A Computational Library for Multiscale Modeling of Material Failure

Hossein Talebi[1], Mohammad Silani[2], Stéphane P. A. Bordas[3], Pierre Kerfriden[3], Timon Rabczuk[1,4]

[1] Institute of Structural Mechanics, Bauhaus-Universitaet Weimar, Marienstr. 15, D-99423 Weimar, Germany

[2] Department of Mechanical Engineering, Isfahan University of Technology, Isfahan, Iran 84156-83111

3 Institute of Mechanics and Advanced Materials, Theoretical and Computational Mechanics, Cardiff University, Cardiff, UK, CF24 3AA

4 Corresponding author: School of Civil, Environmental and Architectural Engineering, Korea University, South Korea

Abstract

We present an open-source software framework called PERMIX for multiscale modeling and simulation of fracture in solids. The framework is an object oriented open-source effort written primarily in Fortran 2003 standard with Fortran/C++ interfaces to a number of other libraries such as LAMMPS, ABAQUS, LS-DYNA and GMSH. Fracture on the continuum level is modeled by the extended finite element method. Using several novel or state of the art methods, the piece software handles semi-concurrent multiscale methods as well as concurrent multiscale methods for fracture, coupling two continuum domains or atomistic domains to continuum domains, respectively. The efficiency of our open-source software is shown through several simulations including a 3D crack modeling in clay nanocomposites, a semi-concurrent FE-FE coupling, a 3D Arlequin multiscale example and an MD-XFEM coupling for dynamic crack propagation.

Keywords: Crack, Fracture, Molecular Dynamics, XFEM, PERMIX

1 Introduction

Multiscale modeling of fracture or modeling fracture in general is today still a significant challenge. Multiscale methods are powerful for extracting material properties based on the fine-scale details. While numerous multiscale methods (see e.g. [57, 38]) were developed for intact materials, far fewer methods are available for fracture simulations. It is believed that Molecular Dynamic (MD) simulations of fracture will help gaining a fundamental understanding about the failure of solids. However, MD simulations are not feasible for specimens of reasonable size. Therefore, multiscale methods have been developed that couple MD-models to continuum models.

Multiscale methods can be classified into hierarchical, semi-concurrent and concurrent methods, Fig. 1. In hierarchical multiscale methods, information are passed from the fine-scale to the coarse-scale only. Numerical homogenization is a classical upscaling technique. Here a macroscopic constitutive model is assumed and the parameters of the model are computed from the microstructure. Among many other we refer to the studies by Nakamura and Pettermann and Suresh [100], Suresh [87], Christman et al. [29], van der Sluis et al. [133], and extensions to composite materials were conducted in [32, 60, 1, 70, 106].

Though attractive because of their computational efficiency, the extension of hierarchical multiscale methods to model fracture is complex. In the presence of a crack in the micro-scale, one basic assumption for the application of homogenization theories is the existence of disparate length scales [88]: $\ell_{Cr} \ll \ell_{RVE} \ll \ell_{Spec}$ where ℓ_{Cr}, ℓ_{RVE} and ℓ_{Spec} are the crack length, the RVE (Representative Volume Element) and specimen-size, respectively. For problems involving fracture, the first condition is violated as ℓ_{Cr} is in the order of ℓ_{RVE}. Moreover, periodic boundary conditions (PBC) often used at the fine-scale to improve convergence, cannot be used when there exists an edge crack touching a boundary as the displacement jump on that boundary violates the PBC.

In semi-concurrent multiscale methods, there is an interaction between the fine-scale and the coarse-scale during the simulation, Fig. 1. A classical semi-concurrent multiscale method is the FE^2 [37] developed originally for intact materials. In concurrent methods, there is a direct coupling of fine and coarse-scales. Numerous concurrent multiscale methods for intact materials were proposed in the literature. There are two categories that can be classified into handshake coupling and interface coupling methods [84].

In general, a multiscale model can involve continuum, atomistic or quantum mechanical descriptions of the matter at each scale. Also, the discretization strategy at each scale can be different i.e. one can use for example finite elements or meshfree methods to model the continuum behavior. As a result, due to the complexity of the implementation of multiscale methods, the class of problems approached by other researchers are mostly limited to small, two dimensional and academic cases. The reason as noted by Shepard et al. [115] is that so far

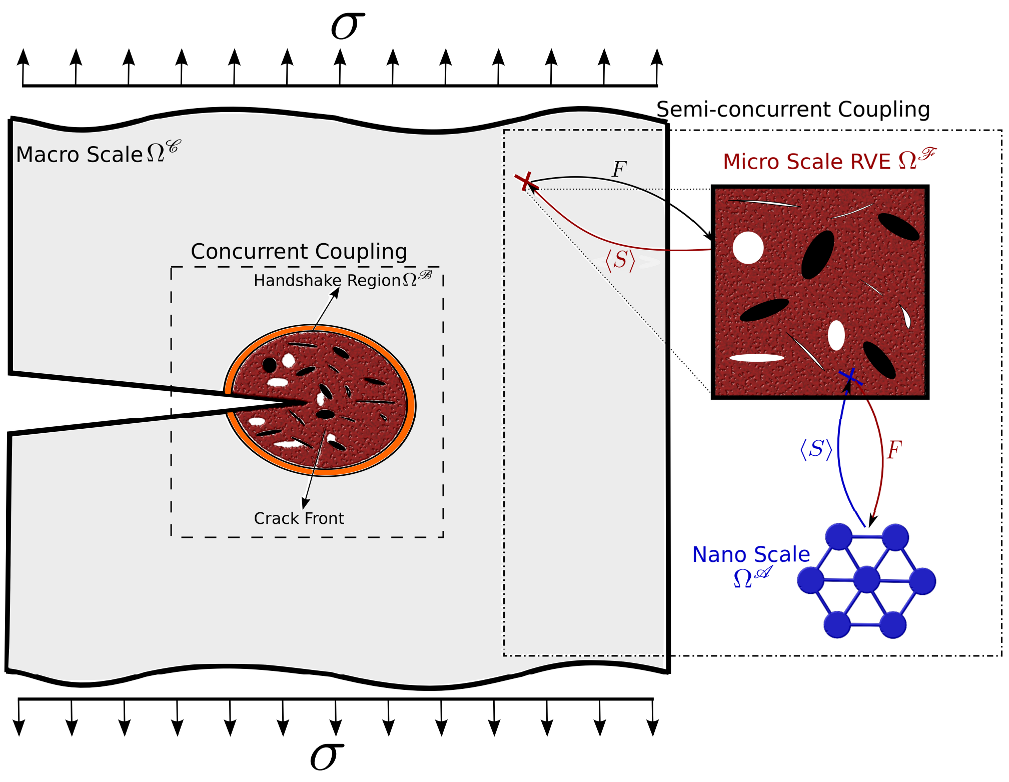
σ
Semi-concurrent Coupling
Macro Scale $\Omega^{\mathscr{C}}$
Micro Scale RVE $\Omega^{\mathscr{F}}$
F
$\langle S \rangle$
Concurrent Coupling
Handshake Region $\Omega^{\mathscr{B}}$
Crack Front
$\langle S \rangle$
F
Nano Scale $\Omega^{\mathscr{A}}$
σ

Figure 1: Schematic of a concurrent and semi-concurrent multiscale methods.

there is no unified tool that enables one to easily connect different abstract components with multiscale methods. Notable multiscale modeling tools are the one offered by Miller et al. [84] where the open source implementation offers 14 different multiscale methods bridging atomistic to continuum methods. Digimat is a commercial tool which connects several commercial finite element packages. However, those open-source components focus on multiscale methods for intact materials. To our knowledge, no piece of software enabling multiscale fracture simulation in an integrated framework is available.

Many computational methods exist to model propagating cracks in a single scale. Continuum mechanics approaches that smear the crack width over several elements are computationally expensive as the length scale of the specimen is magnitudes larger than the width of the cracks. Examples of computational methods for discrete fracture are interelement-separation methods [138], Extended Finite Element Method (XFEM)/Generalized Finite Element Method (GFEM) [17, 85, 125], Extended Element Free Galerkin Method (XEFG) [140, 26, 130] or Cracking Particles [105]. In XFEM for example, the crack can be captured within a single element. Fracture in Molecular Dynamics (MD) is modeled by breaking bonds between neighboring atoms. In the Peridynamics method also, fracture is modeled with breaking bonds between the particles [121].

In order to implement a framework for multiscale analysis, the software design must account for many different model situations. The framework should be able to handle several simulation components with and without different modeling concepts. Also, the framework should be able to communicate with existing tools such as commercial or open-source software to model a certain scale of the model. The communication with other tools demand a proper interface that efficiently transfers data. This can be achieved, amongst other means, by a full object oriented design which connects several tools and libraries together and with its own capabilities.

Most of the previous work in object-oriented design of FE software was performed using C++ and Java. However, Fortran is the traditional language used by many advanced commercial FE software. The use of Object Oriented Programming (OOP) concepts in Fortran 90 for finite elements was shown in [4, 95, 34]. A successful use of the Fortran 2003 standard was reported by Nie et al. [94] where modern features of Fortran 2003 were used to implement an adaptive multi-physics library. Fortran 2003 as a modern programming language and backward compatibility with Fortran 9x/77 convinced the authors of this work who used it as their primary language to implement the computational library. The attractive features of Fortran 2003 such as derived data types, type extension and inheritance, dynamic memory allocation, polymorphism, enhanced interoperability with the C language and finally its general ease of implementation and debugging compared to C++ [82] will help the maintenance and extension of the tool.

In this manuscript we offer a computational software (named PERMIX) that has several advantages which makes it a versatile and efficient toolbox for multiscale simulation of material failure. Several scales can coexist in 2D and 3D problems. Hierarchical, semi-concurrent and concurrent multiscale methods are available for both continuum and atomistic domains. For continuum based methods, it supports large deformation in addition to different nonlinear material models. For atomistic domains, the code has its own parallel atomistic library with some basic atomistic potentials useful for metals. Moreover, PERMIX has a full interface to the LAMMPS Library that includes Peridynamics [120, 121, 98] and the SPH method [56, 72] in addition to its comprehensive molecular dynamics library. Therefore, nearly all capabilities of

LAMMPS can be utilized in a multiscale simulation. The code currently supports several novel and state of the art multiscale methods to model dynamic fracture. Finally, further extension of the multiscale methods, finite elements, material models etc. are easily possible thanks to the novel design of the code.

This paper is organized as follows. We first describe some important theory and the notable research background on the semi-concurrent and concurrent multiscale methods used in our open-source library and as well as XFEM, that we use to model fracture in the continuum domain. We also summarize the atomistic models available in our open-source software. In section 3, we introduce our software framework PERMIX and its design concepts. We describe the interface generation strategy in the section 4. Section 5 is devoted to show actual strategies used in facing multiscale problems involving fracture in terms of implementation. In section 6 we will show several examples which cover the main capabilities of the current implementation of PERMIX to model multiscale cracks. Section 7 will explain the future trends of the project and concludes our manuscript.

2 Theory and Background

2.1 Multiscale Methods

In this section we briefly cover the multiscale methods which are used in our software framework. A continuum mechanics model is used at the coarse-scale whilst an atomistic model or another continuum model containing fine-scale features is employed at the fine-scale. Consider the domain $\Omega = \Omega^{\mathscr{F}} \bigcup \Omega^{\mathscr{C}}$, the superscripts $\mathscr{F}$ and $\mathscr{C}$ denoting the fine-scale and coarse-scale, respectively. The outer boundary of $\Omega^{\mathscr{C}}$ is denoted by $\Gamma^{\mathscr{C}}$ with $\Gamma^{\mathscr{C}} = \Gamma_t^{\mathscr{C}} \bigcup \Gamma_u^{\mathscr{C}} \bigcup \Gamma_c^{\mathscr{C}}$ and $\Gamma_t^{\mathscr{C}} \bigcap \Gamma_u^{\mathscr{C}} = \emptyset$, $\Gamma_c^{\mathscr{C}} \bigcap \Gamma_u^{\mathscr{C}} = \emptyset$, $\Gamma_t^{\mathscr{C}} \bigcap \Gamma_c^{\mathscr{C}} = \emptyset$; the subscripts u, t and c indicate 'displacement-', 'traction-' and 'crack-', respectively. The area in front of the crack tip (crack front in 3D) is of particular interest and will therefore be modelled by a fine-scale domain (which can contain features of the micro-structure of the material), Fig. 1. Note that though most of the figures are illustrated in 2D, our implementation is completely three-dimensional.

