom whose energy is explicitly included in the total energy (atoms included in the first sum of eqn. (21)), the ghost force is defined as any force the atom would not feel if its environment was truly atomistic everywhere. At the same time, ghost forces on the nodes defining the continuum are those forces that the nodes would not feel if their environment were truly just the continuum. When implementing the forces, it then becomes clear which contributions are the ghost forces. This prescription is closely related to the "force-based" CADD/FEAt method to be discussed in section 5. Essentially, the differences between the forces used in the force-based methods and the actual derivative of the energy functional in the QC/CLS method are defined as the ghost forces. Using this same guiding principle, it is possible to incorporate a ghost force correction within any of the other existing methods. Indeed, we shall discuss next how this was adopted within the cluster-based QC in [15] and how we propose a similar correction to the AtC force-based method in section 6.3.1.

The second difficulty can be treated by occasionally re-computing the ghost forces, effectively changing the energy functional that is being minimized from time to time. In the limit of continuously updating the ghost forces (say for every evaluation of the forces in an energy minimization or dynamic simulation), one is no longer minimizing any energy functional at all, since the energy functional is continuously changing. Instead, one has moved to the realm of the "force-based" methods that we will discuss in section 5. Note that CGMD has the desirable feature that the ghost forces are constant, therefore in this method the second difficulty is not an issue.

4.7.2 Deadload ghost force correction in CQC(m)-E

It has been recently demonstrated in reference [15] that the ghost force correction via constant deadloads proposed for the original QC method can also be efficiently applied to the CQC(m)-E. They recognized that large cluster sizes are required for good accuracy in CQC(m)-E, but

that it is these large clusters which significantly slow down the method. However, one can get good accuracy from small clusters (even from "clusters" of only one atom at each repatom site) if a ghost force correction is used.

Rather than try to write down analytic expressions for the ghost forces, which is exceedingly complex in the CQC(m)-E case, an approximation was proposed in reference [15], as follows. For a given cluster radius, r_{clust}, around each repatom site, and a given model configuration (characterized by the repatom displacements $\mathbb{U}$), the forces on each repatom, $\boldsymbol{f}^{\text{CQC}-\text{E}}$, can be computed. The correct forces, denoted $\boldsymbol{f}^*$, are unknown, but would be the forces computed in the limit of "infinite" cluster radius, meaning that every atom is included in the energy calculation. To compute an *approximate* $\boldsymbol{f}^*$, one can recompute the forces for a given model configuration but with a cluster-radius slightly larger than r_{clust} (such that it includes the next neighbor-shell of atoms for the crystal, for example) . The ghost forces are then defined as the difference between the two sets of forces:

$$\boldsymbol{g} = \boldsymbol{f}^{\text{CQC}-\text{E}} - \tilde{\boldsymbol{f}}^*, \tag{32}$$

where $\tilde{\boldsymbol{f}}^*$ is the approximation to the correct forces. The forces $\boldsymbol{g}$ can then be added as deadloads, exactly as in eqn. (31).

This is an appealing approach in that the ghost force calculation makes use of the same machinery as a regular force calculation; it only requires a slightly expanded list of cluster members and appropriate weight functions. In section 7 we will quantify the extent to which this correction improves the speed and accuracy of the CQC(m)-E model.

5 The Force-Based Formulation

The existence of ghost forces is, it seems, a necessary consequence of having a well-defined energy functional. An alternate approach is to abandon the energy-based approach and instead start from forces directly. As we show in this section, methods of this type can indeed eliminate the ghost forces. However, the lack of a well-defined energy can be problematic for a number of reasons:

- If the forces are not constructed carefully, it makes the search for equilibrium more difficult. In the worst case, one will not find a solution at all if the forces are highly non-conservative[16]. Assuming a stable model is built, one may still find "equilibrium" solutions that are not physical because they correspond to saddle points or maxima (rather than minima) on some unknown energy surface.

- The lack of an energy functional also means that force-based methods cannot be used to compute the difference in energy between different equilibrium states or the activation energies along transition paths that are often the goal of static methods.

- Another difficulty that is not often mentioned, but can be cause for practical concern, is that debugging force-based schemes can be more difficult since the workhorse of

[16]The forces in an energy-based method are always conservative by construction, but in a force-based method one often deliberately constructs a non-conservative set of forces to avoid the ghost force problem.

energy-based method debugging, the numerical derivative test, may not be applicable in the absence of an energy functional.

- In dynamic systems, the possibility of an unstable, non-conserving system is grounds for considerable caution. For example, a force-based dynamical method cannot simulate a system in the microcanonical ensemble since the energy of the system will not be conserved.

In essence, a force-based method is based on the following philosophy: To eliminate ghost forces, *design* the method so that the forces are identically zero when the perfect crystal is in its correct equilibrium state. Since it does not seem possible to do this in general using an energy functional except for certain special forms of the interatomic potential, we construct forces directly without recourse to a total energy.

Superficially, it would seem that this is no different than choosing to couch things in terms of Hamiltonian or Newtonian mechanics. In the former, forces are obtained by differentiating an energy, whereas in the latter forces are used directly. However, the Newtonian and Hamiltonian formulations are only equivalent for a *conservative* system, and therein lies the rub. A force-based multiscale formulation is not conservative exactly because a total energy functional cannot be defined. The forces are constructed *ad hoc* and as such there is a danger that the resulting system will be potentially ill-behaved or unstable. The developers of force-based methods hope that by building force schemes that are physically-motivated, they are taking liberties that are not too terribly egregious.

Let us consider, again, the coupled problem shown in fig. 2(b), with the left-hand region modeled atomistically and the right-hand region treated using finite elements. We can approach the problem of how to construct a force-based coupled model by imagining that two independent potential energy functionals exist: one that treats the entire body atomistically, Π^{atom}, and one that models the entire body as a continuum using finite elements, Π^{FE}.

For Π^{atom}, we imagine that the atoms underlying the continuum region are in positions determined by their reference crystal structure and the displacement field of the finite elements. We could then, in principle, compute the potential energy of this atomic configuration. Differentiation of this potential energy with respect to the position of any one of the real atoms gives us the force on this atom. In practice, the finite range of interaction used to define an interatomic potential means that we only need to worry about a padding of atoms in the continuum near the atomistic/continuum interface, as discussed previously.

At the same time, forces on the finite element nodes are computed by starting from the finite element energy functional, Π^{FE}. To this end, we can now imagine that the entire body is modeled by finite elements, with the actual continuum region using the same elements as defined for the coupled model. Assuming a strictly local finite element formulation, the details of the finite element mesh inside the atomistic region are not important, since the force on a node is computed entirely from elements in direct contact with it. In other words, we only need to know the positions of the handful of atoms defining the displacement boundary condition for the elements near the atom/continuum interface.

Thus for atoms, the forces are defined as

$$\boldsymbol{f}^{\alpha} = \frac{\partial \Pi^{atom}}{\partial \tilde{\boldsymbol{u}}^{\alpha}}, \tag{33}$$

while for the nodes

$$\mathbf{F}^I = \frac{\partial \Pi^{FE}}{\partial \mathbb{U}^I}. \tag{34}$$

Note that this is not the same as minimizing the combined energy functional $\Pi^{atom} + \Pi^{FE}$. The forces computed in eqns. (33) and (34) cannot be derived from a unified energy functional. But the advantage is that the ghost forces will be zero by design.

In the following sections, we look at the details of several existing force-based methods. We will see that even some force-based methods, with what seem to be very reasonable prescriptions for the forces, can have their own spurious effects that are similar to the ghost force artefacts that plague energy-based implementations.

6 Review of Force-Based Coupling Methods

6.1 FEAt and CADD

The earliest force-based method (and, indeed, the earliest of all the methods presented here) was the Finite Element-Atomistic (FEAt) method of [25]. In this approach, there is no handshake region and strong compatibility is enforced, making it comparable to the QC and CLS methods but with a force-based rather than energy-based governing formulation. FEAt took an additional step of introducing a nonlocal elasticity formulation in the finite elements, in an effort to mitigate the abrupt transition from a local continuum to nonlocal atoms.

More recently, the same force-based coupling was used in the development of the Coupled Atomistic and Discrete Dislocations (CADD) method [47, 48]. The focus of this development was the connection to discrete dislocation methods in the continuum region. This focus on dislocations necessitates the use of linear elasticity (as opposed to the Cauchy-Born (CB) rule) in the continuum region, but in the limit where there are no dislocations in the continuum, CADD can use the CB rule or any other nonlinear constitutive law in the finite elements. In this limit (no dislocations and the CB rule) CADD shares the features of the QC method, but with a force-based rather than energy-based coupling scheme. In fact, CADD can be described as a force-based QC formulation, which is how it is sometimes referred to in the literature.

The CADD/FEAt coupling is therefore simple to describe in relation to the QC method as presented in section 4.1. There is no handshake region and strong compatibility sets the positions of the padding atoms and the nodes along the heavy, jagged line in fig. 2(b). The *forces* on every atom in B^A are computed as if the continuum did not exist, from the derivative with respect to atom positions of an energy functional:

$$\Pi^{A\cup P} = \sum_{\alpha\in\{B^A\cup B^P\}} E^\alpha - \sum_{\alpha\in\{B^A\cup B^P\}} \boldsymbol{f}^\alpha_{\text{ext}} \cdot \tilde{\boldsymbol{u}}^\alpha. \tag{35}$$

This is, in effect, the same as Π^{atom} introduced in the last section. Note that this energy functional is fundamentally different from eqn. (4) since it contains the padding atoms as well[17]. Similarly, the *forces* on the nodes are obtained from the derivative with respect to

[17]Note that force-based approaches also require a padding region which is twice as thick as that for

nodal positions of an energy functional:

$$\widehat{\Pi}^C = \sum_{e=1}^{n_{\text{elem}}} \sum_{q=1}^{n_{\text{q}}} w_q V^e W(\boldsymbol{F}(\boldsymbol{X}_q^e)) - \bar{\mathbf{F}}^T \mathbb{U}, \tag{36}$$

however without regard for the energy of the atoms. These forces are then used to move the atoms and nodes (either dynamically or towards equilibrium in a static solution using a suitably modified conjugate gradient method or quasi-Newton scheme) and the forces are re-computed for the new atom and node positions.

The strong compatibility employed in CADD/FEAt means that there is effectively a displacement boundary condition on the atoms in the form of a hard constraint on the motion of the padding atoms. Conversely, it means that there is a displacement boundary condition on the finite elements, as the last row of atoms impose their displacements on the first row of nodes. Note that this is not an iterative, alternating Schwartz type of an approach, even though the presentation may suggest it. The equations for the forces (*i.e.*, the selective derivatives of eqns. (35) and (36)) are solved simultaneously with the strong compatibility conditions as a constraint. Of all the methods presented here, only BSM and CACM of sections 4.4 and 4.5 respectively require repeated iterations between the atomistic and continuum domains.

6.2 The Hybrid Simulation Method (HSM)

Like QC and CLS, both FEAt and CADD require strong compatibility to be enforced in the interface region. Another force-based method that we will call the Hybrid Simulation Method (HSM) was proposed by [28]. It partially removes the strong compatibility condition of CADD/FEAt by generalizing the coupling to include a handshake region of finite width. There is still a padding region where strong compatibility between the atoms and the continuum displacement field is used, but atoms in the handshake region are free to move as they like. In fact, the purpose of the handshake region in this model is to provide a prescription for a weak compatibility scheme for the boundary condition on the FE region. As in the CACM method described earlier, this method relaxes strong compatibility by an averaging method using eqn. (26). In the original paper [28], the developers used a function g^I defined on a sphere of radius r_{av} surrounded node I and linearly decaying to zero at $r = r_{av}$. The size of the sphere could be varied to study the effects of this averaging, and of course other weighting functions could be used.

As with the BD method discussed earlier, the main advantage to this approach is that it makes mesh generation easier. Later in section 7 we will see that this approach tends to be less accurate than strong compatibility for a comparable mesh design.

We discussed earlier how the BD method could be shown to reduce to the QC or CLS method in certain limiting cases. Similarly, the HSM method reduces to the CADD/FEAt coupling method if we choose eqn. (27) for g^I, refine the elements to atomic spacing at

a comparable energy-based method. This is because the forces are derived from an energy functional that involves more atoms, as evidenced in eqn. (35). These extra atoms must be properly coordinated by additional padding, which means that force-based methods will be slightly slower than energy-based methods, all other things being equal.

the interface, and shrink the width of the handshake region to zero. In addition, the HSM method used nonlinear elasticity instead of the simpler linear version or the more expensive Cauchy-Born rule. It is straightforward, however, to exchange the constitutive model in any of these methods.

6.3 The Atomistic-to-Continuum (AtC) Method

The Atomistic-to-Continuum (AtC) method was presented and analyzed in a series of papers, namely references [19, 35, 5, 4]. This method is, in essence, a force-based version of the BD method discussed earlier. Recall that the BD method used eqn. (2), with the energy of the handshake region coming from the blending of a continuum and atomistic energy via eqn. (6). The AtC method achieves its coupling by blending at the level of forces, as follows.

The derivation starts by assuming that the atomistic and continuum regions are completely uncoupled, even though they physically overlap in the handshake region. The forces on the nodes and the atoms are

$$\mathsf{F}^I = -\sum_{e=1}^{n_{\text{elem}}} \int_{B_e} \boldsymbol{P}(\widehat{\boldsymbol{F}}(\mathbb{U})) \frac{\partial N^I}{\partial \boldsymbol{X}}\, dV \tag{37}$$

$$\boldsymbol{f}^\alpha = \sum_{\beta \neq \alpha} \boldsymbol{f}^{\alpha\beta}. \tag{38}$$

Here F^I is the force residual on node I, $\boldsymbol{P}$ is the first Piola-Kirchhoff stress tensor obtained from the FE constitutive law, and N^I is the shape function of node I. The numerical approximant to the deformation gradient $\widehat{\boldsymbol{F}}$ is obtained from the finite element displacement field. In the second equation, $\boldsymbol{f}^\alpha$ is the force on atom α and $\boldsymbol{f}^{\alpha\beta}$ represents the force exerted by atom β on atom α. We have omitted the possibility of any externally applied forces, just to simplify the discussion somewhat.

The forces between atoms are gradually weakened across the handshake region from the atomistic to the continuum side, using a weight function, η, that linearly decreases from 1 to 0. The weight for atom α is

$$\eta^\alpha = \eta(\boldsymbol{X}^\alpha). \tag{39}$$

Then, the force between two atoms α and β is weakened by a factor

$$\eta^{\alpha,\beta} = \frac{\eta^\alpha + \eta^\beta}{2}, \tag{40}$$

so that the atomic forces become

$$\boldsymbol{f}^\alpha = \sum_{\beta \neq \alpha} \eta^{\alpha,\beta} \boldsymbol{f}^{\alpha\beta}. \tag{41}$$

The symmetric definition of $\eta^{\alpha,\beta}$ ensures that Newton's third law is satisfied, *i.e.*, the weakened force exerted on atom α by atom β is equal to the weakened force exerted by β on α.

A complementary weight function[18], $\Theta = 1 - \eta$, is used to gradually weaken the finite element nodal forces across the handshake region from the continuum to the atomistic side:

$$\mathsf{F}^I = -\sum_{e=1}^{n_{\text{elem}}} \int_{B_e} \Theta(\boldsymbol{X}) \boldsymbol{P}(\widehat{\boldsymbol{F}}(\mathbb{U})) \frac{\partial N^I}{\partial \boldsymbol{X}} \, dV, \tag{42}$$

which can be integrated numerically by Gauss quadrature to yield

$$\mathsf{F}^I = \sum_{e=1}^{n_{\text{elem}}} \sum_{q=1}^{n_{\text{q}}} \Theta(\boldsymbol{X}_q^e) w_q V^e \left[-\boldsymbol{P} \frac{\partial N^I}{\partial \boldsymbol{X}} \right]. \tag{43}$$

Up to this point, there has been no coupling; all that has been done is to systematically weaken the forces through the handshake region using two complementary functions η and Θ. Two additional steps are required to effect the coupling. First, atoms *inside* the handshake region are constrained to follow the displacements of the finite elements (strong compatibility) via eqn. (1). Second, the forces computed on the handshake region atoms are projected to the finite element nodes. Consequently, the forces on the nodes become:

$$\mathsf{F}^I = \sum_{e=1}^{n_{\text{elem}}} \sum_{q=1}^{n_{\text{q}}} \Theta(\boldsymbol{\xi}_q^e) w_q V^e \left[-\boldsymbol{P} \frac{\partial N^I}{\partial \boldsymbol{X}} \right] + \sum_{\alpha \in \mathcal{S}^I} \sum_{\beta \neq \alpha} \eta^{\alpha,\beta} \boldsymbol{f}^{\alpha\beta} N^I(\boldsymbol{X}^\alpha). \tag{44}$$

By the compact support of the FE shape functions, a node only receives force contributions from atoms inside the elements contacting the node. We emphasize this by defining the set of atoms $\alpha \in \mathcal{S}^I$ as all atoms within the compact support of node I.

In the atomistic region, the atoms are free to move according to the dictates of the modified forces in eqn. (41). Since $\eta = 1$ everywhere inside B^A, the introduction of η will only have a relatively small effect, confined to atoms with neighbors in the handshake region where η is less than 1.

6.3.1 Spurious Forces in Force-Based Methods

Although the prescription of forces for the various force-based methods all appear reasonable, they are not all equally good. Later, in section 7, will will quantitatively compare them. For now, it is worth illustrating how the AtC method in particular leads to spurious forces that need to be corrected to achieve reasonable accuracy.

Consider the 1D chain illustrated in fig. 7 and modeled using the AtC prescription of forces just described. The atoms are shown at their equilibrium spacing, and the strain in the elements is zero everywhere, so there should be no forces on this initial configuration. For definiteness, imagine that the atoms interact via a 2nd-neighbor Lennard-Jones potential. Now consider an atom like the one labeled "3" in this figure. It interacts with atoms 1, 2, 4 and 5, so that the total force on atom 3 is zero in the equilibrium configuration:

$$f^3 = f^{13} + f^{23} + f^{43} + f^{53} = 0. \tag{45}$$

[18]Note that Θ is the same as the weight function introduced in the BD method earlier.

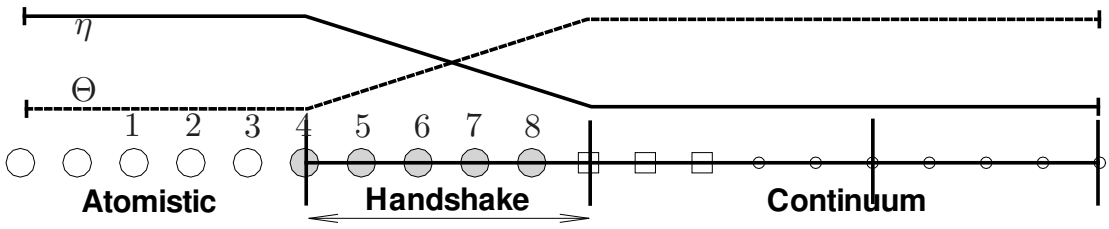


Figure 7: A 1D implementation of the AtC method. As in fig. 2, handshake region atoms appear as filled circles and padding atoms are open squares. Elements and nodes are indicated by heavy lines, while the graph above the model indicates the variation of the weighting functions.

Note that the sum of these interatomic forces is zero, but the four contributions are not zero on their own. In fact, the symmetry of the chain implies that

$$f^{13} = -f^{53}, \tag{46}$$
$$f^{23} = -f^{43}. \tag{47}$$

When we now weight the AtC atomic forces according to eqn. (41), we have

$$f^3 = \eta^{1,3} f^{13} + \eta^{2,3} f^{23} + \eta^{4,3} f^{43} + \eta^{5,3} f^{53}. \tag{48}$$

All of these weights will be equal to 1 with the exception of $\eta^{5,3}$, which is somewhat less than one due to the value of the blend function η at the position of atom 5. Using this result and eqns. (46)-(47), we are left with a non-zero force on atom 3

$$f^3 = (\eta^{5,3} - 1) f^{53} \neq 0. \tag{49}$$

Similar non-zero forces exist on atoms 4 through 8 in the handshake region, which are projected onto the nodes at either end. As such, there are non-zero forces throughout the interfacial region in this force-based method.

As we shall see in section 7, these forces seriously damage the accuracy of the AtC method. Fortunately, they can be approximately corrected using exactly the same deadload correction approach for ghost forces described in section 4.7.1. Any force present in the undeformed model is spurious, and so once these are computed they can be subtracted from the forces obtained in deformed configurations.

It is worth noting that the need for this correction is only obvious if a sufficiently complex interatomic model is studied. A harmonic model, even with second-neighbor interactions, is a special case where all forces $\boldsymbol{f}^{\alpha\beta}$ are identically zero for an undeformed perfect lattice. Since this is the model that was used in the development of the AtC method, the existence of these spurious forces was initially overlooked.

6.4 Cluster-based Quasicontinuum (CQC(m)-F)

In section 4.6, we discussed the energy-based variant of the Cluster-based Quasicontinuum method (CQC(m)-E). This approach can also be couched as a force-based method, and in fact the original formulation of reference [24] was a force-based model.

In the force-based CQC(m)-F method, clusters are chosen around each node, exactly as in the energy-based version. However, instead of approximating the energy of the missing atoms via the energy of the cluster atoms, the *forces* on the nodes are determined from the forces on atoms in the cluster. For a given configuration of the nodes, the interpolated displacement fields between the nodes determine the deformed positions of the cluster atoms and any adjacent atoms required to build neighbor lists. Then, the force $\boldsymbol{f}^\alpha$ on any cluster atom, α, in that constrained configuration can be determined. The force on node I follows as

$$\mathsf{F}^I = \sum_{J=1}^{n_{\text{nodes}}} n^J \sum_{\alpha \in \mathcal{C}^J} \boldsymbol{f}^\alpha N^I(\boldsymbol{X}^\alpha). \tag{50}$$

In words, this means the following. Each node J represents n^J atoms. The value of n^J can be determined in a number of ways [24], but one physically sensible approach is to assign all atoms within the Voronoi cell of node J to n^J as discussed earlier. Then a cluster of atoms is selected around each node J, denoted by the set of all $\alpha \in \mathcal{C}^J$. The force on each cluster atom α is computed and is multiplied by the weight factor n^J. Finally, this weighted force $n^J \boldsymbol{f}^\alpha$ is distributed amongst the nodes defining the element in which α lies, using the partition-of-unity shape functions $N^I(\boldsymbol{X}^\alpha)$. Writing this as a sum over all the nodes as in eqn. (50) is convenient notation, but we must remember that the compact support of the shape functions greatly limits the computational effort this entails.

The developers of CQC(m)-F observed that taking "clusters" of a single atom at each node leads to extremely poor results; the method is unstable. Making the clusters larger restores stability, but still leaves a certain level of error in the analysis. This can be controlled by making larger and larger clusters, but at the expense of increased computational cost. In the next section, we re-visit this more quantitatively.

7 Quantitative comparison between the methods

Method	Acronym	Key References	Continuum Model	Handshake	Coupling Boundary Condition	Governing Formulation
Quasicontinuum	QC	[53, 45] section 4.1	Cauchy-Born	None	Strong Compatibility	Energy-Based
Coupling of Length Scales	CLS	[44] section 4.2	Linear Elasticity	None	Strong Compatibility	Energy-Based
Bridging Domain	BD	[60] section 4.3	Cauchy-Born	Linear mixing of energy	Weak Compatibility (penalty)	Energy-Based
Bridging Scale Method	BSM	[59, 39] section 4.4	Cauchy-Born	None	Weak/Stong Mix (least-squares fit)	Energy-Based
Composite Grid Atomistic Continuum Method	CACM	[10] section 4.5	Linear Elasticity	None	Weak Compatibility (average atomic positions)	Iterative Energy-Based (two energy functionals)
Cluster-Energy Quasicontinuum	CQC(m)-E	[15] section 4.6	Averaging of atomic clusters	None	Strong Compatibility	Energy-Based
Ghost-force corrected Quasicontinuum	QC-GFC	[46] section 4.7.1	Cauchy-Born	None	Strong Compatibility	Energy-Based with dead load GFC
Ghost-force corrected Cluster-Energy QC	CQC(m)-GFC	[15] section 4.7.2	Averaging of atomic clusters	None	Strong Compatibility	Energy-Based with dead load GFC
Finite-Element/Atomistics Method	FEAt	[25] section 6.1	non-linear, nonlocal elasticity	None	Strong Compatibility	Force-Based
Coupled Atomistics and Discrete Dislocations	CADD	[47, 48] section 6.1	Linear Elasticity	None	Strong Compatibility	Force-Based
Hybrid Simulation Method	HSM	[28] section 6.2	Non-Linear Elasticity	atomic averaging for nodal B.C.	Weak Compatibility (average atomic positions)	Force-Based
Concurrent AtC Coupling	AtC	[19, 4, 5, 35] section 6.3	Linear Elasticity	Linear mixing of stress and atomic force	Strong Compatibility	Force-Based
Ghost-force Corrected Concurrent AtC Coupling	AtC-GFC	unpublished section 6.3.1	Linear Elasticity	Linear mixing of stress and atomic force	Strong Compatibility	Force-Based
Cluster-Force Quasicontinuum	CQC(m)-F	[24] section 6.4	Averaging of atomic clusters	None	Strong Compatibility	Force-Based

Table 1: Summary of the methods discussed in this presentation.

In table 1, we summarize the methods discussed in this study and their key features. The acronyms are meant to help keep the various methods straight. For example, we distinguish between the energy-based cluster method (CQC(m)-E) of section 4.6, the same method with a ghost-force correction (CQC(m)-GFC) of section 4.7.2 and the force-based version (CQC(m)-F) of section 6.4. Any method with a ghost force correction includes "GFC" in its acronym. Some of the acronyms are taken directly from the original papers, others are coined here.

In this section we discuss the results of implementing all of these methods in a single computer code [30] and directly comparing their performance on a universal test problem. The result of the study will be a quantitative comparison of the relative accuracy and speed of the different approaches.

One important question is the determination of what constitutes a suitable test problem. Making the problem too simple might hide problems or mask differences between methods. For example, we have already seen how using linear springs as the atomistic model hides spurious forces in the AtC method. As another example, it is possible to completely eliminate ghost forces in the CQC(m)-E method in 1D, but not in higher dimensions. On the other hand, making the problem too complex will make it difficult to analyze the results and perhaps cloud the picture of what constitutes the error. It is also important to choose a problem that is robust and has a well-defined unique solution, otherwise the comparison between methods becomes meaningless. We have chosen a problem that we believe satisfies these requirements. It is a robust problem that provides a suitably simple but not trivial test of the methods. The test is described below in section 7.2.

7.1 The computer implementation

The methods are implemented within the code framework of version 1.2 of the QC method code provided at the QC method website [51]. This framework allows us to study static 3D deformation problems in a 2D model. More specifically, the model is defined in the X_1X_2 plane, but the displacement field has three components. Furthermore, the underlying crystal structure is 3D (in our example it is face-centered cubic (fcc)), but periodic in the X_3-direction with the minimum possible simulation box length along X_3. This restricts, for example, dislocations to be infinite straight defects with their line direction along X_3, but at the same time allows their Burgers vectors to have both edge and screw components.

Our code, together with the input files and a brief explanation of its use, is available for download at the quasicontinuum website [51]. This may be useful for developers that wish to further explore comparisons between the methods or for someone trying to understand the details of the implementations, but it is not intended as a code capable of doing production runs. This is because our focus was on a code that was equitable to all the methods at the expense of overall efficiency. The code is set up so that all the methods in the comparison use exactly the same routines for force and energy calls, as well as the same neighbor finders, solution algorithms and convergence criteria. Any improvements to the efficiencies of these aspects of the methods could, in principle, be applied to all the methods and therefore leave their relative efficiencies unchanged. At the same time, someone looking to assess the overall efficiency of multiscale methods compared to fully atomistic calculations should take these results as a worst-case scenario; any of these methods can be made considerably more efficient

than our test implementation would suggest.

To compare accuracy, it is of course necessary that we use the same atomistic potentials for all the methods, and it would be nice to use the same continuum constitutive law as well. The finite element basis of the methods makes it easy to swap the constitutive law, and with the exception of the cluster-based methods (which have their own unique prescription for treating the "continuum" region) one can argue that no method is ideologically tied to a certain choice. As such, we will use the identical Cauchy-Born continuum constitutive law within all but the cluster-based methods. This will likely improve the accuracy of some methods, at the expense of their efficiency; if linear elasticity will suffice then the CB method is a waste of computational resources. We will return to address this specific point later, but for now a common constitutive choice ensures a fairer comparison between the methods.

One discrepancy between methods that cannot be eliminated completely is the difference between the force-based and energy-based formulations, which necessarily use different solvers. Here, we use the conjugate-gradient (CG) solver packaged with the QC code [51] to minimize the energy for all of the energy-based methods. We use the same CG solver for the force-based methods, modified to work without having an energy functional to minimize. Specifically, during a standard CG minimization, a series of 1D line searches are performed to minimize the energy along particular search directions. For the force-based CG solver (CG-FB), we replace the line search energy minimization with a line search to find the location along the search direction at which the dot-product of the search direction with the force vector is zero. The difference between the CG and CG-FB solvers means that a direct comparison of the efficiency of force-based and energy-based methods cannot be carried out. Instead, we have applied both solvers to the fully-atomistic problem that serves as the gold standard for this test. When discussing the efficiency of an energy-based or force-based method, we will compare it to the fully-atomistic simulation performed using the same solver.