2.1.1 Semi-concurrent multiscale methods

In semi-concurrent multiscale methods, information are passed from the fine-scale to the coarse-scale and vice versa. In the FE^2 method, the strain measure (e.g. deformation gradient) is passed from the coarse-scale to the fine-scale and the stress measure is sent back from the fine-scale to the coarse-scale. The stress tensor is computed based on the first-order homogenization theory [63, 64]. This framework fits within a standard continuum mechanics theory for cases where the principle of separation of scales fully holds. First order homogenization is not applicable for 'small' macro-sizes (miniaturization), analysis of geometrical size effects and engineering localization problems (failure). For non-negligible micro-structural size, significant macroscopic gradients and macroscopic localization or failure, second-order method should be used [65].

Semi-concurrent multiscale methods are very flexible and simple to implement; they can be applied to very complex micro-structures and allow different commercial or open-source tools to be coupled together without complications. For example, semi-concurrent methods were implemented in ABAQUS by Yuan and Fish [139]. Numerous studies can be found in the manuscripts by Guedes and Kikuchi [50], Feyel and Chaboche [37], Kouznetsova et al. [64], Talebi et al. [131], Fish et al. [40, 41], Ghosh et al. [44], Smit et al. [122], Miehe et al. [83] and an extension to thermal problems in Ozdemir et al. [96], Monteiro et al. [86], Larsson et al. [68] and to multi-physics problems in [97, 112].

For phenomena involving strain softening materials and cracking, the transfer of length scale is a very challenging issue when using semi-concurrent multiscale methods. As shown by Gitman et al. [47] and Bazant [14], the applicability of standard/conventional semi-concurrent methods for strain softening materials is questionable. Also as discussed in [46], the RVE looses representativeness in case of strain localization. Discontinuous semi-concurrent methods were introduced to resolve this issue. This is done by determining a cohesive law for the crack at the macro-scale from the micro-structure. For more information, please refer to manuscripts by Matous et al. [78], Kulkarni et al. [66] and Hirschberger et al. [55], Alfaro et al. [6], Nguyen et al. [91, 89, 90, 136, 93] and Verhoosel et al. [134].

One other strategy to deal with multiscale models involving cracks is the continuous-discontinuous semi-concurrent method. In a nutshell, in these models a crack is injected into

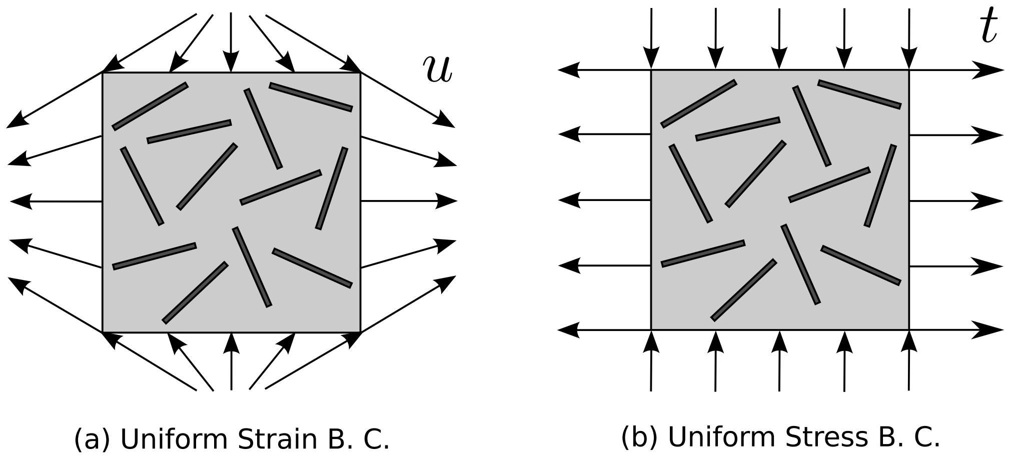

Figure 2: Different boundary condition approaches, (a) uniform strain and (b) uniform stress boundary conditions

the macroscale model when failure or instability occurs at the RVE. Please refer to the work of Massart et al. [76, 77] for the masonry cracking, Belytschko et al. [21], Belytschko and Song [22] (the MAD method), Allen et al. [123, 124], Nguyen et al.[91, 89, 90, 136] for more details.

In the following, we will explain the very basic but general theory of semi-concurrent methods that leads us to a common strategy for their computational implementation. The solution procedure for semi-concurrent multiscale methods can be described as follows. At each integration point of the coarse-scale model, a fine-scale RVE is constructed. The deformation measure from the macro level is applied to the RVE by means of boundary conditions. In the literature at least five different approaches are available to apply boundary conditions (BCs) to the RVE: periodic boundary condition (PBC), uniform stresses (constant traction), uniform strains (linear displacement BCs) [54, 52], Taylor BCs and minimal kinematic BCs [81]. Please refer to the review by Nguyen et al. [92] for detailed description of the RVE boundary conditions and their differences. For the uniform strain conditions, Fig. 2,

$$\boldsymbol{u} = \boldsymbol{E}^{\mathscr{C}} \cdot \boldsymbol{x} \quad \text{for all } \boldsymbol{x} \text{ on } \Gamma^{\mathscr{F}} \tag{1}$$

where $\boldsymbol{u}$ is the displacement vector, $\boldsymbol{x}$ are the fine-scale coordinates, $\boldsymbol{E}^{\mathscr{C}}$ is the coarse-scale strain measure and $\Gamma^{\mathscr{F}}$ denotes the boundary of the fine-scale RVE. Using the deformation gradient, the displacements of the boundary nodes are computed as:

$$\boldsymbol{u} = \boldsymbol{x} - \boldsymbol{X} = \boldsymbol{F}^{\mathscr{C}} \boldsymbol{X} - \boldsymbol{X} \tag{2}$$

where $\boldsymbol{F}^{\mathscr{C}}$ is the coarse-scale deformation gradient and $\boldsymbol{X}$ and $\boldsymbol{x}$ are the fine-scale coordinate vectors in the initial and current configuration, respectively. In the fine-scale simulations, any kind of micro-mechanical constitutive model can be used. After solving the fine-scale RVE, a homogenized stress tensor is calculated at each load increment:

$$\langle \boldsymbol{S}^{\mathscr{C}} \rangle \overset{def}{=} \frac{1}{V_{\Omega_{\mathscr{F}}}} \int_{\Omega_{\mathscr{F}}} \boldsymbol{S}^{\mathscr{F}} \, d\Omega \tag{3}$$

where $V_{\Omega_{\mathscr{F}}}$ refers to the volume of the fine-scale domain. This stress tensor will be sent back to the coarse-scale model (i.e. to the corresponding integration point). This process will continue for all integration points and for each load increment of the coarse-scale model.

2.1.2 Concurrent multiscale methods

As a whole, concurrent multiscale methods can be divided into two categories that are discrete-continuum coupling (particle-continuum, atomistic-continuum) and the continuum-continuum coupling. As it is explained in [84], atomistic-continuum methods differ from their continuum-continuum methods in four ways:

1. The governing formulation i.e. energy based (for example Quasicontinuum (QC) [129, 114], Coupling of Length Scales (CLS) [110], Bridging Domain (BD) [137], Bridging Scale Method (BSM) [135, 104], Composite Grid Atomistic Continuum Method (CACM) [33]) or force based (Finite-Element/Atomistics (FEAt) [62], Coupled Atomistics and Discrete Dislocation (CADD) [117, 118], Hybrid Simulation Method (HSM) [74], Concurrent AtC Coupling (AtC) [39, 12])

2. The Coupling Boundary Conditions i.e. how the compatibility conditions are enforced (strong or weak compatibility)

3. The Handshake Region i.e. how the transition from continuum to atomistic is treated (for

example linear mixing of energy in the BD method)

4. Treatment of the Continuum i.e. how the material properties in the continuum is modelled (for example Cauchy-Born or linear elasticity)

For the case of modeling crack and dislocation propagation using concurrent atomistic-continuum method, Shilkrot et al. [117, 118] developed a concurrent multiscale methods for propagation of dislocations. However, the method is based on the discrete dislocation method and hence restricted to linear solids. Furthermore, the method is restricted to the static case. An interesting concurrent multiscale method was proposed in the context of dynamic fracture in 2D by Xiao [137], Gracie and Belytschko [48] and dynamic fracture with XFEM in 2D by Aubertin et al. [9]. It avoids the transfer of length scales around the crack tip or dislocation core and adaptively adjusts the fine-scale domain when dislocations or cracks propagate in the fine-scale domain. The key challenge is the detection of the dislocation core or crack tip in the fine-scale domain and the adaptive adjustment of the fine-scale domain [113, 49].

For the continuum–continuum coupling methods, similar problems and strategies exist. In the case of concurrent multiscale methods for fracture, some of the methods are based on domain decomposition methods. This means the method recognizes hotspots (crack tip mostly) and models them with a fine-scale formulation. The material properties away from the crack tip is modeled with homogenized properties. The works by Guidault et al. [51], Eckardt and Könke [35] and Lloberas-Valls et al. [73] are such examples.

Another approach is based on the variational multiscale method. The foundation of this method is based on separation of the displacement field into a fine and coarse scale part. The work by Hettich et al. [53] is an examples in the area of crack propagation. Strong macro-micro coupling is another concurrent approach that was proposed in the manuscript by Ibrahimbegovic and Markovic [58]. This method is applied mostly where the scales remain coupled throughout the computations that implies a constant communication between the numerical models employed at each scale.

To treat the handshake region, the Arlequin method proposed by Dhia and Rateau [23] and the work by Lim et al. [71] are two frameworks. The Arlequin method is a very general method to couple the two scales that is based on linear weighting of energy in the handshake region. On the other hand in [71], variable-node elements are employed as transition elements that link the standard finite elements to the domain where the microstructure is explicitly modelled in detail.

Since multiscale methods are computationally expensive, adaptive multiscale methods combining the hierarchical and concurrent methods are gaining popularity. In the method presented by Ghosh et al. [45], three levels are introduced in the computational domain: macro, macromicro and microscopic scales. Continuum constitutive relations at the macro scale determine the hotspots and asymptotic homogenization is used to couple the scales in regions where periodic microstructures exist. Other studies are presented by Larsson and Runesson [67] and Temizer and Wriggers [132].

Our selected numerical method for the concurrent coupling of different length scales is the Arlequin method [23] which in the case of MD/FE coupling is known as the bridging domain method [128]. In the Arlequin method, the coarse-scale is continuously blended into a fine-scale in a so-called handshake domain $\Omega^{\mathscr{B}} = \Omega^{\mathscr{C}} \bigcap \Omega^{\mathscr{F}}$. Usually, a continuum formulation is used for the coarse-scale while both molecular and continuum formulations can be used at the fine-scale. Several scales can be coupled but in this paper and for sake of simplicity, only two scales were considered. The Arlequin method and the bridging domain methods are used since the amount of wave reflections is minimum in dynamic applications [79, 10]. The Arlequin method possesses many properties of the other multiscale methods which makes the implementation challenging. On the other hand, the implementation of the Arlequin method can be easily modified to implement other methods where suitable. Since we are offering a computational framework for multiscale analysis, this was another reason to choose the Arlequin method and the Bridging Domain Method (BDM) to present the software. The difference between the Arlequin and BDM is the using the atomistic description in the fine scale region. This will introduce some inconsistencies which led to several sub-methods that are only dealing atomistic-continuum coupling.