Some of the methods offer flexibility in the choice of certain parameters during implementation which can, in principle, effect the accuracy or efficiency. To explore the effect of varying these myriad parameters is beyond the scope of this study. Rather, we have attempted to choose reasonable or, where known, optimal values. These details are elaborated here.

- BD Method. We found that the most efficient method to implement compatibility was to set $\beta_1 = 0$ and $\beta_2 = 10$ eV/Å^2 in eqn. (24).

- CQC(m)-E. Without the ghost force correction, all cluster calculations are carried out with a cluster comprised of the atom at the node and its nearest neighbors. This means (for the fcc example) 13-atom clusters of radius $r_{\text{clust}} \approx 2.9$Å, denoted CQC(13)-E. Repatom weights were computed using the volume of the Voronoi cell around each repatom, as discussed in the text following eqn. (29).

- QC-GFC. The deadloads used for the ghost-force correction in the QC (and any other method with such a correction) are computed once at the beginning of each load step, using the configuration of the atoms in the last relaxed step.

- CQC(m)-GFC. When the ghost force correction is applied to the cluster-based methods, the clusters are reduced to single atom clusters, *i.e.*CQC(1)-GFC. This greatly

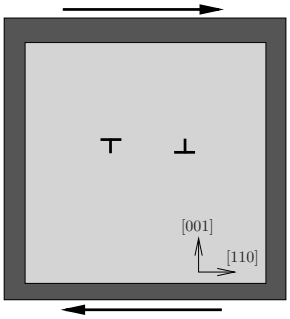


Figure 8: Test problem used to compare the various multiscale techniques. A Lomer dipole, 40Å wide, is centered in the model. The darker region around the edges of the model are held fixed to various levels of applied shear strain to force the dislocations to move.

enhances the speed of the method while only having a slight impact on the overall error. For the ghost force correction of eqn. (32), $\tilde{\boldsymbol{f}}^*$ is computed with a 13-atom cluster.

- HSM. Weak compatibility is enforced on the nodes adjacent to the atomistic region by applying eqn. (26) where

$$g^I(\boldsymbol{X}^\alpha) = \begin{cases} 1/N^I, & \text{when } |\boldsymbol{X}^\alpha - \boldsymbol{X}^I| < 2r_{\text{cut}} \\ 0, & \text{otherwise.} \end{cases} \tag{51}$$

  In other words, the displacement of the node is just the average of the displacement of all atoms within $2r_{\text{cut}}$ of the node.

- CQC(m)-F. The clusters for the force-based cluster methods are chosen to be 13-atom clusters (CQC(13)-F), as in the CQC(13)-E without the ghost force correction.

- CACM. The details of the convergence criteria for CACM are elaborated in section 7.5.1.

7.2 The test problem

The test problem is a block of single crystal aluminum[19] containing a dipole of Lomer dislocations [21], as schematically illustrated in fig. 8. The crystal is roughly 400Å$\times$ 400Å in the X_1X_2 plane and periodic in X_3 (with a periodic length of 2.85Å). Since the lattice constant for this model of aluminum is 4.032Å , this region contains $27,760$ atoms.

This problem is studied both "fully-atomistically" using lattice statics (we call this the "exact" solution), as well as with the various multiscale methods. For the multiscale models, we use meshes that are approximately the same across the methods. The meshes used are illustrated in fig. 9 and will be denoted by an extension "10", "20" or "30" as appropriate.

[19]Modeled using the EAM potentials of [16]; a sufficiently "multi-bodied" model to ensure a rigorous test of the methods.

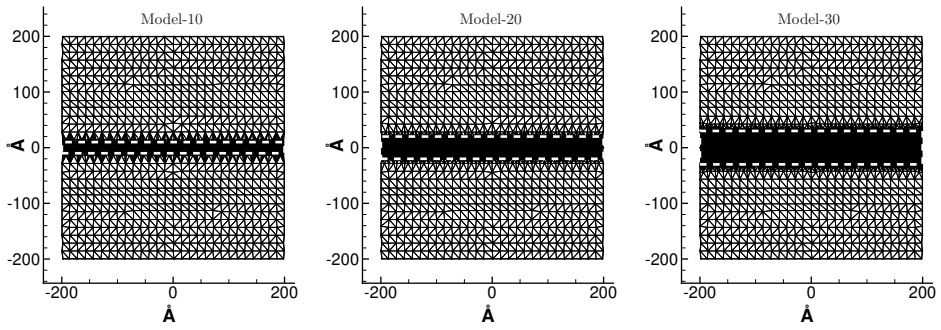


Figure 9: Meshes used in the test problem, whereby the width of the atomistic region (indicated by the dashed lines) is systematically increased.

The numbers indicate the extent of the fully-refined atomistic region along the middle of the model. For example, "10" means that the atomistic region extends from $X_2 = -10$Å to $X_2 = +10$Å. In all cases, the Lomer dipole lies on the $X_2 = 0$ plane, with the two cores initially at $X_1 = \pm 20$Å.

For models that do not require the mesh to be refined to the atomic scale along the atomistic/continuum interface (for example, the BD, BSM or HSM models), the atomic region covers the same extent as the other models (from $y = -10$Å to $y = +10$Å, for instance) and the continuum region uses a nearly uniform mesh of roughly the same size as the other methods away from the atomistic region, as shown in fig. 10. These coarse-grained elements typically contain about 34 atoms. The handshake region, if it exists, is outside of the fully-atomistic region as illustrated in the close-up of the central regions in fig. 10. The thickness of the handshake region is kept constant at approximately 10Å for all models (-10, -20, and -30).

We consider a dipole configuration rather than a single defect to make the application of the remote boundary conditions simpler, as the displacements around a dipole die-off rapidly to zero as we move away from the two cores. We study the Lomer dislocation, rather than the usual $\{111\}\langle 110\rangle$ dislocation in fcc because the Lomer is relatively sessile; with a Burgers vector $\boldsymbol{b} = [110]$ lying in the (001) plane, the Lomer dislocation has a large Peierls resistance to motion. This is an advantage because the dipole configuration remains stable without any externally applied stress, despite the elastic attraction between the two cores.

By applying a shear of the appropriate sign, we can introduce Peach-Koehler forces on the two cores that tends to drive them apart. At some critical level of shear, we expect the two cores to move from their initial configuration towards the outer edges of the crystal. Thus, there are two ways that we will quantify the error associated with each multiscale method. We first look at the relaxed configuration of this dipole under no applied shear, and quantify the discrepancy from the exact solution produced by a fully-atomistic model. Second, we study the shear levels at which the defects move and the range of motion, again in comparison with the exact atomistic result. It is interesting that there seem to be very few problems that are, like the shearing of the Lomer dipole, well-suited to the goals of this study. We considered, for example, what we thought to be a "simpler" problem of shearing a twin boundary contained a single step of one atomic plane (described in [54] and in the QC

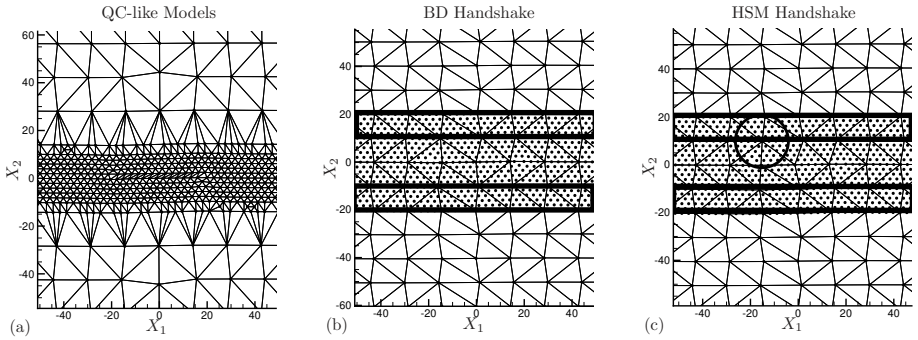


Figure 10: Close-up of the center of the "Model-10" meshes used. The fully-atomistic region extends from $y = -10$Å to $y = +10$Å in all cases. (a) For QC, CLS, FEAt, CADD and CQC, there is no handshake region and the padding atoms (not shown) lie within the continuum elements. (b) For the BD, BSM, CACM and AtC methods the handshake region extends an additional 10Å beyond the atomistic region as shown within bold lines. (c) For the HSM method, the same handshake region as in BD is used, and the averaging sphere to determine nodal positions is shown, actual size.

Tutorial Guide version 1.3 [52]). But it seems that such a step has multiple local minimum structures (much like the problem of modeling grain boundaries in general [40, 20]). Slight changes to the loading of the step (via changes to the multiscale model used) caused different structures to appear in different models. This made the quantifying of error difficult, since it is likely that even the exact result has multiple solutions.

The introduction of the dipole and the boundary loading are imposed in exactly the same manner in both the fully-atomistic model and all the multiscale models. As illustrated in fig. 8, we divide the model into the "free" region (light grey) and the "fixed" region (dark grey) along the boundary where the loading conditions are applied. The fixed region is of finite thickness (greater than $2r_{\text{cut}}$ for the potential) so that surface effects are avoided. The initial dipole configuration is created by applying the isotropic elastic displacement fields [21] to all the atoms in the free region and letting the structure relax. The dislocation field is not applied to the fixed region. This means that there is a slight initial strain on the dislocations, but it is only small since the dipole displacements rapidly decrease to zero in the far-field. In any case, since the same slight strain is applied to both the exact and model solutions, the comparison will remain valid. Shear strain is applied by prescribing the positions of the fixed region to be consistent with a uniform shear strain:

$$u_1 = \gamma X_2, \tag{52}$$

where u_1 is the X_1-component of the displacement and X_2 is the reference coordinate. For the first step, $\gamma = 0$ and the relaxation is only due to rearrangements of the dislocation core structure which is not exactly described by the elastic solution. Subsequent values of γ are then introduced by superimposing a uniform shear strain increment on the relaxed configuration of the previous load step. For example, after load step n, the atomic displacements are relaxed to $\tilde{\boldsymbol{u}}(n)$. As the boundary positions are incremented from γ_n to $\gamma_{n+1} = \gamma_n + \Delta\gamma$,

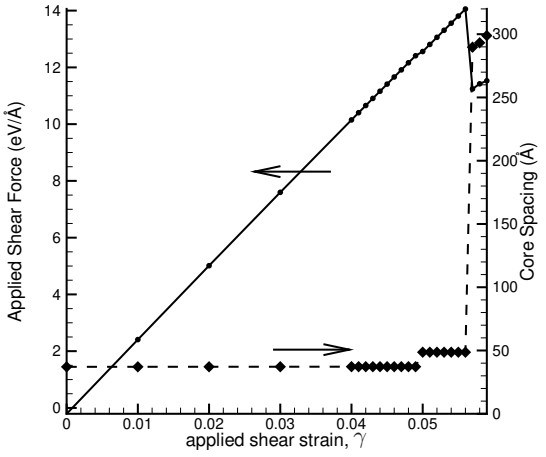


Figure 11: Applied shear load and distance between the dislocation cores as a function of applied shear strain for the exact atomistic result.

atoms and nodes within the free region are displaced in the X_1-direction according to

$$\tilde{u}_1^0(n+1) = \tilde{u}_1(n) + \Delta\gamma X_2, \tag{53}$$
$$\tilde{u}_2^0(n+1) = \tilde{u}_2(n), \tag{54}$$
$$\tilde{u}_3^0(n+1) = \tilde{u}_3(n). \tag{55}$$

From this initial guess, $\tilde{\boldsymbol{u}}^0(n+1)$, the free region is then allowed to relax to the solution, $\tilde{\boldsymbol{u}}(n+1)$ for step $n+1$.

7.3 The exact solution

Fig. 11 and fig. 12 illustrate the result of the fully-atomistic simulation; this will be the benchmark against which we compare the multiscale models. In fig. 11, the applied shear force and the distance between the two cores are plotted against the amount of shear strain, γ, that has been applied to the boundary. The shear force is defined as the sum of all X_1-components of force acting on the constrained nodes positioned above the $X_2 = 0$ line after relaxation. The core spacing is easily determined by identifying atoms with broken near-neighbor centrosymmetry [22] and defining the center of mass of these atoms as the center of the core. Points on the graph indicate the results from the actual load steps taken, which are joined by a line simply to guide the eye.

We see an essentially linear load-strain relation until $\gamma = 0.05$, at which point the dislocations move apart a relatively small amount; they increase their spacing from about 40Å to about 50Å. This results in a small but noticeable change in slope of the loading curve. Later at $\gamma = 0.057$, the dislocations move a long distance and the load drop is more significant. The dislocations never completely leave the crystal; the constrained atoms around the boundary act as a rigid wall against further dislocation motion.

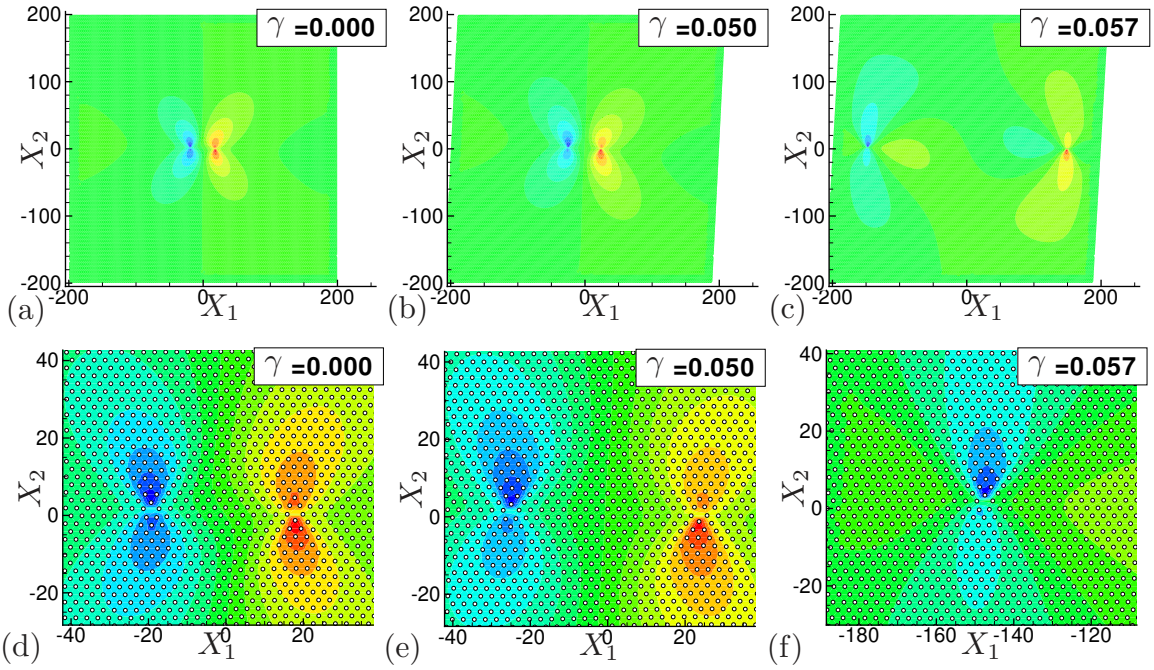


Figure 12: The exact solution for shearing of a Lomer dipole. (a)-(c) show the whole model, (d) and (e) show a close up of the dipole and (f) shows a close-up of the left-hand core after it has moved to the edge of the model. Contours show the X_2 component of displacement (range of contours: ± 0.72Å), merely to help visualize the dislocation.

Fig. 12 shows contour plots of the X_2-component of displacement at three points along the simulation. The top three images, (a)-(c), show the entire model while the lower images, (d)-(f), show a close-up of the dislocation cores with the actual atomic positions superimposed on the contours. In (d)-(f), the pentagonal core structure of the Lomer dislocations is clearly visible, and other than some elastic distortion as the core approaches the rigid boundary there is no significant change to this core structure as the dislocations move. Rather, the core simply moves from one Peierls valley to the next. Less clear from this image, but verifiable, is that the dislocations remain confined to a single (001) plane along $X_2 = 0$; they move only left or right on the plane where they were originally introduced.

7.4 Comparing the accuracy of multiscale methods

7.4.1 The Displacement Error

We first quantify the error in the results by comparing the displacement fields from the multiscale models to the exact atomic displacements. In the multiscale approaches, there is always a fully-atomistic region, where ideally we desire perfect agreement between the multiscale and exact results, and a coarse-grained or continuum region where the positions of most of the atoms are not explicitly tracked. However, given the reference position of an atom, we can obtain the displacement field values at that position and compare these to the true atomic displacements. We define an atom-by-atom error measure of the displacement error as follows. Denoting the exact displacement vector of atom α to be $\tilde{\boldsymbol{u}}^{\alpha}_{\text{exact}}$ and the displacement obtained from a multiscale model as $\tilde{\boldsymbol{u}}^{\alpha}$, we define the error as the L_2-norm of the difference between these two vectors

$$e^{\alpha} = \|\tilde{\boldsymbol{u}}^{\alpha} - \tilde{\boldsymbol{u}}^{\alpha}_{\text{exact}}\|. \tag{56}$$

The global displacement error for a given model is similarly defined as the L_2-norm of the difference between the global displacement vectors ($\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{u}}_{\text{exact}}$ are of length $3N_{\text{A}}$ for N_{A} atoms), normalized by the number of atoms (in this case, recall $N_{\text{A}} = 27,760$):

$$e = \sqrt{\frac{\sum_{\alpha=1}^{N_{\text{A}}}(e^{\alpha})^2}{N_{\text{A}}}} = \sqrt{\frac{\|\tilde{\boldsymbol{u}} - \tilde{\boldsymbol{u}}_{\text{exact}}\|^2}{N_{\text{A}}}}. \tag{57}$$

As defined here, both e^{α} and e have units of Å. To turn the global error into a percent-error, $e_{\%}$, in a sensible way, we can divide by the average of the atomic displacement norm in the exact solution:

$$e_{\%} = \frac{e}{\tilde{u}_{\text{avg}}} \times 100, \tag{58}$$

where

$$\tilde{u}_{\text{avg}} = \frac{1}{N_{\text{A}}} \sum_{\alpha=1}^{N_{\text{A}}} \|\tilde{\boldsymbol{u}}^{\alpha}_{\text{exact}}\| = 0.05755\text{Å}. \tag{59}$$

These percent-errors are presented in table 2, to show the scale of the errors for this particular test problem.

Table 3 presents the displacement error results again for each model, but normalized by the lowest value obtained, $e_{\text{min}} = 2.262 \times 10^{-3}$Å ($e_{\%} = 3.93\%$), from the CADD/FEAt-30

Method	Model-10	Model-20	Model-30
CADD/FEAt	11.42 %	5.13 %	3.93 %
QC-GFC	12.42 %	5.72 %	4.05 %
AtC-GFC	14.59 %	8.28 %	5.04 %
BD	14.63 %	9.09 %	6.44 %
QC/CLS	14.66 %	9.54 %	8.50 %
BSM	17.59 %	12.07 %	10.17 %
HSM	22.03 %	15.44 %	12.25 %
CQC(13)-E	22.59 %	16.62 %	16.55 %
CQC(1)-GFC	40.06 %	20.10 %	20.19 %
CQC(1)-E	86.75 %	43.61 %	38.48 %
CACM	42.59 %	40.13 %	39.92 %
CQC(13)-F	70.84 %	60.75 %	46.42 %
AtC	55.11 %	70.17 %	83.65 %

Table 2: Summary of the percent displacement errors, $e_{\%}$, for each method at the different levels of model refinement. All error values are using the definition of eqn. (58). Methods are explained in table 1.

Method	Model-10	Model-20	Model-30
CADD/FEAt	2.91	1.30	1.00
QC-GFC	3.16	1.46	1.03
AtC-GFC	3.71	2.11	1.28
BD	3.72	2.31	1.64
QC/CLS	3.73	2.43	2.16
BSM	4.47	3.07	2.59
HSM	5.61	3.93	3.12
CQC(13)-E	5.75	4.23	4.21
CQC(1)-GFC	10.19	5.11	5.14
CQC(1)-E	22.07	11.09	9.79
CACM	10.84	10.21	10.16
CQC(13)-F	18.02	15.45	11.81
AtC	14.02	17.85	21.28

Table 3: Summary of normalized global displacement error for each method at the different levels of model refinement. All error values are normalized as $e/e_{\min}$ where $e_{\min} = 2.262 \times 10^{-3}$Å was the best error value (obtained with model CADD/FEAt-30). Methods are explained in table 1.

model. The models are presented in order of decreasing accuracy based on the "Model-30" column. We see that all models except the AtC method appear to converge with increased model refinement, although the CACM model does so at an extremely slow rate. The ghost-force corrected AtC-GFC model restores convergence to the AtC approach. The best results come from CADD/FEAt and QC-GFC. Next are BD, QC/CLS, AtC-GFC and HSM that show fair accuracy, while the accuracy of the remaining models is relatively poor.

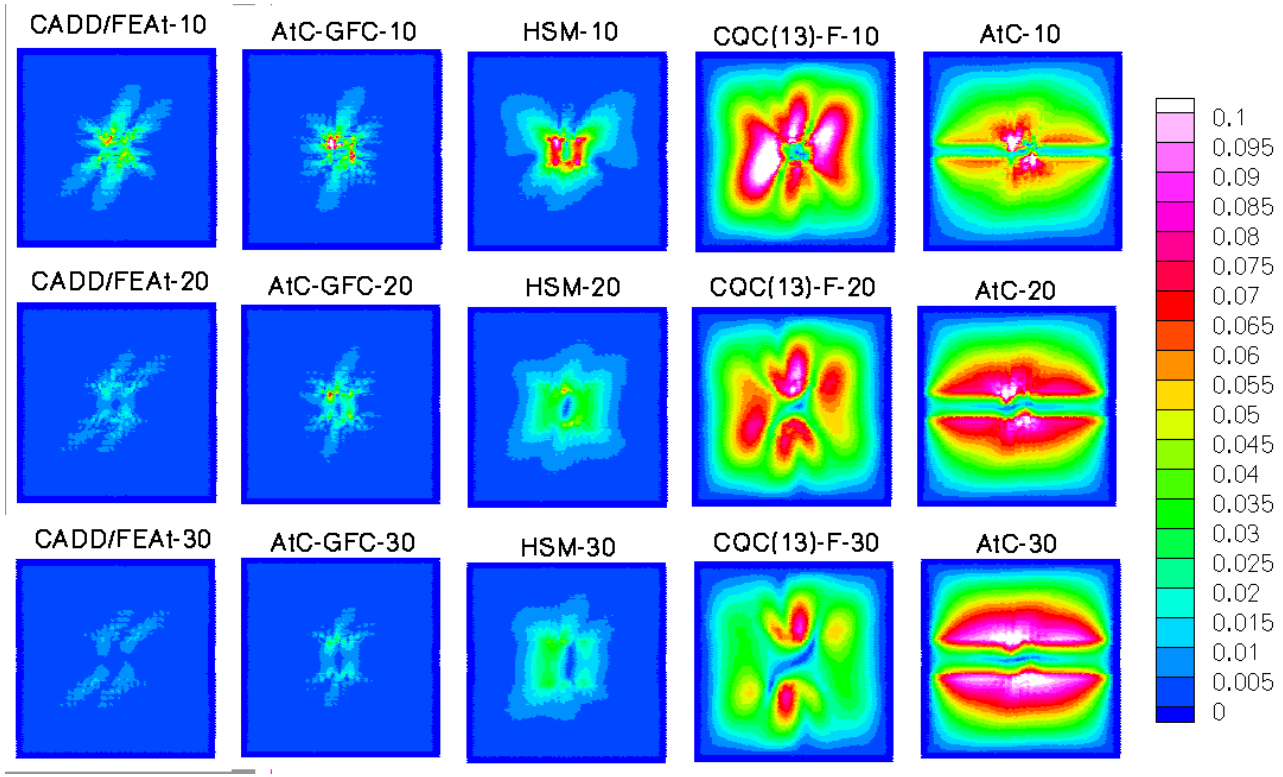


Figure 13: Plots of the atomic displacement error, e^{α}, in units of Å, for the force-based multiscale models. Each image is of the entire model, which can be seen in fig. 9.

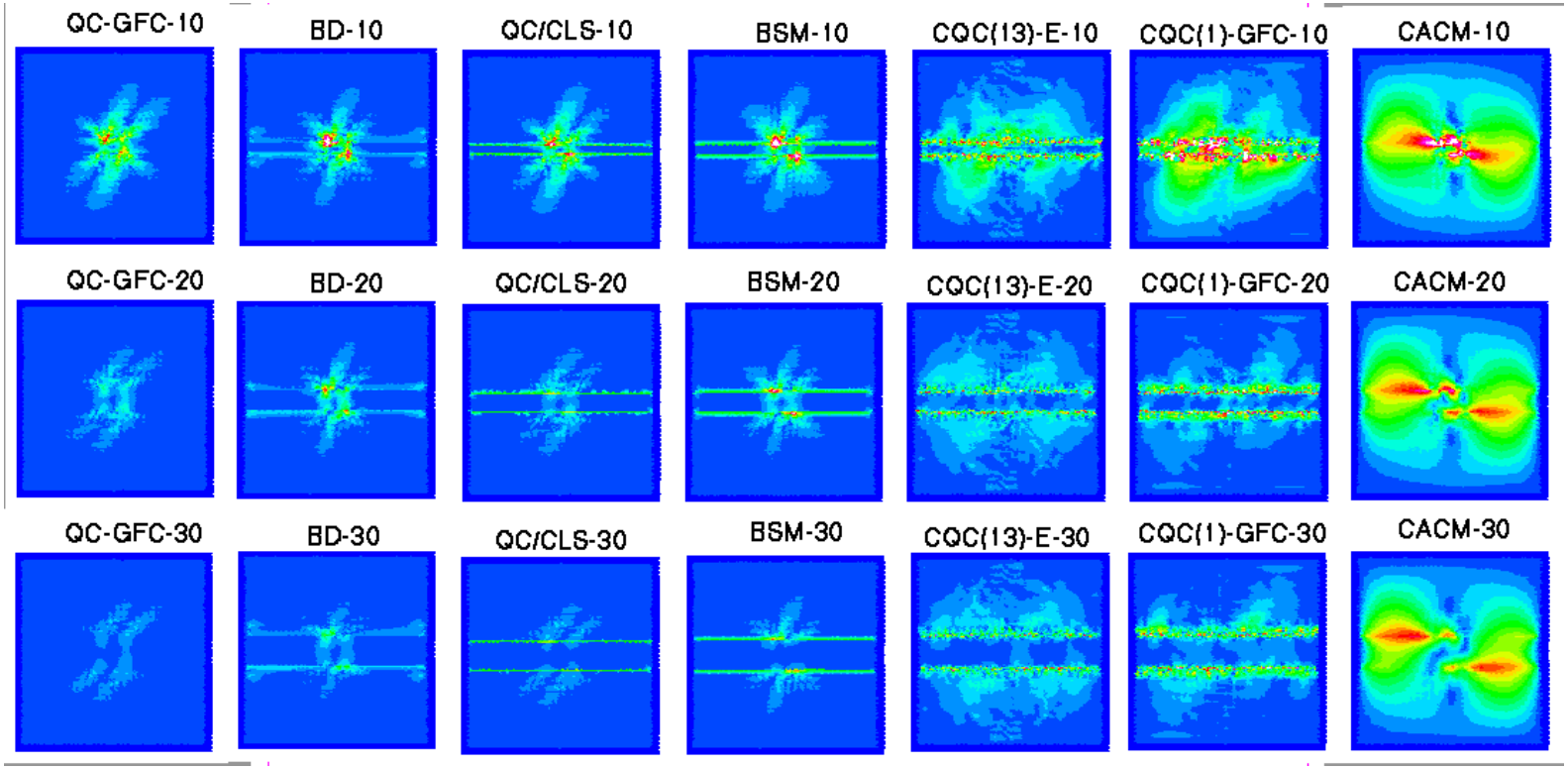


Figure 14: Plots of the atomic displacement error, e^{α} (Å), for the energy-based multiscale models. The contour legend from fig. 13 applies to this figure as well. Each image is of the entire model, which can be seen in fig. 9.

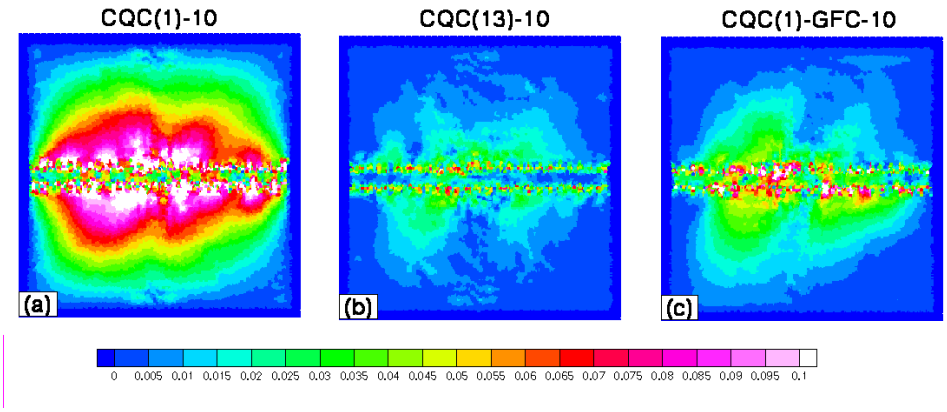


Figure 15: Comparing methods of improving the accuracy of the cluster-based CQC(m)-E model. From (a) to (b) is the effect of increasing 1-atom to 13-atom clusters. From (a) to (c) is the effect of applying the ghost-force correction to a model with 1-atom clusters.