2.1.3 Coarse-scale formulation

Let the reference and the current configurations of the domain be denoted by Ω_0 and Ω, respectively. The material coordinates of a point in $\Omega_0^{\mathscr{C}}$ are denoted by $\mathbf{X}$ and the spatial coordinates by $\mathbf{x}$. The linear momentum conservation equations are:

$$\frac{\partial P_{ji}}{\partial X_j} + \rho_0 b_i = \rho_0 \ddot{u}_i \tag{4}$$

where $\boldsymbol{P}$ is the first Piola-Kirchhoff stress tensor, $\boldsymbol{b}$ is the body force vector per unit mass and $\boldsymbol{u}$ is the displacement vector. For hyperelasticity, the first Piola-Kirchhoff stress tensor can be calculated from a continuum potential:

$$\boldsymbol{P} = \frac{\partial w^{\mathscr{C}}(\boldsymbol{F})}{\partial \boldsymbol{F}} \tag{5}$$

where $w^{\mathscr{C}}$ is the potential energy per unit volume and $\boldsymbol{F}$ is the deformation gradient tensor. The total potential energy of the coarse model is:

$$W^{\mathscr{C}}_{int} = \int_{\Omega_0^{\mathscr{C}}} w^{\mathscr{C}}(\boldsymbol{F}) d\Omega_0^{\mathscr{C}} \tag{6}$$

In an isolated system, the sum of the potential and kinetic energies is constant in time. This summation is identified as the Hamiltonian. At the coarse-scale, the Hamiltonian is given by:

$$H^{\mathscr{C}} = K^{\mathscr{C}} + W^{\mathscr{C}} = \int_{\Omega_0^{\mathscr{C}}} \frac{1}{2} \rho \boldsymbol{v}^T \boldsymbol{v} d\Omega_0^{\mathscr{C}} + W^{\mathscr{C}} \tag{7}$$

$$W^{\mathscr{C}} = -W^{\mathscr{C}}_{ext} + W^{\mathscr{C}}_{int} = -\int_{\Omega^{\mathscr{C}}} \boldsymbol{b}.\boldsymbol{u} d\Omega^{\mathscr{C}} - \int_{\Gamma_t^{\mathscr{C}}} \bar{\boldsymbol{t}}.\boldsymbol{u} d\Gamma^{\mathscr{C}} + \int_{\Omega_0^{\mathscr{C}}} w^{\mathscr{C}}(\boldsymbol{F}) d\Omega_0^{\mathscr{C}} \tag{8}$$

where $K^{\mathscr{C}}$ and $W^{\mathscr{C}}$ are kinetic and potential energies in the coarse-scale. $\boldsymbol{v}$ denotes the velocity and $\boldsymbol{b}$ is the body forces acting on the coarse- scale which is situated "far" from the influence of the crack tip.

We use the eXtended Finite Element Method (XFEM) to model fracture in the continuum domain. As the area around the macroscopic crack tip is modeled by the fine-scale, a purely step-enriched XFEM-approach is sufficient at the coarse-scale:

$$\boldsymbol{u}^h(\boldsymbol{X}) = \underbrace{\sum_{I \in \mathcal{N}} N_I(\boldsymbol{X}) \boldsymbol{u}_I}_{\mathbf{u}^{\text{cont}}} + \underbrace{\sum_{I \in \mathcal{N}_b} N_I(\boldsymbol{X}) S(f_I(\boldsymbol{X})) \boldsymbol{a}_I}_{\mathbf{u}^{\text{discont}}} \tag{9}$$

where $\mathcal{N}$ is the set of all nodes in the domain and $\mathcal{N}_b$ is the set of nodes that belong to all elements which are completely cut by the crack; $\boldsymbol{u}_I$ and $\boldsymbol{a}_I$ are the standard and the enriched degrees of freedom, respectively and S is the discontinuous enrichment (Heaviside) function:

$$S\left(f\left(\boldsymbol{X}\right)\right) = \begin{cases} 1 \quad \text{if} \quad f\left(\boldsymbol{X}\right) > 0 \\ 0 \quad \text{if} \quad f\left(\boldsymbol{X}\right) < 0 \end{cases} \tag{10}$$

with

$$f\left(\boldsymbol{X}\right) = \operatorname{sign}\left[\boldsymbol{n} \cdot \left(\boldsymbol{X}_I - \boldsymbol{X}\right)\right] \min_{\mathbf{X_I} \in \Gamma} \left\|\boldsymbol{X}_I - \boldsymbol{X}\right\| \tag{11}$$

where $\boldsymbol{n}$ is the outward normal to the crack surface.

$$\boldsymbol{u}^h\left(\boldsymbol{X}\right) = \underbrace{\sum_{I \in \mathcal{N}} N_I\left(\boldsymbol{X}\right)\boldsymbol{u}_I}_{\mathbf{u}^{\text{cont}}} + \underbrace{\sum_{I \in \mathcal{N}_b} N_I\left(\boldsymbol{X}\right) S\left(f_I\left(\boldsymbol{X}\right)\right)\boldsymbol{a}_I + \sum_{I \in \mathcal{N}_t} N_I\left(\boldsymbol{X}\right) \sum_K \boldsymbol{B}_K\left(\boldsymbol{X}\right)\boldsymbol{b}_{KI}}_{\mathbf{u}^{\text{discont}}} \tag{12}$$

where $\mathcal{N}_t$ are the set of nodes that are influenced by the crack tip and $\boldsymbol{B}$ is a set of branch functions. In PERMIX, the following set of branch functions are used:

$$\boldsymbol{B} = [B_1,\, B_2,\, B_3,\, B_4] = [\sqrt{r}sin(\frac{\theta}{2}), \sqrt{r}cos(\frac{\theta}{2}), \sqrt{r}sin(\theta)sin(\frac{\theta}{2}), \sqrt{r}sin(\theta)cos(\frac{\theta}{2})] \tag{13}$$

where r and θ are the local polar co-ordinates defined at the crack tip.

2.1.4 Fine-scale formulation

Continuum and atomistic formulations can coexist in the fine-scale domain. In case of a continuum fine-scale, the Hamiltonian can be expressed in a similar way to the Hamiltonian of the coarse-scale. Therefore, only the Hamiltonian of the atomistic formulation of the fine-scale is presented here. For the atomistic subdomain, the Hamiltonian can be expressed as:

$$H^{\mathscr{F}}(\boldsymbol{x}_\alpha(t), \boldsymbol{p}_\alpha^{\mathscr{F}}(t)) = \sum_\alpha \frac{1}{2m_\alpha}\boldsymbol{p}_\alpha^{\mathscr{F}} \cdot \boldsymbol{p}_\alpha^{\mathscr{F}} + W^{\mathscr{F}}(\boldsymbol{x}_\alpha(t)) \tag{14}$$

where $\boldsymbol{x}_\alpha$ and m_α are the current position vector and mass of atom α respectively. The location of atom α in the reference and spatial configurations can be related by the displacement vector

$\boldsymbol{d}$:

$$\boldsymbol{x}_\alpha = \boldsymbol{X}_\alpha + \boldsymbol{d}_\alpha \tag{15}$$

$\boldsymbol{p}_\alpha^{\mathscr{F}}$ is the momentum of atom α and defined by

$$\boldsymbol{p}_\alpha^{\mathscr{F}} = m_\alpha \dot{\boldsymbol{x}}_\alpha = m_\alpha \dot{\boldsymbol{d}}_\alpha \tag{16}$$

$W^{\mathscr{F}}(\boldsymbol{x})$ is the potential function due to any kind of force fields, such as pair-wise interactions or three-body potentials:

$$W^{\mathscr{F}}(\boldsymbol{x}_\alpha) = \sum_\alpha W_1(\boldsymbol{x}_\alpha) + \sum_{\alpha,\beta>\alpha} W_2(\boldsymbol{x}_\alpha, \boldsymbol{x}_\beta) + ... \tag{17}$$

Assuming the external potential results only from a constant external force, $\boldsymbol{f}_\alpha^{ext}$, and a pair-wise potential, the total potential can be expressed as:

$$W^{\mathscr{F}} = -W_{ext}^{\mathscr{F}} + W_{int}^{\mathscr{F}} = -\sum_\alpha \boldsymbol{f}_{ext\,\alpha} \boldsymbol{d}_\alpha + \sum_{\alpha,\beta>\alpha} W^{\mathscr{F}}(\boldsymbol{x}_\alpha, \boldsymbol{x}_\beta) \tag{18}$$

The canonical form of the Hamiltonian equations of the fine-scale problem can be expressed as:

$$\begin{cases} \dot{\boldsymbol{p}}_I^{\mathscr{F}} = -\frac{\partial H}{\partial \boldsymbol{x}_I} = -\frac{\partial W^{\mathscr{F}}}{\partial \boldsymbol{x}_I} \\ \dot{\boldsymbol{x}}_I = \dot{\boldsymbol{d}}_I = \frac{\partial H}{\partial \boldsymbol{p}_I^{\mathscr{F}}} = \frac{\boldsymbol{p}_I^{\mathscr{F}}}{m_I} \end{cases} \tag{19}$$

Finally, these two equations can be combined to obtain:

$$m_\alpha \ddot{\boldsymbol{d}}_\alpha = -\frac{\partial W^{\mathscr{F}}}{\partial \boldsymbol{x}_\alpha} = \frac{\partial W_{ext}^{\mathscr{F}}}{\partial \boldsymbol{d}_\alpha} - \frac{\partial W_{int}^{\mathscr{F}}}{\partial \boldsymbol{d}_\alpha} = \boldsymbol{f}_{ext\,\alpha} - \boldsymbol{f}_{int\,\alpha} \tag{20}$$

In this equation, $\boldsymbol{f}_{int}$ is the internal force vector.

The modeling of cracks in the fine-scale region can be also handled with the conventional strategies. When a continuum formulation is used, we use XFEM to represent cracks in the fine-scale region as explained in the previous section. When molecular dynamics is used, the cracks initiate naturally since the nature of atomistic modeling can represent crack surfaces

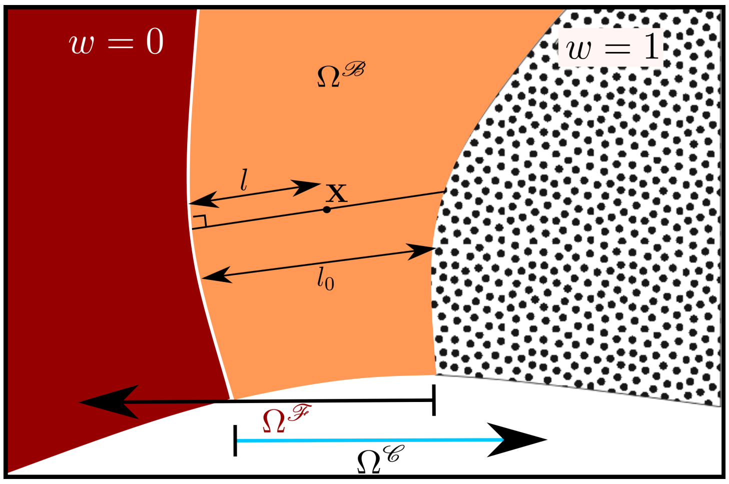

Figure 3: The weighting function in the handshake domain in two dimensions

automatically. To introduce a pre-crack in the atomistic region, we modify the neighbor lists to artificially reproduce a crack similar to the visibility criterion used in meshfree methods [20], where atoms do not "see" each other through the crack.

2.1.5 Coupling Method

In the Arlequin method, the total energy of the system is a weighted average of the fine and coarse models in the bridging domain $\Omega^{\mathscr{B}}$. To implement this concept, a scalar weight function, w is defined which is unity outside the fine-scale domain, zero inside the fine-scale and linearly increasing/decreasing in the blending region:

$$w(\boldsymbol{X}) \begin{cases} = 0 & \forall \mathbf{X} \in \Omega^{\mathscr{C}} \setminus \Omega^{\mathscr{F}} \\ \in [0,1] & \forall \mathbf{X} \in \Omega^{\mathscr{B}} \\ = 1 & \forall \mathbf{X} \in \Omega^{\mathscr{F}} \setminus \Omega^{\mathscr{C}} \end{cases} \tag{21}$$

The weight function w at any point $\mathbf{X}$ can be computed by a normalized distance function:

$$w(\boldsymbol{X}) = \frac{l(\boldsymbol{X})}{l_0} \tag{22}$$

where $l(\boldsymbol{X})$ is the orthogonal projection of $\boldsymbol{X}$ onto the interior boundary of the coarse-scale domain $\Omega^{\mathscr{C}}$ and l_0 is the length of this orthogonal projection to the boundary of the fine-scale $\Omega^{\mathscr{F}}$, Fig. 3.