Plots showing the magnitudes of the atomic displacement error, e^α, are presented in fig. 13 and fig. 14 to better understand the distribution of the error. All plots are for the crystal under no load, *i.e.*, $\gamma = 0$. We find that as the strain increases, there is very little change in the relative accuracy of the various models until the dislocations start to move. If, for a given model, the dislocations move at the wrong load step or move the wrong distance, this discrepancy dominates the error result. As such, the most useful comparison of the error plots is made at $\gamma = 0$. In both figures, each column shows the results for a specific method, while moving down the column shows the effect of refining the model used for a given method by expanding the atomistic region (*cf* fig. 9). Fig. 13 shows the results for the force-based methods, going from the most accurate to the least accurate as we move from left to right. The scale bar on the right applies not only to this figure, but to figs. 14 and 15 as well. Fig. 14 presents the results for the energy-based methods, again going from most accurate on the left to least accurate on the right.

Note that once we made the decision to use the same CB constitutive law for all models in this comparison, some of the methods became indistinguishable, hence the combined results for "QC/CLS" and "CADD/FEAt" are presented. A number of important observations are worth highlighting from these figures, as follows.

First, we clearly see the effects of ghost forces along the atomistic/continuum interface in the BD, QC/CLS, BSM, CQC(13)-E and AtC models. The band of error at ±10Å, ±20Å and ±30Å in each of the models respectively is the effect of spurious ghost force relaxations. The correction of the ghost forces with deadloads, as in QC-GFC, CQC(1)-GFC and AtC-GFC, can be seen to almost entirely remove this component of the error.

The results for the CQC(m)-E methodology requires further clarification. It appears from fig. 14 and table 3 that the ghost-force correction to CQC(m)-E (denoted CQC-GFC) does not improve the accuracy. In fact, this is because we have chosen to apply the ghost force correction to the model with 1-atom clusters around each node, whereas the uncorrected result is for a model with 13-atom clusters. We can see that the two approaches give similar error,

but we will see later that the the 1-atom cluster is considerably faster. For comparing the error, however, we should also consider the CQC(1)-E model *without* the ghost-force correction. In fig. 15 we compare the error distributions of models CQC(13)-E-10 and CQC(1)-GFC-10 with that of CQC(1)-E-10. In effect, this illustrates two ways of improving the CQC(m)-E accuracy: either by going to larger clusters (moving from CQC(1)-E-10 to CQC(13)-E-10) or by applying the ghost-force correction (moving from CQC(1)-E-10 to CQC(1)-GFC-10). The two strategies achieve comparable improvements to the accuracy. Increasing the cluster size is slightly more effective in this regard, but we shall see later that it is also considerably more expensive. Finally, we see that while both techniques mitigate the errors, they are still less accurate than many of the other approaches presented.

7.4.2 The energy Error

The displacement error presented in the previous section is useful for two principal reasons: it serves as a convenient relative measure of the error of the various methods and it provides a good way to visualize the distribution of the error throughout the system. However, it is not easy to assess the significance of this error; it is not easy to say what accuracy is "good enough" in these terms. For this purpose, we consider the error in the energy of the atomistic region in this section. We also look at the error in the dislocation motion predictions in the next section.

The energy of the entire system is only defined for the energy-based methods, but it is possible to compute the energy of just the atomistic region for all the methods provided that the underlying atomistic model can be unambiguously divided into an energy per atom (as is the case for EAM). We define two types of energy error, for reasons that will become clear shortly. We also examine the change in this error as the applied shear strain, γ, increases. In order to do this concisely we treat the energy of the atomistic region as a function of the applied strain and the Burgers vector, b, of the dislocations:

$$\tilde{\mathcal{E}}(\gamma, b) = \sum_{\alpha \in B^A} E^\alpha(\tilde{\boldsymbol{u}}[\gamma, b]). \tag{60}$$

Then we define the *error in the total atomistic energy* as

$$\epsilon(\gamma) = \left| \frac{\tilde{\mathcal{E}}(\gamma, b) - \tilde{\mathcal{E}}_{\text{exact}}(\gamma, b)}{\tilde{\mathcal{E}}_{\text{exact}}(\gamma, b)} \right|, \tag{61}$$

which is dependent on the level of applied strain, γ.

Another useful measure can be the error in the energy difference between two states. This is relevant, for example in computing transition paths or, in the context of our example, energy barriers to dislocation motion. The point of this measure is to assess whether a systematic error in a model (for example, due to ghost forces) has any profound effect on the energy *differences* which are often the most important quantity for a problem. For our example, the energy difference is effectively the core energy associated with the dislocation dipole, and we write the *error in the core energy* as

$$\epsilon_\Delta(\gamma) = \left| \frac{[\tilde{\mathcal{E}}(\gamma, b) - \tilde{\mathcal{E}}(\gamma, 0)] - [\tilde{\mathcal{E}}_{\text{exact}}(\gamma, b) - \tilde{\mathcal{E}}_{\text{exact}}(\gamma, 0)]}{\tilde{\mathcal{E}}_{\text{exact}}(\gamma, b) - \tilde{\mathcal{E}}_{\text{exact}}(\gamma, 0)} \right|. \tag{62}$$

Note that this is the error in the energy *difference* between a defected crystal and a defect-free crystal ($b = 0$) at a certain level of applied strain.

In table 4, we present the error in the total atomistic energy for the various methods at zero applied strain. Errors greater than 5% are shown in bold, and the methods are sorted in ascending order of the last column. Since this is for the problem at zero applied strain, the results correspond directly the the displacement error results previously discussed. The rather high error results, especially for the energy-based methods, can be attributed to the fact that the actual energy being computed here is rather small; it is essentially only the core energy of a closely spaced dipole. Any errors are likely to appear large relative to the small quantity being computed. By this error measure, the energy-based methods generally perform more poorly than the force-based methods unless they employ a ghost force correction. The presence of ghost forces in the energy-based methods leads to a systematic error in the energy that does not decrease with increased size of the atomistic region, because the spurious relaxations along the interface persist. This is most obvious in the QC method, and less so in methods like BD or BSM where some of the ghost force error is spread into the handshake region. The error in some of the cluster methods and the AtC method is less severe by this measure than by the displacement error result. This appears to be due to the fact that these methods allow a spurious rigid motion of the atomistic region that doesn't influence the energy error as strongly as the displacement error.

The systematic error due to the ghost forces is largely removed from the problem when we consider the core energy error, ϵ_Δ in table 5, where all of the methods exhibit very low error. This suggests that the errors associated with the atomsitic/continuum coupling for all the methods is relatively localized, and does not strongly influence the atoms deep in the atomistic region. However, this is for a special case where the is no applied loading and no long-ranged stress or displacement field associated with the interesting atomistic features (in this case, the dislocation dipole). The effect of loading and how well the atomistic/continuum interface transmits this loading is examined next.

In table 6, we return to the error in the total energy, $\epsilon(0.03)$, but now at an applied strain of $\gamma = 0.03$. This is about 2/3 of the strain at which many of the models exhibited dislocation motion (as discussed in the next section), and so it represents a substantial level of loading. Now the total energy in the atomistic region is considerably more due to stored strain energy, and therefore the relatively constant ghost-force energy error represents a smaller percentage of the energy being computed. At the same time, we are now able to assess the ability of the various methods to accurately transmit loading across the interface because the shear load is no longer zero. We see that by this measure, all of the methods show excellent accuracy (with the exception of the QC/CLS method without a ghost force correction). The CQC(m)-E methods exhibit curious behaviour in that there is a rather large fluctuation in the error as the model is refined. This might imply some type of instability in the cluster-based formulation. However, when "Model-30" is considered for these methods, the error levels are quite good.

For completeness, we also show the error in the core energy at $\gamma = 0.03$ in table 7. We see that with the exception of the CACM method, the conclusions do not change from those drawn from table 5. In essence, these methods do a good job of computing the core energy of the defects.

Method	Model-10	Model-20	Model-30
CADD/FEAt	3.5%	0.8%	0.3%
CQC(13)-F	1.1%	1.0%	0.4%
QC-GFC	3.8%	1.0%	0.4%
AtC-GFC	**7.3%**	2.2%	0.8%
CACM	**9.1%**	3.1%	0.9%
HSM	3.8%	1.3%	0.9%
BD	**10.6%**	**5.5%**	4.0%
BSM	**9.1%**	**6.0%**	4.8%
AtC	**13.1%**	**8.1%**	**6.9%**
CQC(13)-E	**5.9%**	**26.9%**	**7.6%**
CQC(1)-E	**5.1%**	**68.8%**	**8.6%**
CQC(1)-GFC	**11.4%**	**44.2%**	**10.1%**
QC/CLS	**23.9%**	**25.3%**	**29.6%**

Table 4: Summary of the error in the total atomistic energy at zero strain, $\epsilon(0)$, for each method at the different levels of model refinement. Methods are explained in table 1, errors greater than 5% are shown in boldface.

Method	Model-10	Model-20	Model-30
CQC(1)-GFC	1.3%	0.9%	0.0%
CQC(1)-E	0.6%	0.6%	0.2%
CQC(13)-E	1.7%	0.9%	0.2%
BSM	3.7%	1.2%	0.3%
CADD/FEAt	3.5%	0.8%	0.3%
QC/CLS	4.3%	0.9%	0.3%
CQC(13)-F	1.1%	1.0%	0.4%
QC-GFC	3.8%	1.0%	0.4%
BD	**6.7%**	2.0%	0.7%
AtC	**7.7%**	2.3%	0.8%
AtC-GFC	**7.3%**	2.2%	0.8%
CACM	**9.1%**	3.1%	0.9%
HSM	3.8%	1.3%	0.9%

Table 5: Summary of the error in the core energy at zero strain, $\epsilon_\Delta(0)$, for each method at the different levels of model refinement. Methods are explained in table 1, errors greater than 5% are shown in boldface.

Method	Model-10	Model-20	Model-30
CADD/FEAt	2.1%	0.4%	0.1%
QC-GFC	2.3%	0.5%	0.2%
AtC-GFC	4.4%	1.0%	0.3%
HSM	1.9%	0.8%	0.4%
CQC(13)-F	0.3%	0.8%	0.6%
BSM	**6.1%**	2.4%	1.2%
BD	**6.5%**	2.5%	1.5%
CQC(13)-E	1.5%	**11.0%**	1.8%
CQC(1)-E	1.8%	**28.6%**	2.1%
CACM	**8.2%**	3.7%	2.7%
AtC	**8.6%**	4.3%	3.0%
CQC(13)-GFC	**14.0%**	**18.2%**	3.5%
QC/CLS	**13.4%**	**10.7%**	**10.1%**

Table 6: Summary of the error in the total atomistic energy at an applied shear strain of $\gamma = 0.03$, $\epsilon(0.03)$, for each method at the different levels of model refinement. Methods are explained in table 1, errors greater than 5% are shown in boldface.

Method	Model-10	Model-20	Model-30
QC	4.3%	0.9%	0.2%
BSM	3.6%	1.0%	0.3%
CQC0	2.0%	0.1%	0.3%
CQC-GFC	**16.6%**	1.2%	0.4%
CADD	3.7%	1.0%	0.5%
QC-GFC	4.1%	1.2%	0.5%
AtC	**8.0%**	2.3%	0.5%
CQC-E	2.2%	1.5%	0.7%
BD	**7.1%**	2.2%	0.8%
AtC-GFC	**7.8%**	2.5%	1.0%
HSM	3.3%	1.9%	1.2%
CQC-F	0.5%	2.1%	2.1%
CACM	**14.5%**	**9.2%**	**8.9%**

Table 7: Summary of the error in the core energy at an applied strain of $\gamma = 0.03$, $\epsilon_\Delta(0.03)$, for each method at the different levels of model refinement. Methods are explained in table 1, errors greater than 5% are shown in boldface.

Method	Total time T, (sec)	Global overhead T_{OH}, (sec)	Timestep overhead t_{OH}, (sec)	Solve time T_s, (sec)	Force calls N_{force}	Time per force call $t_s = T_s/N_{\text{force}}$, (sec)
Exact	27182.71	0	0	27182.71	19193	1.42
QC/CLS	2885.68	102.9	1.76	2738.78	17066	0.16
BD	10478.58	17.27	0.29	10454.06	56251	0.19
BSM	2291.4	10.4	0.09	2278.84	14476	0.16
CACM	16062.42	17.27	0.29	16038.19	110903	0.14
CQC(13)-E	9038.42	150.09	3.47	8801.58	14098	0.62
QC-GFC	2624.63	103.23	1.73	2478.15	15285	0.16
CQC(1)-GFC	4886.57	356.6	3.47	4446.69	20032	0.22

Table 8: Raw time results for timing study of the energy-based methods. All calculations were performed using a single, 64-bit, 1.6 GHz Itanium processor on an 8-CPU Sun Altix 350 cluster running Linux. The main memory available to the system was 8 GB.

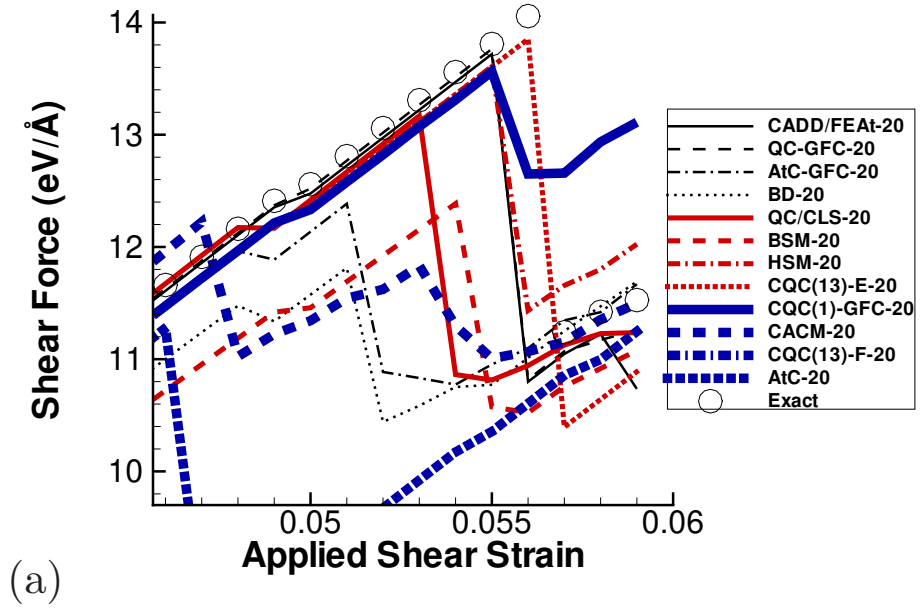
7.4.3 Errors in dislocation motion

Another test of accuracy is how well the multiscale methods reproduce the overall deformation behaviour of interest. For this simple test problem, the curves of shear load vs. applied shear and core separation as a function of applied shear are the most interesting physical results one may like to extract. The exact curves for these results were already presented in fig. 11, and we are now in a position to compare them to the multiscale results in fig. 16 and fig. 17. To render the figures comprehensible, we show only the "Model-20" results in (a) of both figures, and show the effect of model refinement for only two of the methods in (b). Not surprisingly, models for which we have already demonstrated a low accuracy are also inaccurate in predicting these curves. Most of this inaccuracy comes from the timing of the motion of the dislocations (*i.e.* the load steps at which these movements occur), and how far they move each time. This is most obviously reflected in the plots of core spacing, but it can also be seen in the location and depth of the load drops in the load vs. strain curves.

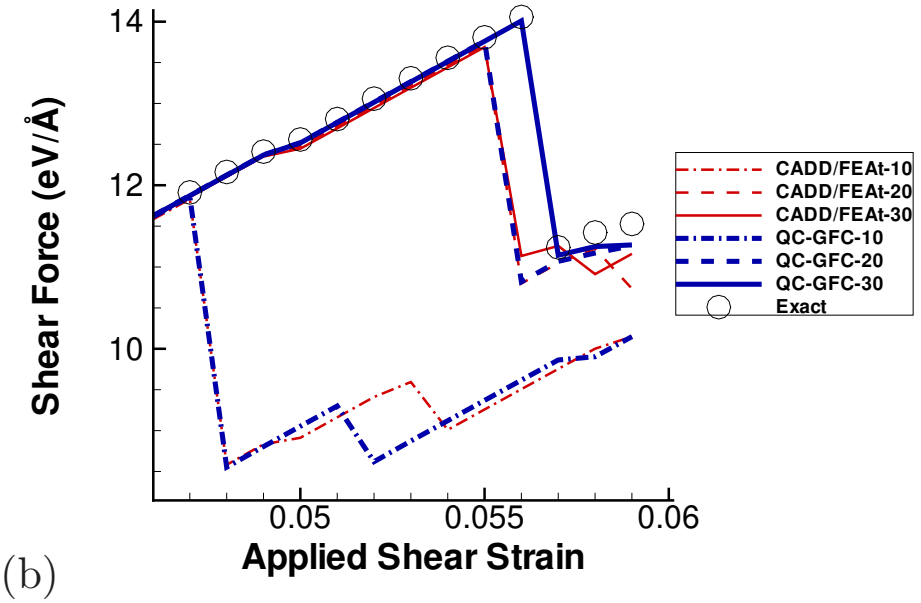
It is difficult to observe any general truths from these two figures, since there is no obvious trend in the results. It is worth noting that we are focusing on the last few load steps, since all of the models are reasonably correct up to the points shown. Perhaps the most important message that can be taken from these figures is that features like these are at the same time very difficult to reproduce *exactly* and relatively easy to reproduce *approximately*. Many of the models do a *reasonable* job of predicting load-strain curves and dislocation motion that are in qualitative agreement with the exact result. But atomistic systems are so complex, and their energy landscapes so complicated, that exact agreement from a multiscale model is unlikely to ever be achieved.

7.5 Comparing the speed of multiscale methods

The medium-sized model ("Model-20") is used here for a comparison of the efficiencies of the various methods, the results of which are summarized in table 8 and table 9. The tables

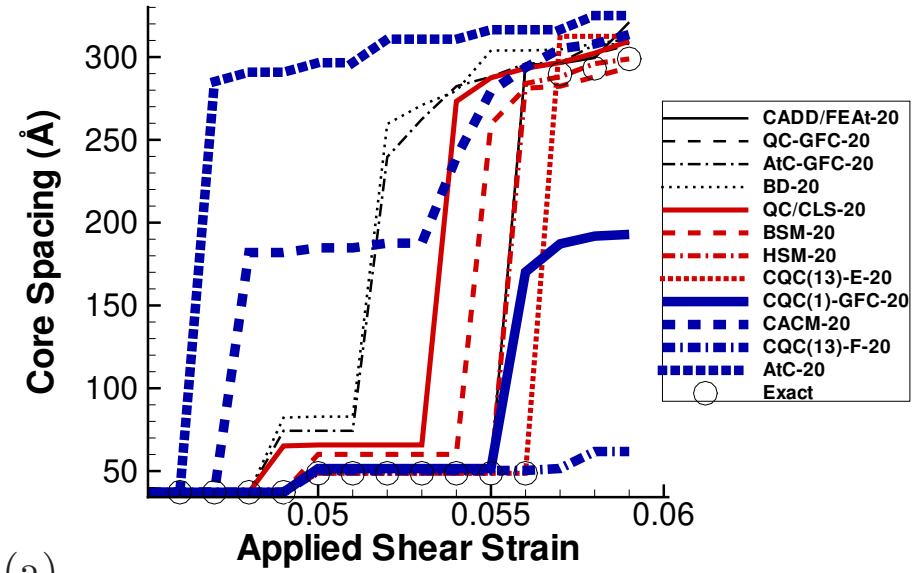


(a)

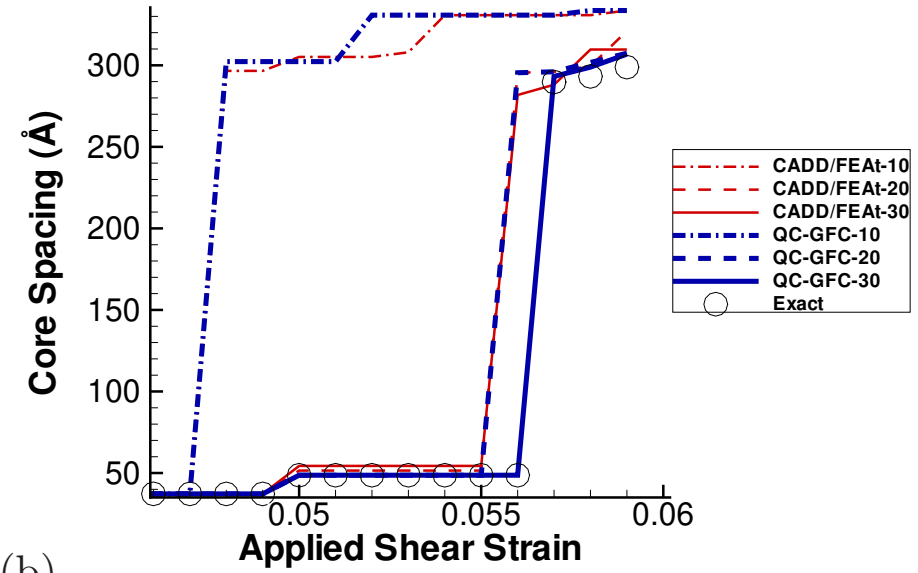


(b)

Figure 16: Comparison of the load vs. applied shear curves for the various multiscale models. (a) All methods, using the intermediate "Model-20" mesh. (b) The effect of model refinement for QC-GFC and CADD/FEAt. Note that at this scale it is very difficult to resolve the difference between CADD/FEAt-20 and QC-GFC-20, and also that the CQC(13)-F-20 result is outside the range of the graph.



(a)



(b)

Figure 17: Comparison of the core spacing vs. applied shear curves for the various multiscale models.

Method	Total time T, (sec)	Global overhead T_{OH}, (sec)	Timestep overhead t_{OH}, (sec)	Solve time T_s, (sec)	Force calls N_{force}	Time per force call $t_s = T_s/N_{\text{force}}$, (sec)
Exact	45520.11	0	0	45520.11	33225	1.42
CADD/FEAt	4629.61	128.06	1.96	4452.55	21153	0.21
HSM	8910.46	23.35	0.26	8880.87	37314	0.24
AtC	10423.34	23.44	0.12	10397.02	38375	0.27
AtC-GFC	11182	23.44	0.15	11154.96	40737	0.27
CQC(13)-F	23994	444.62	6.75	23387.38	21336	1.10

Table 9: Raw time results for timing study of the force-based methods. All calculations were performed using a single, 64-bit, 1.6 GHz Itanium processor on an 8-CPU Sun Altix 350 cluster running Linux. The main memory available to the system was 8 GB.

present the total time to simulate the shearing of the Lomer dipole, T, with table 8 showing the energy-based methods and table 9 showing the force-based methods. As discussed at the start of section 7, the exact results were obtained using both the energy-based and force-based solver. Incidentally, both of these solvers lead to essentially the same result, with the "displacement error" in the force-based solver result being on the order of 10^{-6} Å/atom (three orders of magnitude better than any error result reported for the multiscale methods). All calculations were performed using a single, 64-bit, 1.6 GHz Itanium processor on an 8-CPU Sun Altix 350 cluster running Linux. The main memory available to the system was 8 GB.

The simulation time is broken down into various parts as

$$T = T_{OH} + T_s + N_{\text{step}} t_{OH}, \tag{63}$$

where T_s is the time spent actually minimizing energy (or equilibrating forces) within each load step, $N_{\text{step}} = 24$ is the number of load steps in this example, and T_{OH} and t_{OH} are the time consumed in "overhead" associated with each method. The overhead is divided into "global" overhead, T_{OH}, that has to be carried out once at the start of the simulation and "timestep" overhead, t_{OH}, that must be repeated each timestep. The former, for example, may include building the model mesh and related data structures, while the latter may include such things as updating the ghost forces for the current configuration of the displacements. The "solve" time, T_s is the sum, over all 24 load steps, of the time spent inside the minimizer routine. We finally provide the time per force-call in the last column, which we define to be

$$t_s = T_s/N_{\text{force}}. \tag{64}$$

Note, however, that the T_s value includes both the actual calculation of forces and the updating of neighbor lists. Since neighbor lists are only updated occasionally using the Verlet list method [58], t_s averages these neighbor updates over all force calls. Our code is not written in an optimal way, especially in regards to the overhead specific to each method. While it is clear that some methods will have more overhead than others (for example, the searching and storing of clusters in CQC methods is generally a rather costly process), this

Method	F_{DOF}	F_{force}	F_{sim}
BSM	6.73	9.00	11.93
QC-GFC	7.13	8.73	10.96
CADD/FEAt†	7.13	6.73	10.22(5.87)
QC/CLS	7.13	8.82	9.92
CQC(1)-E	7.09	6.14	8.05
CQC(1)-GFC	7.09	6.38	6.18
HSM†	5.02	5.95	5.13 (3.05)
AtC†	5.04	5.23	4.38 (2.61)
AtC-GFC†	5.04	5.17	4.08 (2.43)
CQC(13)-E	7.09	2.27	3.09
BD	4.02	7.62	2.60
CQC(13)-F†	7.09	1.29	1.95 (1.13)
CACM	4.02	9.79	1.69

Table 10: Factors by which each method improves the number of degrees of freedom, time per force calculation and total simulation time. Numbers in parenthesis are the factors for force-based methods when compared with the faster exact solution (obtained with the energy-based CG method). †indicates a force-based method.

still comprises a relatively small part of the simulation that can, if one were diligent, be optimized. In fairness to the methods, we will compare only the true solve time, T_s.

To make the comparison easier, we use table 10 to present efficiency factors defined as

$$F_{\text{DOF}} = \frac{N_{\text{A}}}{n_{\text{nodes}}}, \qquad F_{\text{force}} = \frac{t_s^{\text{Exact}}}{t_s^{\text{Model}}}, \qquad F_{\text{sim}} = \frac{T^{\text{Exact}}}{T^{\text{Model}}}. \tag{65}$$

These are the factors by which each quantity, the number of degrees of freedom (DOF), the time per force call, and the simulation time, respectively, are improved by the multiscale models. They are defined such that bigger is better. The goal in multiscale modeling is to achieve large reductions in the number of DOF and computational time with minimal loss of accuracy. In table 10 we order the methods by decreasing values of the most important of these factors, namely F_{sim}. We identify, by a horizontal line, a high-efficiency and low-efficiency group. For the force-based methods, a second value of F_{sim} is indicated. This is the value that would have been obtained if we normalized the simulation time by the faster of the two exact results (computed using the energy-based CG method). While the first number is a fairer test of the efficacy of a given force-based method, the second is a reminder that the force-based solver we have used is a factor of about 1.7 times slower than the energy-based solver.

The factor by which the number of DOF are reduced in the various methods is not the same due to different constraints in the definition of the models. For example, the BD and HSM methods have handshake regions in which extra atoms are included relative to the other methods. For this particular model ("Model-20"), the values range from about 4 to about 7. The number is not really a test of the method's efficiency, but rather a normalizing influence on the timing results; if two methods have about the same value of F_{sim}, the one

with a smaller F_{DOF} has actually performed better by computing more degrees of freedom in the same amount of time.

Notwithstanding our earlier caveat about the overall (but equitable) inefficiency of the implementation, it is worth emphasizing the general significance of these numbers. Take the BSM case as an example. We have designed a test problem in which we have, somewhat arbitrarily, reduced the number of degrees of freedom by a factor of 6.73 compared to the fully atomistic problem. The resulting speed-up was a factor of 11.93. These numbers are an indication of the benefit of a multiscale method, but only for *this specific test problem*. We could easily have inflated the speed-up 100-fold by making the continuum region very large and coarsening the mesh. To determine the benefit of a multiscale method over direct atomistics, we need to be able to establish the minimum model size required to answer a specific problem, or alternatively to quantify the errors and limitations associated with a smaller model. These things depend on the degree to which behaviour beyond the applicability of a continuum model is spatially localized, and on the importance of long-ranged stresses and interactions. Such questions are not within the scope of this current paper, where we are only trying to assess the *relative* efficiency of the various approaches.

In the simulation times, we start to see some patterns emerging. The fastest methods are energy-based, although the CADD/FEAt method is a fast force-based alternative relative to other force-based methods. Of the four fast methods, three employ strong compatibility and no handshake region; only the BSM employs a handshake region and some elements of weak compatibility.