The governing equations are derived from the Hamiltonian of the systems, H, which is the sum of the Hamiltonians of each subdomain:

$$\begin{aligned} H &= wH^{\mathscr{F}} + (1-w)H^{\mathscr{C}} \\ &= \sum_{\alpha} w(\boldsymbol{X}_{\alpha}) \frac{\boldsymbol{p}_{\alpha}^{\mathscr{F}} \cdot \boldsymbol{p}_{\alpha}^{\mathscr{F}}}{2m_{\alpha}} + wW^{\mathscr{F}} + \sum_{I} (1 - w(\boldsymbol{X}_I)) \frac{\boldsymbol{p}_I^{\mathscr{C}} \cdot \boldsymbol{p}_I^{\mathscr{C}}}{2M_I} + (1-w)W^{\mathscr{C}} \end{aligned} \tag{23}$$

where $H^{\mathscr{F}}$ and $H^{\mathscr{C}}$ are the Hamiltonians of the fine and coarse sub-domains.

The coarse and fine-scale domains are constrained on the bridging domain, $\Omega^{\mathscr{B}}$ by the Lagrange multiplier method. In this overlapping domain, the fine-scale displacements are required to conform to the coarse-scale displacements. In the Lagrange multiplier method, the total Hamiltonian is written as:

$$H_L = H + \boldsymbol{\lambda}^T \boldsymbol{g} \tag{24}$$

where $\boldsymbol{\lambda}$ is the Lagrange multipliers vector and $\boldsymbol{g} = \boldsymbol{u}^{\mathscr{C}} - \boldsymbol{u}^{\mathscr{F}}$ is the gap vector between coarse-scale displacement and fine-scale displacement. To compute the Lagrange multiplier unknowns we use the method described in [128]. Finally, we obtain the following semi-discrete equations of motion:

$$\forall I \in \mathcal{N}^{\mathscr{C}}\ \forall i \in \{1,2,3\}\ :\ M_{IJ}\ddot{u}_{Ji} = f_{Ii}^{\text{ext}} - f_{Ii}^{int} + f_{Ii}^{\lambda\mathscr{C}} \quad , \tag{25}$$

$$\forall \alpha \in [\![1,\ldots,n^{\mathscr{F}}]\!],\ \forall i \in \{1,2,3\}\ :\ m_{\alpha}^{\mathscr{F}}\ddot{d}_{\alpha i}^{\mathscr{F}} = f_{\alpha i}^{\mathscr{F}} + f_{\alpha i}^{\lambda\mathscr{F}} \quad , \tag{26}$$

where $\ddot{u}_J$ and $\ddot{d}_{\alpha}^{A}$ are the accelerations of node J and atom α, respectively, $m_{\alpha}^{\mathscr{F}}$ is the mass of atom α and M_{IJ} is the consistent mass matrix:

$$\forall I, J \in \mathcal{N}^{\mathscr{C}}\ :\ M_{IJ} = \int_{\Omega_0^{\mathscr{C}}} (1-w)\,\rho_0 N_I N_J d\Omega_0^{\mathscr{C}} \quad , \tag{27}$$

The mass matrix of the coarse-scale is diagonalized according to the mass-lumping scheme for XFEM proposed in [80]. The internal forces at the coarse-scale are:

$$\forall I \in \mathcal{N}^{\mathscr{C}},\ \forall i \in \{1,2,3\}\ :\ f_{Ii}^{int} = \int_{\Omega_0^{\mathscr{C}}} (1-w)\,P_{ij}\frac{\partial N_I}{\partial X_j} d\Omega_0^{\mathscr{C}}, \tag{28}$$

The external forces on each atom are determined from the interatomic potential W as:

$$\forall \alpha \in [\![1 \dots n^{\mathscr{F}}]\!], \forall i \in \{1,2,3\} \, : \, f_{\alpha i}^{\mathscr{F}} = -\sum_{\beta} \frac{1}{2} \left(w\left(X_\alpha\right) + w\left(X_\beta\right) \right) \frac{\partial W\left(r_{\alpha\beta}\right)}{\partial d_{i\beta}^{\mathscr{F}}} \quad , \tag{29}$$

where β ranges over all atoms within the cutoff radius of atom α. The forces $f^{\lambda \mathscr{C}}$ in the coarse-scale domain and $f^{\lambda \mathscr{F}}$ in the fine-scale domain, due to the coupling are given by:

$$\forall I \in \mathcal{N}^{\mathscr{C}}, \forall i \in \{1,2,3\} \, : \, f_{Ii}^{\lambda \mathscr{C}} = \sum_{\alpha \in \Omega_0^{\mathscr{B}}} \lambda_{\alpha i} N_I \left(X_\alpha\right) \quad , \tag{30}$$

and

$$\forall \alpha \in [\![1 \dots n^{\mathscr{F}}]\!], \forall i \in \{1,2,3\} \, : \, f_{\alpha i}^{\lambda \mathscr{F}} = -\lambda_{\alpha i}. \tag{31}$$

2.2 Crack propagation

When the molecular scale model is described in terms of first principle molecular dynamic/mechanics, the crack propagation and nucleation occurs naturally by breaking bonds between atoms. However, at the continuum mechanics level, crack nucleation and propagation must be taken care of. Once a crack initiates, the nucleation and propagation of the crack at each time step should be checked and if the crack propagates, the propagation direction of the crack should be determined. Different methods such as maximum circumferential stress criterion (MCSC), minimum strain energy density criterion (MSEDC) and maximum strain energy release rate criterion (MSERRC) are available to determine the propagation direction [27].

The loss of hyperbolicity criterion is another criterion for crack nucleation and propagation [18] which is used in PERMIX. Assume that the rate form of the constitutive equations are

$$\boldsymbol{\sigma}^{\nabla}(\mathbf{X}, t) = \mathbf{C}^t(\mathbf{X}, t) : \mathbf{D}(\mathbf{X}, t) \tag{32}$$

where $\mathbf{C}^t$ is the tangential stiffness and $\mathbf{D}$ is the velocity gradient; the superscript ∇ denotes the Truesdell rate. Eq. (32) loses hyperbolicity when

$$e = \min_{\mathbf{n},\mathbf{h}} (\mathbf{n} \otimes \mathbf{h} : \mathbf{A}(\mathbf{X}, t) : \mathbf{n} \otimes \mathbf{h}) \leq 0 \tag{33}$$

where $\mathbf{A} = \mathbf{C}^t + \boldsymbol{\sigma} \otimes \mathbf{I}$ and $\mathbf{n}$ and $\mathbf{h}$ are two arbitrary unit vectors. Loss of hyperbolicity is determined by minimizing e with respect to $\mathbf{n}$ and $\mathbf{h}$; if e is negative for any combination of $\mathbf{n}$ and $\mathbf{h}$, the PDE has lost hyperbolicity at that material point. Hyperbolicity can also be checked by conditions on the acoustic tensor $\mathbf{Q} = \mathbf{n} \cdot \mathbf{A} \cdot \mathbf{n}$. A material point loses stability if the minimum eigenvalue of the acoustic tensor is smaller or equal to zero:

$$\min \operatorname{eig}(\mathbf{Q}) \leq 0 \tag{34}$$

where $\mathbf{n}$ is the normal to crack surface when $\min \operatorname{eig}(\mathbf{Q}) \leq 0$.

2.3 Molecular Dynamics

Molecular dynamics (MD) [7] is one of the simplest but most insight generating simulation techniques in modern materials modeling. This method was first introduced by Alder and Wainwright in the late 1950's [5] to study the interactions of hard spheres, and was advanced to model water by Rahman in 1964 [107]. MD predicts the motion of atoms governed by their mutual interatomic interaction, and to do so, requires numerical integration of the equations of motion of each particle

$$\mathbf{f}^{\mathscr{A}} = m\ddot{\mathbf{r}} \tag{35}$$

where $\mathbf{f}^{\mathscr{A}}$ is the force vector on the particle, m the particle mass and $\ddot{\mathbf{r}}$ is the acceleration vector. The force vector $\mathbf{f}^{\mathscr{A}}$ for each particle is given by the gradient of the total energy at a given particle location in space:

$$\mathbf{f}^{\mathscr{A}} = -\nabla E_{tot}^{\mathscr{A}} \tag{36}$$

where

$$E_{tot}^{\mathscr{A}} = \sum_i \phi_i \tag{37}$$

denotes the total energy of the system which is obtained by summing the potential energy of all particles.

The forces between particles can be short-ranged or long-ranged. In case of short-ranged forces only particles within a critical cutoff sphere contribute to the energy of the atom. The equations of motion Eq. (35) are usually integrated using a velocity Verlet scheme [7, 127] which is similar to the central difference method for explicit time integration. Therefore, in

PERMIX the main routine for explicit dynamic solution handles all atomistic and finite element components together, when running with the same time step.

3 The PERMIX Code

The object oriented design of PERMIX allows the program to include several different components of one system at several scales with different formulation concepts. The software provides its own Extended Finite Element implementation both in statics and dynamics. Several powerful parallel linear direct and iterative solvers were added to the program in addition to a serial linear solver. PERMIX is platform independent and known to be built and executed on Windows, Linux and MacOS operating systems. In general, any standard F2003 compiler can be used to build PERMIX. We will continue with describing the overall design of the program and give more details about each component in the following sections.

PERMIX is implemented in Fortran 2003. Fortran was invented back in the 1950's for numerical analysis. Since then, Fortran has been regularly updated to meet the new concepts in the software engineering [75]. The latest standard is Fortran 2008, containing only minor modification of the 2003 standard. The F2003 standard allows for object oriented programming concepts such as data encapsulation, inheritance, polymorphism, among others [109]. There are several other programs written mostly in C++ such as OOFEM [99], deal. II [13] and OpenXFEM++ [25]. However, to our best of knowledge, there is no multiscale code for fracture available in the open source domain.

3.1 Object Oriented Design

The PERMIX code exploits many features of the Fortran 2003 standard to produce an efficient and extensible code which can deal with several types of physical systems. PERMIX uses derived data types (like classes in C++), type extensions (inheritance in C++) and polymorphism for the main objects. However for efficiency reasons, the small objects are saved mainly in vectors and matrices rather than derived data types. This enables especial Fortran compiler optimization and finally will result in higher single thread performance. Moreover, in case of distributed memory communication using MPI, transferring data among processors is also easier and faster. The main PERMIX class has several other classes included which are shown

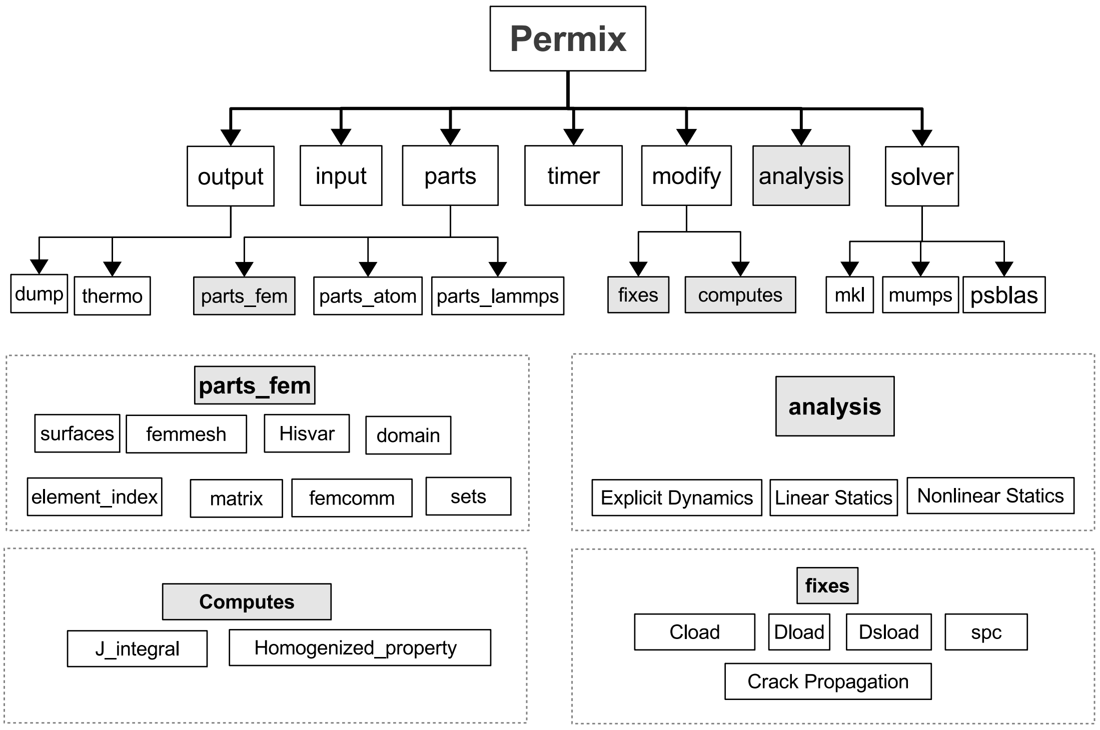

Figure 4: The Object Oriented design of PERMIX with the main objects

in Fig. 4. The sub-classes shown in Fig. 4 are the most important classes and they might have sub-classes as well.

For classes which contain similar methods and properties, a base class was defined via Fortran `abstract` types. Then, the base class is extended to a final class (through derived data type extension). The abstract types contain several properties and methods (`procedure`s). The procedures can be `deferred` or `nonoverridable`. The `deferred` procedures must be defined when an extended type is generated for further program extension. The procedures without the `nonoverridable` attribute can be overridden with another procedure to conduct a specific task.

All extended derived types of an initial base class are grouped in one Fortran vector containing an allocatable polymorphic variable of the base class; and based on the input, the polymorphic data inside the vector is allocated accordingly. A similar method is used in the LAMMPS C++ code to store the objects and in PERMIX a slightly modified Fortran version of it is implemented. To distinguish the type of allocation for the polymorphic variables, the `Type is` command in Fortran is used. In case of adding a new style, a small script in the Perl language is also written to handle the changes in the object allocation subroutine. This way, new styles can be added with minimum modification of other parts of the code and without having an overview on the many other classes.

We explain the above concept with an example. To create different Finite Elements types, a base class is listed in listing 1. As it is clear from the listing, every element must define the material ID or number of degrees of freedom (DOF) per node or the total number of DOFs per element etc.

Listing 1: A base derived data type for different finite element types

```
type,abstract :: ty_FEM_element_base
   character(30) :: style=""          ! style of the element
   logical :: if_f_internal=.false.   ! series of variables showing the capabilities
   logical :: if_mat_stiffness=.false. ! of the element i.e. large def., dynamic etc.
   logical :: if_inital_stiffness=.false. !if capable of computing stiffness.
      ...
   integer :: setup_flag =0 ! if the initial setup of the element is done
   integer :: lg=0          ! total number of unknowns for the element
   logical :: iffix=.false. ! if the element has a boundary
   integer :: el_id=0       ! the element id
   integer :: mat_id=0      ! material id
   integer :: etype=0       ! element type number
   integer :: ndime=0       ! number of dimensions
   integer :: nnode=0       ! number of nodes per element
      ...
   contains
      procedure :: init             => FEM_element_base_init
      procedure :: setup            => FEM_element_base_setup
      procedure :: check_capability => FEM_element_base_check_capability
      procedure :: compute_4one     => FEM_element_base_compute_4one
      procedure :: compute_4all     => FEM_element_base_compute_4all
end type ty_FEM_element_base
```

In listing 2, an allocatable polymorphic variable of the base element types is defined. In order to define many elements, an array of the allocatable polymorphic variables is defined in the same listing. Therefore, the variable `all_elements` has all the defined elements in the PERMIX program. Every element of the array `all_elements` will define an element and after proper initial setup can provide information on that element or can be asked to perform a specific task. For example one can blindly access the number of the nodes for element type 3 by `all_elements(3)%OBJ%nnode`.

Listing 2: An array containing polymorphic variables of the base type

```
type ptr_ty_FEM_element_base
```

```
        class(ty_FEM_element_base), allocatable :: OBJ
end type ptr_ty_FEM_element_base
type(ptr_ty_FEM_element_base)  :: all_elements(NUM_DFND_ELS)
```

An example for the type extension of the base type is provided in listing 3. As can be seen, an extended type can override the base procedures and add new properties to the extended type.

Listing 3: An example of type extension to define a 3D tetrahedral element

```
type,extends(ty_FEM_element_base) :: ty_FEM_V4N_fast
   real(RWP) :: vinc(FEM_V4N_fast_lg) !newly defined variable
   real(RWP) :: kel(FEM_V4N_fast_lg,FEM_V4N_fast_lg) !newly defined variable
   ...
   contains
      procedure :: compute_4one      => FEM_V4N_fast_compute_4one !overridden
      procedure :: setup             => FEM_V4N_fast_setup       !overridden
      procedure :: extract_elinfo    => FEM_V4N_fast_extract_elinfo !new procedure
      procedure :: give_etype_string => FEM_V4N_fast_give_etype_string
      procedure :: define_compatibility => FEM_V4N_define_compatibility
end type ty_FEM_V4N_fast
```

3.2 Base Classes

As explained in the previous section, the versatile extensibility of PERMIX is guaranteed via the base classes. Almost any type of multiscale method can be implemented using the existing base classes. Moreover, new numerical methods such as meshfree methods, boundary conditions, element types etc. can be added using the base classes. The base classes are Fortran abstract derived data types which provide a generic means to implementation paradigm for new numerical methods. In the sequel, we will explain the base classes and their area of usefulness.

Part: Parts are objects by which PERMIX defines its computational models. For example two finite element parts and an atomistic part can be defined in a multiscale model. One can define several parts which can interact. The `part_fem` class contains all the information and methods about one finite element part. This information includes the mesh data, matrices and vectors, domain decomposition, neighbor searches, sets, among many others. For example, `ty_femmesh` class contains a two dimensional integer array to store the element connectivities and other data about an element. A similar array stores the node data such as number of DOFs for a node or the boundary conditions on that node.

Fix: Fixes are part of the modify class and are the routines that are called at different stages of a solution procedure to perform a specific task on certain parts. For example, boundary conditions are grouped into fixes and they can be applied to specific sets of nodes or elements of `part_fem` objects. Fixes have access to all parts and this method offers great flexibility which does not require altering the main solution routine when adding a new fix.

Solver: The solver base class is the mean for PERMIX to implement its own linear equation solvers or interfaces with shared and distributed memory solvers. The solver base accepts a general two dimensional matrices (in sparse or dense form) and the right hand side and solves the system of equations. Current interfaces to external solvers are MKL [59], PARDISO [111], MUMPS [8] and PSBLAS solvers.

Interaction: Interaction is a way of coupling two parts. Examples of interaction is coupling the system of equations, multiscale coupling or contact. All concurrent multiscale methods can be defined as an interaction.

Compute: The variable computation classes are those used to calculate a certain property of interest. The property can be of any type i.e. per node, per element, per integration point or for the whole system. For example a "compute" can be defined to compute the effective stress at the integration points or to compute the homogenized stress of the whole FE part.

Output: The output base class is in charge of all disk outputs from PERMIX. The output is capable of extracting the information from all parts and sending them to disk. The extended types from output will determine the post-processing tool/software which the output is written for. The output base class is also capable of automatically gathering the output of compute classes and write it to disk. The current output styles are Tecplot [16], Gmsh [43] and Paraview [3].

Step: The step base class is the mean of adding solution strategies to PERMIX. The extended step classes can for example implement a nonlinear Newton solver with the arc-length method [24, 30, 42] or an explicit dynamic solver [19]. The step classes work closely with the Analysis class which keeps the information about the analysis. All the parts can participate in a step procedure. The polymorphic step objects are constructed at the time of solution and will be destructed right after finishing the step.

Material: To define the material constitutive models at the integration point level, the material base class can be extended. The material class also has the access to the finite element mesh and its integration points or in general the discretization data (can be Meshfree methods for example). The material classes can also define a full PERMIX object inside for the semi-concurrent multiscale methods. For crack problems, the materials can also provide a stability check routine which determines the instance of material point failure. This simply means one can simulate cracking from simple one scale material models such as the Lemaitre damage [69] law or from a complex multiscale material model.

Element: To add new finite element types, the element base class can be used. The step class determines the element property to compute. For example, the explicit dynamic step routine, determines that the element stiffness need not be computed or internal forces calculated. The elements can also define a stability routine which inserts a crack into an element based on an arbitrary criterion. The element property calculation can be parallelized with OpenMP [31] for shared memory systems.

3.3 Input

There are two ways to input parameters into PERMIX. One is to write a Fortran main file and compile PERMIX as a library and call the different routines of PERMIX from the main file. This way, the objects can be filled from the semi-script type of input and can be very general. The second way of input into PERMIX is through the input class which offers reading the input data from a text file as many other commercial/open-source finite element packages do. The text input was designed to be very intuitive. The PERMIX input commands start with `*COMMAND` and end by `*END_COMMAND`. Inside this block all other properties of the object are defined. The input class will recognize the command and will call the input reader of the corresponding class. Therefore, each class has its own input reader.

3.4 Two and Three Dimensional Extended Finite Elements

The enriched elements and nodes demand information which is particular to the XFEM itself. For example, an enriched element should know the level set values of the crack surface, crack tip information, triangulation points and connectivities, among others. Since these data are not similar to the ordinary finite element data, we created an additional array of polymorphic variables that might contain more information about the elements. The additional information are only activated when needed depending on the type of element. This additional information can be also used for other purposes other than XFEM.

For elements containing cracks, a routine is called to create the extra information i.e. create enriched elements. In this routine the enrichment rules for the nodes and integration points are determined. We triangulate the elements cut by the crack and map the integration points accordingly. The triangulation in 2D and tetrahedralization in 3D is done with two external open-source libraries TETGEN [119] and TRIANGLE [116]. Along with triangulation, there are several computational geometry tasks which are essential to XFEM. Therefore, we also imported the open-source computational geometry library GEOMPACK [11] in Fortran 90 to PERMIX.

Since the cracks in XFEM are embedded inside the element, one cannot directly visualize the cracks explicitly with the existing post-processing tools. To visualize the crack surface, we create two sets of triangles, above and below the crack surface. The location of the triangles are updated with the elements as they deform. Using this method, the enriched elements can be visualized, complying with existing post-processing tools created for ordinary finite elements.

3.5 Crack Propagation

When defined in the input model, the crack propagation fix handles the nucleation and propagation of cracks in the model. Please note that the cracks defined in the initial conditions cannot propagate unless the crack propagation class is active. The general algorithm for crack propagation is as follows. At every load or time step PERMIX searches for the failed elements with a crack path inside the element. This is accomplished via an element routine that can handle crack nucleation and propagation, see section 2.2 for details. Inside the element routine, the cracking criteria are checked at integration points and a crack path is derived from the failed integration points. These routines are specific to the material models. For example, for a linear elastic material, the maximum principle stress or strain can be defined as a measure of failure of the point.

For multiscale materials, the failure of the point can be computed based on some measure computed by the homogenization rules. The multiscale material can define a separate compute object which quantifies the measure of instability. After checking the stability of the point, the crack tip will be propagated to the failed point or a crack will be initiated in the element by means of level sets. During the propagation process, the element types will be changed from uncracked to cracked XFEM types and the history variables will be projected to the new points, if needed.

3.6 Atomistic Part

For the atomistic part (`part_atom`) a code named WARP [102] is adopted. The WARP is a parallel molecular dynamics simulation package and the antecedent of the LAMMPS Molecular Dynamics Simulator [101] for modeling stress and strain in materials using the Lennard-Jones (LJ), embedded atom method (EAM) and modified embedded atom method (MEAM) potentials. It is implemented in Fortran 90 and uses MPI to perform message-passing. WARP already has efficient force computation algorithms and neighbor search essential in three dimensional simulations. The WARP code was modified and merged into PERMIX. Note that the potentials in WARP had to be modified to account for weighting of the atoms in the bridging domain method.

4 Interface Generation

One key issue in the design of a multiscale software is the interface generation among different components of the software that can also be external libraries. In order of efficiency, interfaces can be through files, conversion of data structures in memory, or direct data access between components. Interface generation through files can be used where there is limited information about the data structure of the external library or no wrapping code can be generated to exchange data in memory. This is often the case when using commercial products.