While the CQC(1)-E method is quite fast, recall that we have already concluded that the accuracy of this approach (which uses 1-atom clusters) is unacceptably low. The CQC(m)-E methods with reasonable accuracy are considerably slower. Expanding to clusters of 13 atoms nearly triples the simulation time compared to CQC(1)-E, and makes CQC(13)-E about 4 times slower than the fastest method (BSM). The ghost force corrected CQC(1)-GFC fares somewhat better. In this case, there is a large global overhead which is about double that for CQC(13)-E since we must effectively build and store data for two cluster sizes. But the solve time for CQC(1)-GFC is approaching the values for the faster methods, being only about two times slower. In this case, the slowness can in part be attributed to requiring more force calls to converge, suggesting that the ghost force corrections in this case make the energy landscape somewhat more difficult to navigate for the solver.

The force-based CADD/FEAt method, which we recall was the most accurate of the methods, is also one of the fastest if we compare it to the solution time for the full atomistic problem using the force-based solver, CG-FB (Recall that this solver is about 1.7 times slower than the energy-based CG solver). For this particular example, it seems that the simple modified CG-FB solver is reasonably good for the force-based approach, but does require about 1.2 times as many force calls to find convergence as the comparable energy-based approach (QC/CLS). It may be possible to improve the absolute convergence rate of the CG-FB by using a solver that is better tuned to the force-based methodology. At the same time, the force-based methods require longer time per force call because one must effectively visit more atoms to find the forces. In this particular example this works out to about 15% more atoms. This means that a single force call will be about 15% slower, while a neighbor list update (which in our code is $O(N^2)$) will be about 32% slower. Taken together, these explain the longer time per force call required by the CADD/FEAt method.

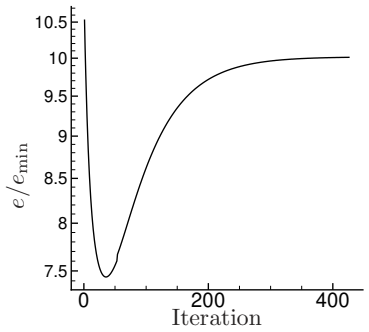


Figure 18: Slow convergence of the iterative energy-based CACM solver. Displacement error values are normalized by the error for the CADD/FEAt-30 model.

This seems to be generally true with force- versus energy-based models. For example, BD and AtC are closely related but for this difference in their formulations, and we see a similar pattern in their values of F_{force}.

The long simulation times for BD and HSM are related to the apparently slow convergence of methods that do not impose strong compatibility. This is evident from the large number of force calls required (which is more or less proportional to the number of iterations required for convergence). This suggests that the use of weak compatibility leads to a much more complicated energy landscape, making it difficult to find minima with the CG and CG-FB approaches. It is possible that an appropriate pre-conditioner for the CG method may improve this performance.

The poor performance gain of CACM is due to the slow convergence of the iterative solver method, which we elaborate on next.

7.5.1 Iterative solvers: The slow convergence of the CACM model

The CACM model is unique in our study in that it attempts to iteratively minimize two energy functionals, alternately imposing boundary conditions from one side of the atomistic/continuum coupling while minimizing the energy of the other. In the results presented here, we found that this strategy led to relatively poor accuracy and slow convergence.[20]

The CACM results shown here were obtained by setting the following convergence criterion. After a minimization of the atomistic energy (while holding the continuum degrees of freedom fixed), if the change in total energy was less that 0.1% since the previous atomistic minimization, the results were deemed to have converged. For a load step in which the dislocations did not move, this typically required between 2-4 iterations to convergence. For load steps in which the dislocation moved, on the other hand, as many as 84 iterations were required.

At the same time, it is clear from the error levels reported for CACM in table 3 that

[20]We recall that that the BSM, as presented in [39], requires an iterative solver due to the decoupling of the atomistic displacements into a coarse and fine scale. Here, we have simplified this approach and abandoned this decoupling. The effect has been a method that is quite fast relative to the others, but it seems likely that the iterative variant of this approach would suffer the same inefficiencies as CACM.

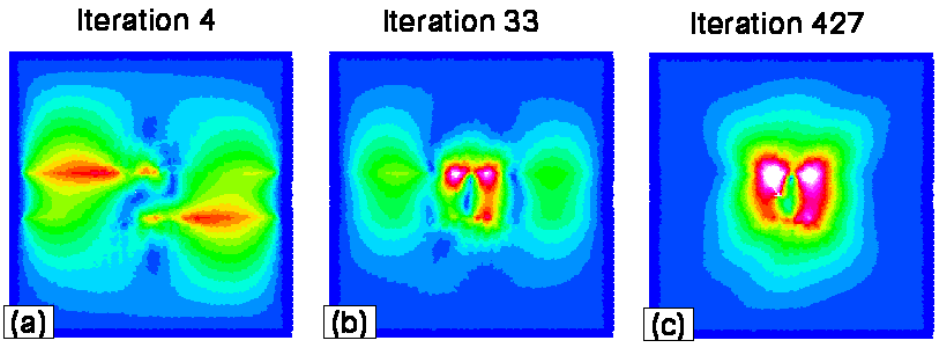


Figure 19: Displacement error contours at selected iterations during the slow convergence of the iterative energy-based CACM solver.

this convergence criterion may not be strict enough; CACM accuracy is relatively poor. To check whether the CACM result is truly "converged", we re-ran CACM-30 on the initial relaxation of the dipole cores (no applied shear) without a global convergence criterion, and monitored the error as the iterations proceeded. Fig. 18 shows these results against the benchmark displacement error of the CADD/FEAt-30 model. The convergence is clearly quite slow, and exhibits surprising behaviour, dipping to a minimum before *rising* with further iterations before finally converging to a normalized value of about 10. In fig. 19, we show the distribution of this error as a function of the number of iterations. In (a), we show the result presented initially, which was deemed "converged" by our criterion of a 0.1% change in the energy. In (b), we see the error distribution at the minimum in fig. 18, while (c) shows the final converged state. The progression explains the strange convergence behaviour; the iterations have the effect of removing ghost force effects along the interface while simultaneously exacerbating error in the vicinity of the dislocation core. It seems that this error around the core is related to the averaging scheme in CACM for determining the displacements of nodes inside the atomistic region, which introduces a systematic error in the continuum solution.

While iterative solvers can be very straightforward to implement and can circumvent the direct treatment of the ghost force problem, it is clear that their convergence behaviour suffers considerably.

7.6 The Effect of the Continuum Constitutive Law

As a test of the effect of the finite element constitutive law, we have recomputed one of the results, CADD/FEAt-20, with the Cauchy-Born rule replaced by anisotropic linear elasticity.[21] The atomistic region remains the same in both cases. The displacement error distributions presented in fig. 20 look similar, however the overall error of the linear elastic model is $e/e_{\min} = 1.68$ (compared to 1.00 for the Cauchy-Born model). Although the percent-error remains small in absolute terms, the loss in accuracy is still significant given the localized na-

[21] We fit the elastic moduli to Ercolessi-Adams fcc aluminum [16], $C_{11} = 0.7371$ eV/Å^3, $C_{12} = 0.3888$ eV/Å^3, $C_{44} = 0.2291$ eV/Å^3.

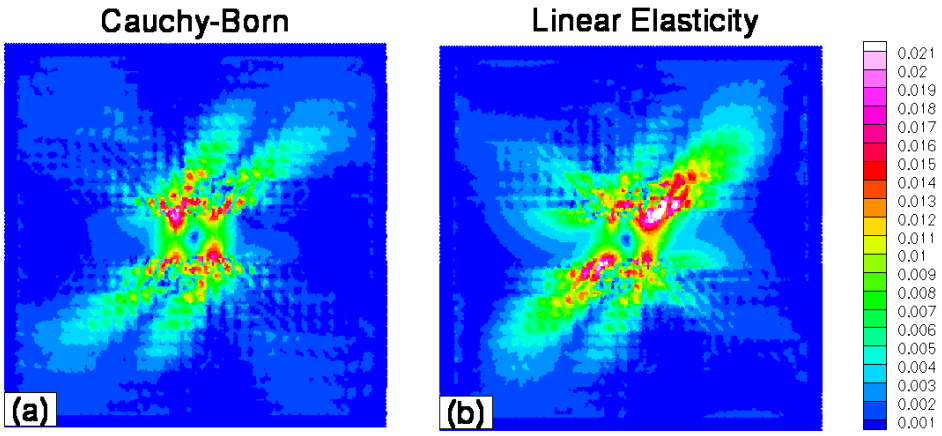


Figure 20: Comparing the CADD/FEAt-20 results using (a) Cauchy-Born and (b) linear elasticity as the continuum constitutive methods. Note that the contour levels are not the same as in previous displacement error plots.

ture of nonlinear deformation in this problem. We expect the use of linear elasticity to have even more serious consequences in problems such as nano-indentation and friction studies, where high compressive stresses can exist in the continuum region and render the material behaviour highly nonlinear. This is discussed, for example, in [47] and [28].

In terms of efficiency, we show the comparison in table 11. Interestingly, we find that while the efficiency of a single force call is improved using linear elasticity, the total simulation efficiency is not. This is due to a larger number of force calls required to find equilibrium (30534 for linear elasticity compared to 21153 for the Cauchy-Born version).

Further analysis of this example gives us an idea of the efficiency of multiscale methods in general. On our system, the average time to evaluate the force contributions from a single Cauchy-Born finite element was $16.1\mu sec$, while the same calculation with linear elasticity took $3.88\mu sec$. Both of these are, of course, independent of element size, and so the savings over direct atomistics will increase as elements are made larger and serve to replace more atoms (for comparison, the atomistic force evaluations took about $24.9\mu sec$ per atom). The Cauchy-Born CADD/FEAt-20 simulation spent about 71% of its force evaluation time on the atoms, and the remaining 29% on the continuum region. In the linear elastic version, this balance shifted to 91% on the atoms and only 9% on the continuum. Of course, the fraction of time spent on the atoms represents the same absolute time in both cases; it is only the efficiency of the continuum region that changes when we change the continuum constitutive law.

8 Summary and Conclusions

Looking back at the efficiency analysis, we observe that generally (but not exclusively) energy-based methods are faster than force-based, strong compatibility is faster than weak,

Method	F_{DOF}	F_{force}	F_{sim}
Cauchy-Born	7.13	6.73	10.22(5.87)
Elastic	7.13	8.66	9.12

Table 11: Comparing the speed of the CADD/FEAt model with a Cauchy-Born and linear elastic constitutive law.

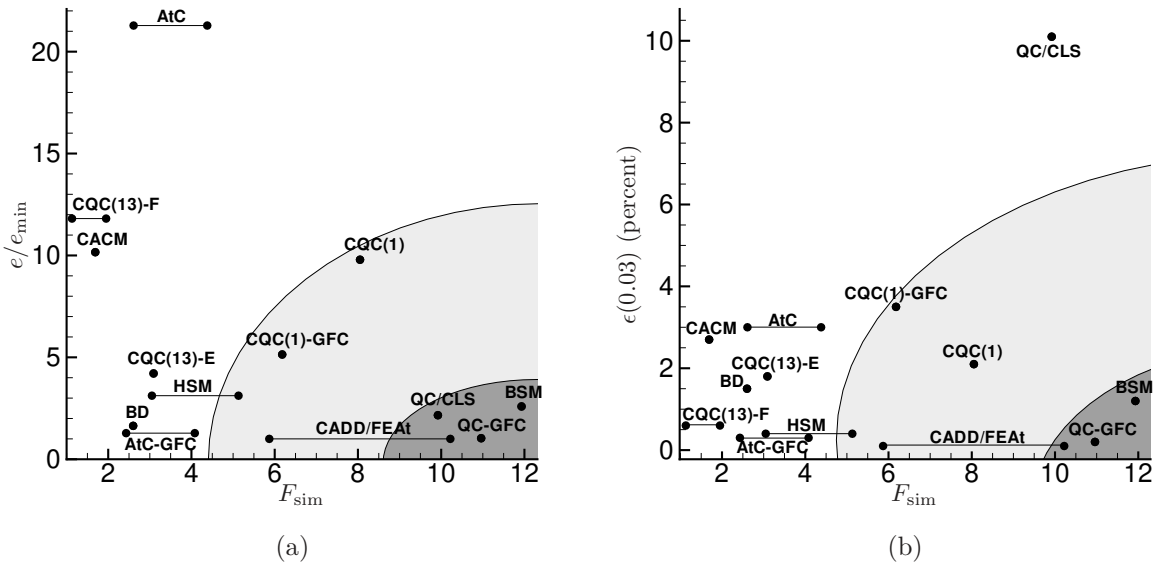


Figure 21: Summary of accuracy and efficiency for all of the methods.

and the absence of a handshake region is faster than the presence of one. The principle exception to this last point is our static, non-iterative, implementation of the BSM, which is fast despite (or perhaps because of) its unique handshake region.

In the displacement error results, it seems that we can make the general observation that force-based methods are more accurate than energy-based, although a simple ghost-force correction can in most cases eliminate this advantage. Further, strong compatibility seems to be more accurate than weak and the presence of a handshake region reduces the accuracy. Interestingly, the ghost force correction introduced to improve accuracy in QC-GFC actually made the QC method faster. This is presumably because the small additional effort to compute the ghost force corrections is offset by a substantial reduction in the number of force calls required to converge the solution on what is presumably a smoother energy landscape.

Considering the energy error measures, there is cause for considerable optimism for coupled methods in general, as the absolute energy error for most of the methods was generally less than 5% for this problem. Although it seems that even this slight error can lead to profound effects on the resulting dislocation motion that is predicted, it seems that none of the methods are especially inaccurate in their estimate of the energy.

The accuracy and efficiency results are combined in two simple graphs in fig. 21, where each method is represented by a point in "error-speed" space. In fig. 21(a), the accuracy is represented by the displacement error for the "model-30" results (table 3), while fig. 21(b)

shows the accuracy in terms of the total energy error, $\epsilon(0.03)$ for "model-30" (table 6). All of the force-based methods are indicated by two points joined by a thin line because the speed-up depends on how we normalize the results (due to the use of two different solvers). The ideal model would be in the lower right corner, with high efficiency and high accuracy (low error). These graphs allow us to identify, somewhat subjectively, the best performing models in the dark grey region, intermediate performance in light grey and poor performance in white.

The force-based methods are slower, to a large extent, because of the rather unsophisticated CG-FB solver we have used. Normalizing for the differences between the CG and CG-FB solver, the force-based CADD/FEAt method is essentially as efficient as the comparable energy-based method (QC-GFC) in improving the speed of the calculation. However, to see this efficiency in absolute terms, it is necessary to develop a force-based solver that is as fast, or faster, than a comparable energy-based one (in our current study CG-FB is about 1.7 times slower than CG). Finding the optimal solver for the force-based methods is certainly an area worthy of further study.