The conversion of data will be necessary when the data structure of the external library is complex or written in a different language or style that cannot be directly modified. When this occurs, another solution can be the implementation of the data modifier routine in the external library as a plugin to avoid data conversion. The third and the most efficient method is to create a wrapper code around the external library and call the routines or use the objects via the wrapper. For C or C++ tools such as SWIG [15] provide automatic creation of the wrapper code. Babel [36] is another interesting language interoperability tool which is especially designed to be high-performance.

In PERMIX we use all three methods to connect to different libraries. And since in PERMIX we often use Fortran/C/C++ external packages and libraries, we use the `ISO_C_BINDING` module of Fortran 2003 as a versatile way to interface with C or C++ libraries. If the external library is written in C or C++ with objects saved in an straightforward fashion, the direct data access approach can be used with Fortran pointers mapped to the C pointers. PERMIX currently interfaces with several external libraries such as LAMMPS, GMSH, Abaqus among others.

4.1 Interface to LAMMPS

The (`part_atom`) provides a practical and standalone simulation tool for metallic systems which enables simulation of molecular systems in PERMIX in a simplified manner. However, WARP is not updated any more and it is limited to three potentials only. Thus, we decided to use LAMMPS as one of the most powerful open-source atomistic simulation software. The connection between LAMMPS and PERMIX is established through a Fortran2003 interface exploiting the `ISO_C_BINDING` module of the Fortran 2003 standard.

LAMMPS, is written in C++ but is very portable since it uses the basic pointers to store vector data. These pointers are similar to the Fortran vectors and matrices. For each class in LAMMPS, we created a Fortran 2003 derived data type and for each variable in classes of LAMMPS, we defined Fortran pointers. At the beginning of the simulation, we map the Fortran pointers to the C++ pointers. This method allows the access to every `public` property of the classes of LAMMPS via Fortran pointers. The handle to the main LAMMPS pointer or other LAMMPS subclasses is saved with a `C_PTR` type. Therefore, we can define many LAMMPS parts in a concurrent simulation (since we have direct access to the variables) or a semi-concurrent simulation inside a material class of PERMIX for example. Currently, the former is used to implement the bridging domain method and the latter is used to create a material model with the Cauchy-Born assumption [129].

5 Multiscale Methods in Practice

In this section we discuss the implementation details of semi-concurrent and concurrent multiscale methods implemented and used in the PERMIX platform. The features explained in this section will shed light on future extensions of such methods using other tools. Concurrent coupling of PERMIX to commercial software is problematic because the commercial products give limited access to their global data structure. The user subroutines in commercial software provide only limited ways of implementing new materials or elements or boundary conditions. As a results, PERMIX currently supports only semi-concurrent coupling to two commercial FE packages: ABAQUS and LSDYNA which are explained subsequently. For concurrent coupling, several FE parts of PERMIX can be coupled to other FE parts from PERMIX or to the LAMMPS or the Atomistic part.

5.1 Hierarchical Style

In PERMIX, the hierarchical multiscale methods are used to extract the elastic properties of the material from a lower scale. In general, the elasticity tensor has 81 components which 21 of them are independent. It means that for the general anisotropy case, one needs at least 21 independent equations to calculate 21 components of the elasticity tensor. In the case of displacement boundary conditions and using six strain control tests, 81 equations are generated. For the isotropic RVE, the elasticity tensor has two independent parameters. Linear material properties in PERMIX are then characterized by solving a forward homogenization problem. PERMIX has a special `step` routine to do the task of hierarchical multiscale material property calculation. This routine calls the first order homogenization `compute` for both anisotropic and isotropic cases. In this case, the micro-scale properties are the user-defined (inputs) properties and the corresponding macro-scale properties are evaluated (outputs).

5.2 Semi-concurrent Style

To use semi-concurrent multiscale methods in PERMIX, three schemes are available: PERMIX-ABAQUS [2] coupling, PERMIX-LSDYNA and PERMIX-PERMIX coupling. The implementation is carried out in a way that both ABAQUS/LSDYNA packages and PERMIX can be used in each of the fine-scale and coarse-scale. PERMIX-LSDYNA and PERMIX-PERMIX coupling is done in the same manner and therefore not explained. When PERMIX is the master program (the coarse-scale), an ABAQUS part (`ty_abq_part`) is defined. This ABAQUS part contains routines to call the ABAQUS software with some external commands. With the use of some additional user-subroutines in ABAQUS, the output of ABAQUS is dumped to the disk in a specific manner readable by PERMIX. The ABAQUS part will then read the results and calls the homogenization compute to calculate homogenized stress for example.

Fig. 5 illustrates the multiscale flowchart when ABAQUS is used for fine-scale and PERMIX for coarse-scale. This figure also shows the ABAQUS subroutines [1] which are used in this multiscale approach. PERMIX calculates the deformation gradient for each integration point of the coarse-scale and passes the deformation gradient to ABAQUS. This deformation gradient is used to apply a uniform strain on the RVE boundary, Eq. (1) (Udisp.f). When the PERMIX-PERMIX semi-concurrent multiscale style is used, a fix is implemented which applies different type of boundary conditions to the RVE such as uniform traction or linear displacement BCs. The advantages of this coupling is that all ABAQUS material models can be used inside the RVE (in the general case, user material subroutines can be employed for the fine-scale material model). After solving the fine-scale model and based on Eq. (3), a homogenized stress tensor is calculated and sent back to the coarse model (Homogen.f). This process is carried out for all integration points of the coarse-scale model.

There is another semi-concurrent multiscale method in PERMIX for coupling atomistic domains to continuum domains. In this method the material properties are computed from the atomistic system via the Cauchy-Born method [129]. Since all the mentioned semi-concurrent methods are at the material point level, specific input are supplied in the material definition. In case of using LAMMPS and the Cauchy-Born rule, the commands for the fine-scale model can also be passed to the material model with the "str_command" of the material model (listing 4). The particular material model will then treat the input commands as intended in its implementation.

Listing 4: A sample input for a multiscale material in semi-concurrent PERMIX-LAMMPS coupling

```
*MATERIAL
```

[1]The user subroutines for PERMIX-ABAQUS coupling can be downloaded from http://code.google.com/p/permix/

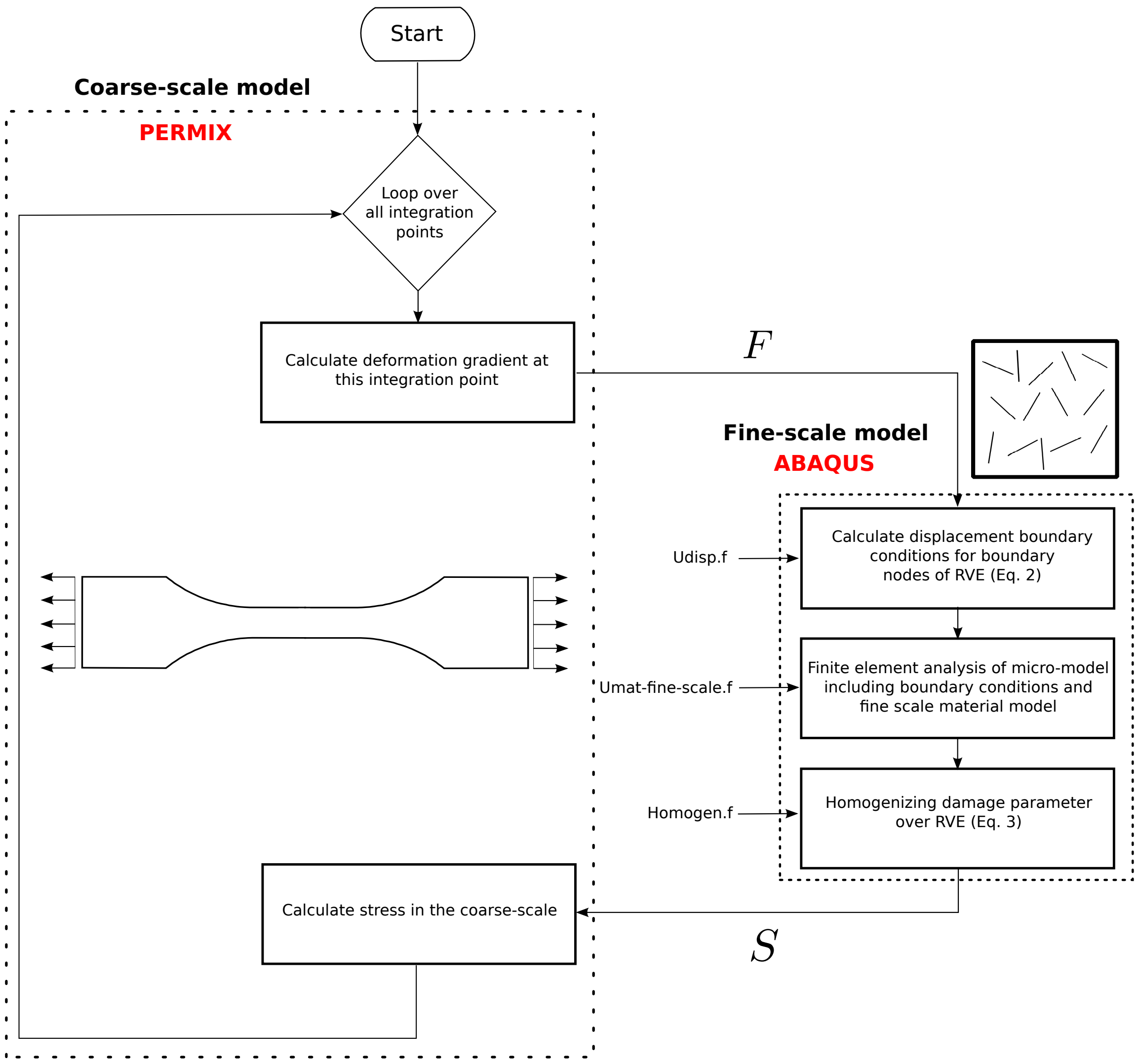
Start
Coarse-scale model
PERMIX
Loop over all integration points
Calculate deformation gradient at this integration point
F
Fine-scale model
ABAQUS
Udisp.f
Calculate displacement boundary conditions for boundary nodes of RVE (Eq. 2)
Umat-fine-scale.f
Finite element analysis of micro-model including boundary conditions and fine scale material model
Homogen.f
Homogenizing damage parameter over RVE (Eq. 3)
Calculate stress in the coarse-scale
S

Figure 5: The multiscale flowchart for coupling PERMIX to ABAQUS

```
   name   mat01
   type    CB_LAMMPS
   str_command dimension              2
   str_command boundary  s s p
   str_command units          metal
   str_command lattice           hex 2.57740
   str_command region            box block 0.0  4.0
   str_command create_box       1 box
   str_command create_atoms    1 box
*END_MATERIAL
```

5.3 Concurrent Style

Since PERMIX has access to all its own vectors and matrices, `ty_part_atom` and `ty_part_lammp` via the interface, the concurrent multiscale coupling is done in a very natural manner. For the case of PERMIX-LAMMPS coupling (the bridging domain method), an input similar to Listing 5 can be defined.

Listing 5: A sample input to define an interaction for multiscale PERMIX-LAMMPS coupling

```
 *INTERACTION
   style                        bdm_adaptive
   name                         the_briding_domain_method2d
   part_fem                     Part−1−1     #refers to the defined  FE part
   part_lammps                  atom_domain #refers to the defined  LAMMPS part
   # defines the coordinates  of  the  fine−scale box
   active_atom_region        −24.0  77.0   72.0   126.0   −100.0  500.0
*END_INTERACTION
```

The input data for the LAMMPS part, however, can be totally arbitrary because atom groups must be defined for LAMMPS, to define which atoms time integration should be performed on. In the Bridging Domain Method of PERMIX, the atoms outside the fine-scale box do not participate in the time integration. Therefore, in the `setup` routine of the `ty_bdm_adaptive` class, some additional groups are defined for LAMMPS.

6 Numerical Examples

6.1 Efficiency Tests

In this section, a middle size problem is solved for linear elasticity to benchmark the efficiency of the element matrix calculations in PERMIX. The example has 5,373,122 elements and 3,438,048

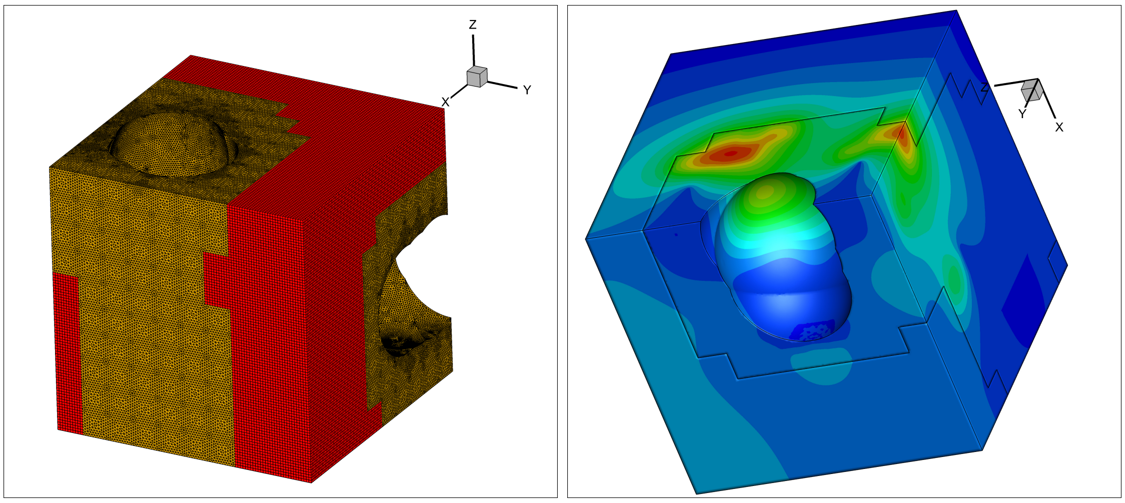

Figure 6: Left) Initial configuration of the middle sized model. Different colors refer to different element types. Right) The x-displacement contour after solution.

degrees of freedom in three dimensions and a linear elastic material model, see Fig. 6. The elements are mixed tetrahedral (four nodes, 1 integration point) and brick (eight node, 8 integration points). Some random point loads were applied to the model. The OpenMP parallelization is also tested and benchmarked.

Please note the computation time given here is with PERMIX compiled with the GCC compiler (version 4.6) which is an open source set of compilers based on the GNU licence [28]. The simulation was carried out on a numaCC system with four Six-Core AMD Opteron CPUs where the CPU frequency is 2.8GHz. All CPUs share a memory of 128GBs.

The right hand side of the Fig. 6 illustrates the displacement contour after the linear solution. Table 1 shows the results of the efficiency tests. We show only the time needed for integration and assembly of the stiffness matrix. The solution time, allocating the matrices and reading input are not reported here as they depend on the OS, solver package and pre-processor. In the table, F90 refers to the same element calculation routine which is implemented in a Fortran 90 fashion to check efficiency of the the object orientation style in PERMIX. For 1 thread and the non-optimized version, the timings have negligible difference. For certain analysis types, explicit dynamics for example, the element calculation can be further optimized by in-lining the functions called inside the main loops and also limiting the application of the element. As clear from the table, the scaling from 4 to 8 threads is not efficient due to the hardware structure of the computer.

Table 1: The elapsed time of building and assembling the stiffness matrix for different configurations

test item	elapsed time (sec)
F2003, 1 thread	60.10
F2003 , 2 thread	35.04
F2003, 4 thread	23.58
F2003, , 8 thread	22.63
F90, 1 thread	63.9
F90, 2 threads	38.2
F90, 4 threads	26.7
F90, 8 threads	24.0

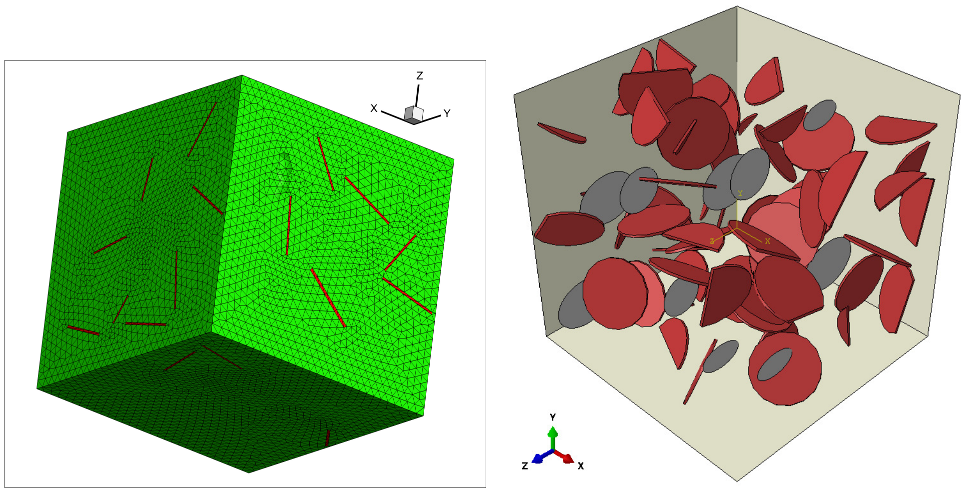

Figure 7: Three dimensional representative volume element (RVE) including randomly oriented and distributed clay particles (red coin shaped objects) and cracks (gray ellipsoids).

6.2 3D Modeling of Cracks in a Nanocomposite

In this section we model a nanocomposite with two different material types and several XFEM cracks. The model is a continuum representative volume element (RVE) of silicate/epoxy nanocomposites with 2% clay weight ratio. A random generator was used to create the coordinates of the particle corners. A condition for non-overlapping and non-intersecting particles was enforced. Fig. 7 shows the initial configuration of the representative volume elements in three dimensions. In the same model, 10 penny shape cracks are initiated in the RVE. In Fig. 7 the cracks are shown in gray without thickness and the disc shape clays are shown in red.

The model in Fig. 7 has 303,470 tetrahedral elements and 54,049 nodes. Both materials are assumed to be linear elastic where the epoxy matrix has Young's modulus of 1.96GPa and a Poisson's ratio of 0.3. The clay has a much higher stiffness with Young's modulus of 200.0GPa

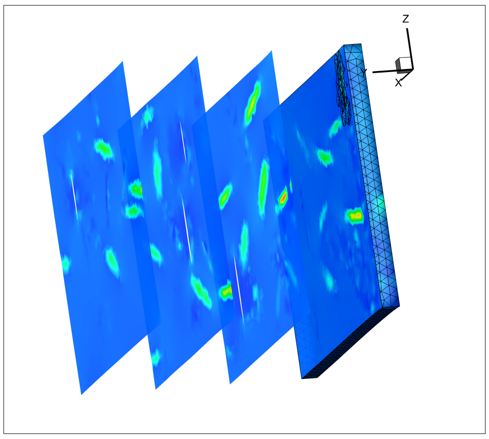

Figure 8: The σ_{xx} contour of the loaded nanocomposite RVE

and a Poisson's ratio of 0.2. The top face of the RVE is loaded with a pressure load with value of -0.1 and the bottom face is fully fixed. Fig. 8 shows the stress in the X direction in several cross-sections. The crack openings are also visible from the same figure.

6.3 Hierarchical multiscale modeling

In this case study, a hierarchical multiscale scheme is used to predict the elastic modulus of a clay/epoxy nanocomposite. Fig. 9 shows the initial geometry of epoxy/clay nanocomposite.

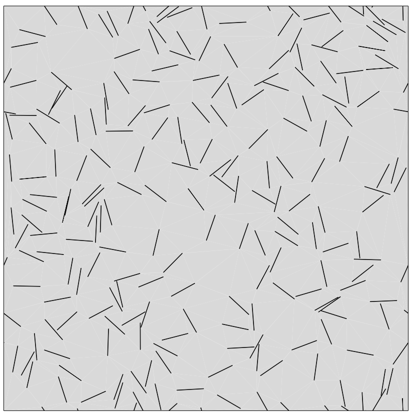

Figure 9: Initial configuration of clay/epoxy nanocomposite RVE

The nanocomposite RVE is a $2\,\%wt$ clay/epoxy nanocomposite. The clay particles are approximated with a rectangular geometry of $l = 100$ nm length and a $t = 1$ nm thickness. The RVE size is $1500 \times 1500\,\text{nm}^2$. Since the bulk composite is isotropic, the RVE size with respect to the clay size should be selected large enough that the RVE will also be isotropic.

Linear elastic material model is used for both clays and epoxy matrix. The Young's modulus,

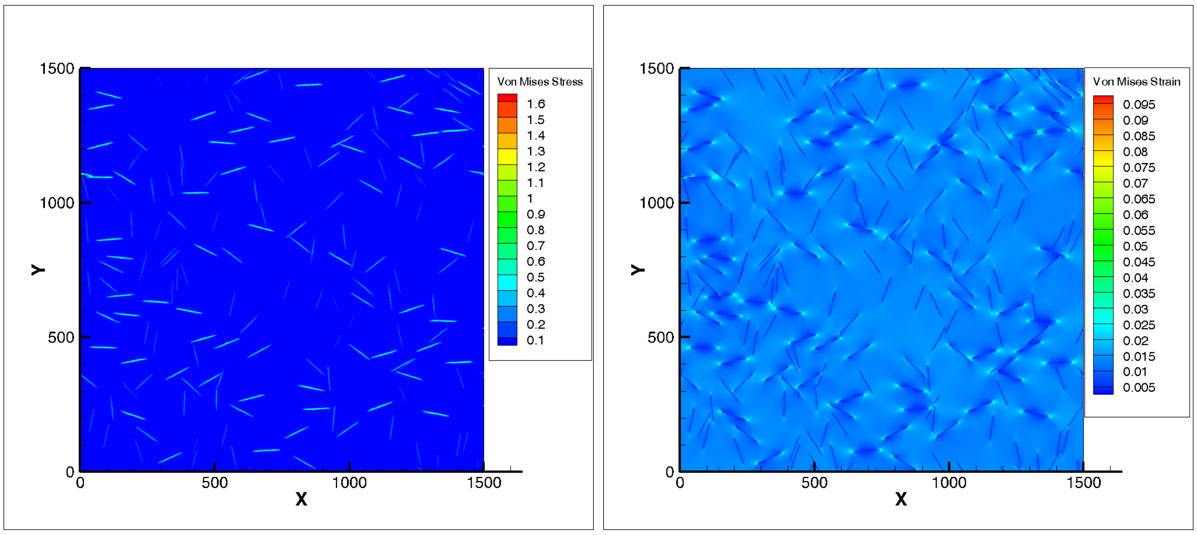

Figure 10: The Von Mises stress and Von Mises equivalent strain contours of the RVE

E, and Poisson's ratio, ν, for the clays is 196 GPa and 0.25 respectively. For the matrix the corresponding elastic values are $E = 1.96$ GPa, and the Poisson's ratio, $\nu = 0.35$. The RVE is discretized with four node quadrilateral plain stress elements. The linear displacement (LD) boundary condition was used to apply an average strain of 0.016 to the RVE in horizontal direction. This kind of boundary condition satisfy Hill's energy criterion and easy to implement.

Fig. 10 shows the Von Mises stress and Von Mises equivalent strain contours of the RVE. A homogenization routine calculates the homogenized stress and strain tensors and then, the effective properties of clay/epoxy nanocomposite can be calculated. The predicted elastic modulus and Poisson's ratio for the clay/epoxy nanocomposite is $E = 2.1725$GPa and $\nu = .343$ that means an approximate 10% of improvement in the elastic modulus of nanocomposite by using only 2% clay.

6.4 Semi-concurrent FE-FE Coupling

In this section, a semi-concurrent multiscale method, (see section 2.1.1), is applied to the clay/epoxy nanocomposite using PERMIX-PERMIX coupling. Fig. 11 illustrates the schematic view of the coarse and fine scale models.

The fine scale geometry and materials are the same as the previous example in section 6.3. For the coarse-scale model, a dog-bone sample was considered, see Fig. 11. The geometry and dimension of the dog-bone sample was selected according to ASTM D 638-08 type V.