Intuitively, it might seem reasonable to expect that using a handshake region would improve accuracy, by providing a more gradual transition from the atomistic to the continuum description. In fact, this seems not to be the case, as handshake methods tended to be both slower and less accurate. This is also true of methods using weak compatibility. Handshake regions and weak compatibility are appealing features because they make mesh generation easier; there is no need of mesh refinement down to the atomic scale or the tricky mesh generation routines that this may require. But this study suggests that this comes at a rather high price in terms of ultimate performance.

Cluster-based methods are generally slower and less accurate than methods which appeal to a traditional continuum constitutive law such as linear elasticity or the Cauchy-Born rule. This finding is consistent with, and analytically explained by, a recent mathematical study [29] comparing cluster-based and element-based models. Our results also indicate that errors in the continuum region significantly contribute to the overall error of these methods.

The CACM model was implemented here as a representative of the class of coupled methods that use an iterative solver. In these approaches, one part (either the atomistic or continuum region) of the problem is held fixed to provide the boundary conditions during energy minimization of the other part, and then the roles are switched back and forth until the change in energy during an iteration is small. Our results show that such methods can be extremely slow, to the point of potentially eliminating any benefit of the coupled method over the full atomistic solution if reasonable accuracy is expected.

Multiscale methods like the ones discussed in this review show much promise to improve the efficiency of atomistic calculations, but they have not yet fully realized this potential. This is in part because the focus to date has mainly been on development of the methodology as opposed to the large-scale application to materials problems. One of the disadvantages of multiscale methods is that, relative to straightforward atomsistics, they tend to be more difficult to implement. In order for multiscale methods to compete with, or eventually replace, atomistics it is necessary that they methods be implemented in 3D, parallel codes optimized to the same degree as atomistic packages like, for example, LAMMPS [37, 36]. It may even be desirable to introduce multiscale modeling as a feature within existing atomistic packages. One of the barriers to this is the wide variety of existing multiscale approaches from

which to choose; it is not clear where to invest the time required to create a highly optimized code. This work has taken a step towards removing this barrier by demonstrating that, to a large extent, there is a great deal of similarity between the methods and that they can be described within a common framework. We have also taken a first step towards identifying certain "best practice" strategies for getting the most efficient and accurate implementation possible.

Once an optimized multiscale implementation is developed, its potential for significantly improving the efficiency of atomistic calculations is enormous. Here, we have demonstrated 12-fold improvements in efficiency for an extremely modest test problem, using a very poorly written code and comparing to an atomistic model (EAM) that is amongst the least expensive available. For materials problems where deformation tends to localize and long-range effects are important, an optimized code could easily yield 100 to 1000-fold gains in efficiency. This efficiency would be further increased in we considered more expensive interatomic models like the Brenner-Tersoff potentials [56, 7], MGPT potentials [32, 33], or ReaxFF model [57].

Acknowledgments

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (discovery grant program) and by the Department of Energy under Award Number DE-FG02-05ER25706. We are grateful to several colleagues for their help with this work. Specifically, we would like to thank Dong Qian, Michael Parks, Catalin Picu and Leo Shilkrot for their help in elucidating the details of several of the implementations, and Mitch Luskin and Bill Curtin for helpful discussions about the work.

References

[1] F. F. Abraham, J.Q. Broughton, N. Bernstein, and E. Kaxiras. Spanning the length scales in dynamic simulation. *Comput. Phys.*, 12:538, 1998.

[2] F.F. Abraham, N. Bernstein, J.Q. Broughton, and D. Hess. Dynamic fracture of silicon: Concurrent simulation of quantum electrons, classical atoms, and the continuum solid. *Mat. Res. Soc. Bulletin*, 25(5):27–32, 2000.

[3] M. Arndt, V. Sorkin, and E. B. Tadmor. Efficient algorithms for discrete lattice calculations. *J. Comp. Phys.*, in press, 2009.

[4] S. Badia, P. Bochev, R. Lehoucq, M. L. Parks, J. Fish, M. Nuggehally, and M. Gunzburger. A force-based blending model for atomistic-to-continuum coupling. *International Journal for Multiscale Computational Engineering*, 5(5):387–406, 2007.

[5] Santiago Badia, Michael Parks, Pavel Bochev, Max Gunzburger, and Richard Lehoucq. On atomistic-to-continuum coupling by blending. *Multiscale Modeling & Simulation*, 7(1):381–406, 2008.

[6] Paul T. Bauman, Hachmi Ben Dhia, Nadia Elkhodja, J. Tinsley Oden, and Serge Prudhomme. On the application of the Arlequin method to the coupling of particle and continuum models. *Comp. Mech.*, 42(4):511–530, SEP 2008.

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

[8] Jeremy Q. Broughton, Farid F. Abraham, Noam Bernstein, and Efthimios Kaxiras. Concurrent coupling of length scales: Methodology and application. *Phys. Rev. B*, 60(4):2391–2403, 1999.

[9] W.A. Curtin and R.E. Miller. Atomistic/continuum coupling methods in multi-scale materials modeling. *Modeling Simul. Mater. Sci. Eng.*, 11(3):R33–R68, 2003.

[10] D.K. Datta, R. Catalin Picu, and Mark S. Shephard. Composite grid atomistic continuum method: An adaptive approach to bridge continuum with atomistic analysis. *Intl. J. Multiscale Computational Engineering*, 2(3):71–90, 2004.

[11] M. Dobson and M. Luskin. Analysis of a force-based quasicontinuum approximation. *Math. Model. Numer. Anal.*, 2007. submitted.

[12] M. Dobson and M. Luskin. Analysis of the effect of ghost force oscillation on the quasicontinuum error. *Math. Model. Numer. Anal.*, 2008. submitted.

[13] Weinan E, Bjorn Engquist, Xiantao Li, Weiqing Ren, and Eric Vanden-Eijnden. Heterogeneous multiscale methods: A review. *Comm. Comput. Phys.*, 2(3):367–450, 2007.

[14] Weinan E, Jianfeng Lu, and J.Z. Yang. Uniform accuracy of the quasicontinuum method. *Phys. Rev. B*, 74(21):214115/1–12, 2006.

[15] B. Eidel and A. Stukowski. A variational formulation of the quasicontinuum method based on energy sampling of clusters. *J. Mech. Phys. Sol.*, 2008.

[16] F. Ercolessi and J.B. Adams. Interatomic potentials from first-principles calculations – the force-matching method. *Europhys. Lett.*, 26:583, 1994.

[17] J.L. Ericksen. *Phase Transformations and Material Instabilities in Solids*, pages 61–77. Academic Press, New York, 1984. Edited by M. Gurtin.

[18] M. Fago, R.L. Hayes, E.A. Carter, and M. Ortiz. Density-functional-theory-based local quasicontinuum method: Prediction of dislocation nucleation. *Phys. Rev. B*, 70(10):100102, 2004.

[19] Jacob Fish, Mohan A. Nuggehally, Mark S. Shephard, Catalin R. Picu, Santiago Badia, Michael L. Parks, and Max Gunzburger. Concurrent AtC coupling based on a blend of the continuum stress and the atomistic force. *Computer Methods in Applied Mechanics and Engineering*, 196(45-48):4548–4560, SEP 15 2007.

[20] S.M. Foiles, D.L. Olmsted, and E.A. Holm. Using atomistic simulations to inform mesoscale simulations of microstructural evolution. In Anter El-Azab, editor, *Proc. 4th International Conference of Multiscale Materials Modeling (MMM)*, pages 362–368, 2008.

[21] J. P. Hirth and J. Lothe. *Theory of Dislocations*. Krieger, Malabar, Florida, 1992.

[22] Cynthia L. Kelchner, S.J. Plimpton, and J.C. Hamilton. Dislocation nucleation and defect structure during surface indentation. *Phys. Rev. B*, 58(17):11085–11088, 1998.

[23] P.A. Klein and J.A. Zimmerman. Coupled atomistic-continuum simulations using arbitrary overlapping domains. *Journal of Computational Physics*, 213(1):86–116, 2006.

[24] J. Knap and M. Ortiz. An analysis of the quasicontinuum method. *J. Mech. Phys. Sol.*, 49:1899–1923, 2001.

[25] S. Kohlhoff, P. Gumbsch, and H. F. Fischmeister. Crack propagation in bcc crystals studied with a combined finite-element and atomistic model. *Phil. Mag. A*, 64(4):851–878, 1991.

[26] W.K. Liu, E.G. Karpov, S. Zhang, and H.S. Park. An introduction to computational nano mechanics and materials. *Comput. Meth. Appl. Mech. Eng.*, 193:1529–1578, 2004.

[27] G. Lu, E. B. Tadmor, and E. Kaxiras. From electrons to finite elements: A concurrent multiscale approach for metals. *Phys. Rev. B*, 73(2):024108, January 2006.

[28] B. Q. Luan, S. Hyun, J. F. Molinari, N. Bernstein, and Mark O. Robbins. Multiscale modeling of two-dimensional contacts. *Phys. Rev. E*, 74:046710–1–046710–11, 2007.

[29] M. Luskin and C. Ortner. An analysis of node-based cluster summation rules in the quasticontinuum method. *SIAM J. Numer. Anal.*, 2009. submitted.

[30] R. E. Miller and E. B. Tadmor. Multiscale benchmark code, `multibench`, 2009. Available at www.qcmethod.com.

[31] Ronald E. Miller and E.B. Tadmor. Hybrid continuum mechanics and atomistic methods for simulating materials deformation and failure. *MRS Bulletin*, 32:920–926, 2007.

[32] J. A. Moriarty. Density-functional formulation of the generalized pseudopotential theory: Transition-metal interatomic potentials. *Phys. Rev. B*, 38(5):3199–3231, 1988.

[33] J. A. Moriarty. Angular forces and melting in bcc transition metals: A case study of molybdenum. *Phys. Rev. B*, 49(18):12431–12445, 1994.

[34] Harold S. Park and W.K. Liu. An introduction and tutorial on multiple-scale analysis in solids. *Comput. Methods Appl. Mech. Engrg*, 193:1733–1772, 2004.

[35] Michael L. Parks, Pavel B. Bochev, and Richard B. Lehoucq. Connecting atomistic-to-continuum coupling and domain decomposition. *Multiscale Modeling & Simulation*, 7(1):362–380, 2008.

[36] S. J. Plimpton. Fast parallel algorithms for short-range molecular dynamics. *J Comp Phys*, 117:1–19, 1995.

[37] S.J. Plimpton. Lammps website, 2009. lammps.sandia.gov.

[38] Serge Prudhomme, Paul T. Bauman, and J. Tinsley Oden. Error control for molecular statics problems. *Int. J. Mult. Comp. Eng.*, 4(5-6):647–662, 2006.

[39] D. Qian, G.J. Wagner, and Wing Kam Liu. A multiscale projection method for the analysis of carbon nanotubes. *Computer Methods in Applied Mechanics and Engineering*, 193:1603–32, 2004.

[40] J. D. Rittner and D. N. Seidman. $< 110 >$ symmetric tilt grain-boundary structures in fcc metals with low stacking-fault energies. *Phys. Rev. B*, 54(10):6999–7015, 1996.

[41] R.E. Rudd and J.Q. Broughton. Atomistic simulation of mems resonators through the coupling of length scales. *J. Model. Sim. Microsys.*, 1(1):29–38, 1999.

[42] Robert E. Rudd. Concurrent multiscale modeling of embedded nanomechanics. *Mat. Res. Soc. Symp. Proc.*, 677:AA1.6.1–AA1.6.12, 2001.

[43] Robert E. Rudd and Jeremy Q. Broughton. Coarse-grained molecular dynamics: Nonlinear finite elements and finite temperature. *Phys. Rev. B*, 72(14):144104, Oct 2005.

[44] Robert E. Rudd and J.Q. Broughton. Concurrent coupling of length scales in solid state systems. *Phys. Stat Solidi B*, 217:251–291, 2000.

[45] V. B. Shenoy, R. Miller, E. B. Tadmor, R. Phillips, and M. Ortiz. Quasicontinuum models of interfacial structure and deformation. *Phys. Rev. Lett.*, 80(4):742–745, 1998.

[46] V. B. Shenoy, R. Miller, E.B. Tadmor, D. Rodney, R. Phillips, and M. Ortiz. An adaptive methodology for atomic scale mechanics: The quasicontinuum method. *J. Mech. Phys. Sol.*, 47:611–642, 1999.

[47] L. E. Shilkrot, R. E. Miller, and W.A. Curtin. Coupled atomistic and discrete dislocation plasticity. *Phys. Rev. Lett.*, 89(2):025501, 2002.

[48] L. E. Shilkrot, R. E. Miller, and W.A. Curtin. Multiscale plasticity modeling: Coupled atomistic and discrete dislocation mechanics. *J. Mech. Phys. Sol.*, 52(4):755–787, 2004.

[49] T. Shimokawa, J. J. Mortensen, J. Schiøtz, and K. W. Jacobsen. Matching conditions in the quasicontinuum method: Removal of the error introduced at the interface between the coarse-grained and fully atomistic region. *Phys. Rev. B*, 69:214104/1–10, 2004.

[50] E. B. Tadmor. *The Quasicontinuum Method*. PhD thesis, Brown University, 1996.

[51] E. B. Tadmor and R. E. Miller. Quasicontinuum method website, 2009. www.qcmethod.com.

[52] E. B. Tadmor and R. E. Miller. *Quasicontinuum Tutorial Guide version 1.3*, May 2007. www.qcmethod.com.

[53] E. B. Tadmor, M. Ortiz, and R. Phillips. Quasicontinuum analysis of defects in solids. *Phil. Mag. A*, 73(6):1529–1563, 1996.

[54] E.B. Tadmor and R.E. Miller. The theory and implementation of the quasicontinuum method. In *Handbook of Materials Modeling, Volume I (Methods and Models)*, chapter 2.13. Springer Science and Business Media, New York, 2006.

[55] E.B. Tadmor, G.S. Smith, N. Bernstein, and E. Kaxiras. Mixed finite element and atomistic formulation for complex crystals. *Phys. Rev. B*, 59(1):235–245, 1999.

[56] J. Tersoff. Empirical interatomic potential for silicon with improved elastic properties. *Phys. Rev. B*, 38(14):9902–9905, 1988.

[57] Adri C.T. van Duin, Siddharth Dasgupta, Francois Lorant, and William A. Goddard III. Reaxff: A reactive force field for hydrocarbons. *J. Phys. Chem. A*, 105:9396–9409, 2001.

[58] Loup Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, 159(1):98, Jul 1967.

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

[60] S.P. Xiao and T. Belytschko. A bridging domain method for coupling continua with molecular dynamics. *Computer Methods in Applied Mechanics and Engineering*, 193:1645–69, 2004.

[61] O. C. Zienkiewicz. *The Finite Element Method*, volume 1-2. McGraw-Hill, London, 4th edition, 1991.