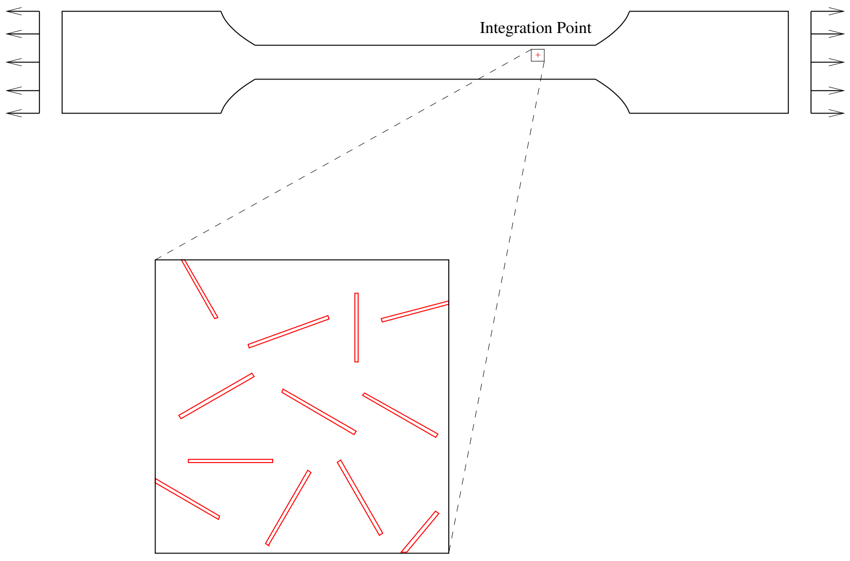

Figure 11: FE^2 multiscale analysis of simple tension test for clay nanocomposites

The numerical experiment is displacement control and due to symmetry, only a quarter of the specimen was modelled. The material parameters of the coarse scale model are computed from the fine scale model where the stress are homogenized at each integration point.

A linear displacement was applied to right edge of the dog-bone sample to produce a constant strain of 0.11 in the gage section. Fig. 12 shows the strain contour in the x direction at the coarse scale. The same figure shows the equivalent Von Mises strain contour plot at the fine scale for an arbitrary selected integration point.

6.5 Concurrent FE-XFEM Coupling

In this example we concurrently couple an FE mesh and an XFEM model using the Arlequin method in explicit dynamics in three dimensions. The example is a plate with an inclined crack at the center. We model the vicinity of the crack with a very 'fine' mesh and the area far from the crack with a 'coarse' mesh. The dimensions of the plate are $100 \times 100 \times 20$ and the length of the crack is 15. Two finite element parts are created with the same dimensions and different mesh sizes. We then use the fine mesh part to compute a reference solution and compare it to the coupled one. Both parts are discretized with eight node brick elements. Fig. 13 shows the initial mesh and the weighting parameters (Eq. 21). As can be seen from the figure, the fine mesh is placed in the middle of the plate specimen and the coupling is activated via the Arlequin interaction style in PERMIX.

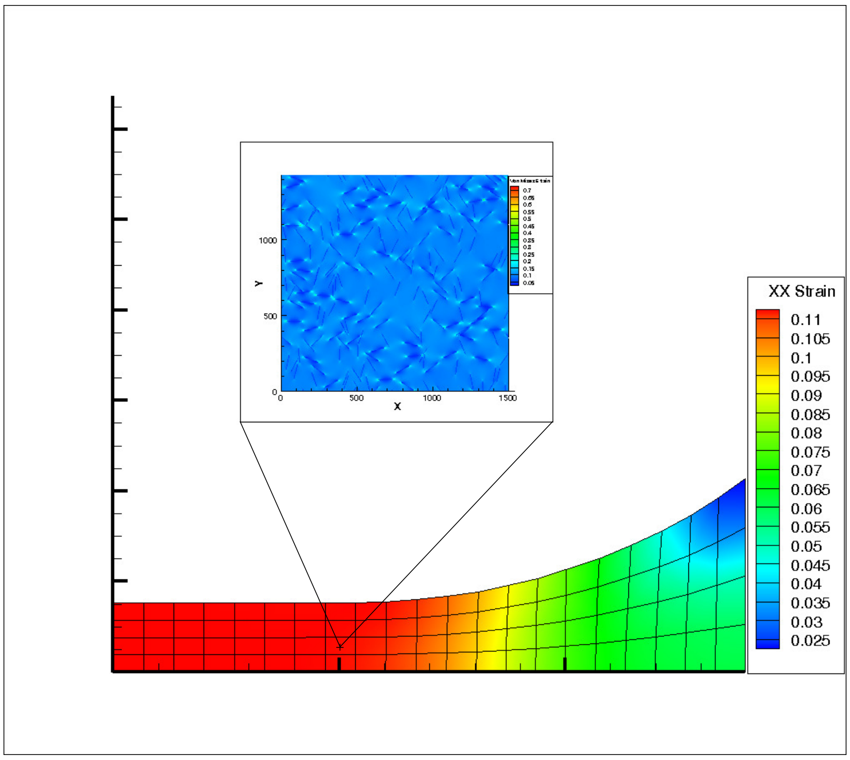

Figure 12: ϵ_{xx} contour at the coarse scale and the equivalent Von Mises strain contour at the fine scale for the FE^2 example

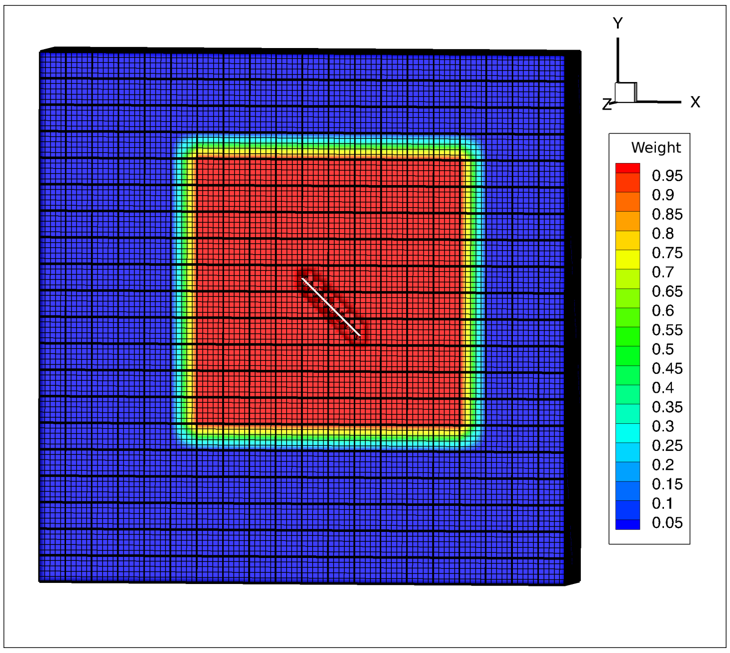

Figure 13: Initial configuration of the coupled FE-XFEM example with the weighting function values

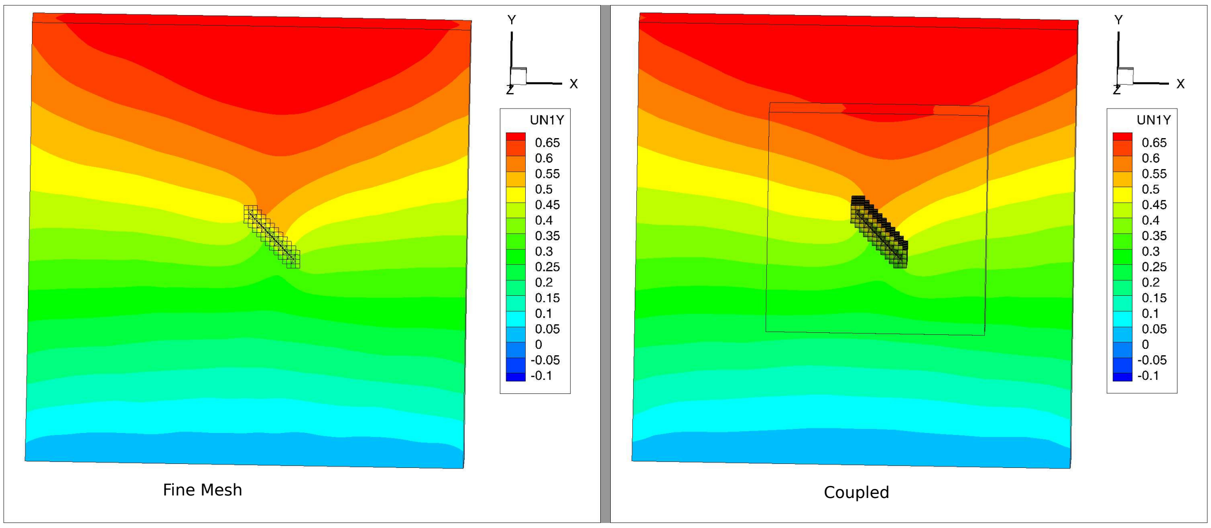

Figure 14: The Y displacement at the time 564.0 seconds, Left: reference solution (fine mesh) Right: the coupled model.

The top of the plate at the coarse-scale is loaded with a pressure load at the top and the value of the pressure is -0.25. Both the coarse and fine-scale have the same material properties with $E = 26.0$, $\nu = .3$ and density, $\rho = 1.0$. The time step, Δt, is 0.05. Stress waves travel from the coarse-scale to the coupling region and to the fine-scale and pass on to the coarse- scale again. Fig. 14 shows the result of the simulation. On the right hand side the displacement in the Y direction is shown for both scales. The left hand side of Fig. 14 shows the reference solution and the Y displacement contours. We do not observe any artificial wave reflection.

Fig. 15 shows the Y displacement at a point located at (60,60,10) for the two domains i.e. the reference solution and the coupled one. As is shown in the figure, the discrepancy in the Y displacements versus time is small and due to the displacement approximation in the fine-scale region of significantly higher resolution.

6.6 MD-XFEM Coupling

Consider a two dimensional single crystal with dimensions of $200 \times 1,00\,units$. In this example a straight crack of length $55\,units$ is assumed present in the domain. The continuum model consists of 253 quadrilateral elements and 576 degrees of freedom. The element size is constant over the domain, about $11\,units$. An atomistic domain of almost the same size is placed on top of the finite element part. Fig. 16 shows a schematic configuration of the system. The LAMMPS part is used to model the atomistic system. Since part of the crack falls within the atomistic

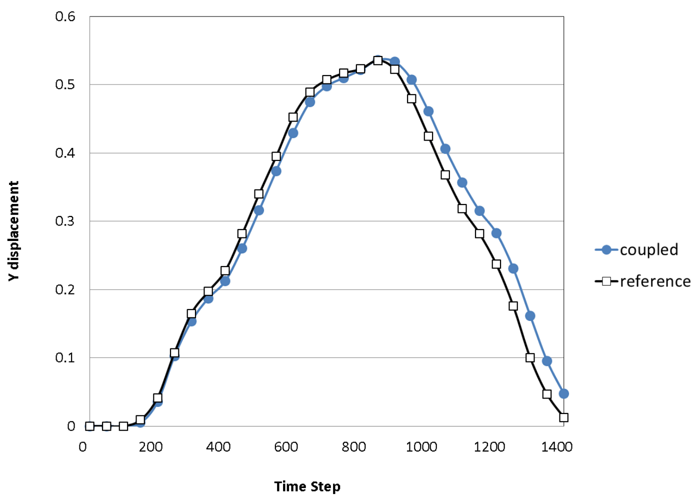

Figure 15: The Y displacement versus time for a point (60,60,10) at the coupled and the reference solution

domain, the crack must be modeled in the atomistic region as well as in the continuum region. We do not follow the generally adopted method of removing rows of atoms along the crack, as this is somewhat arbitrary and introduces extra parameters in the formulation. Instead, we modify the neighbor list of the atoms to prevent force transmission across the crack faces. This method will produce a crack which is consistent with a sharp XFEM crack. A second atomistic part is also defined in the same model which has the same configuration without coupling to any finite element mesh. This allows us to directly compare the full atomistic simulation to the coupled one.

The atomistic domain is a two dimensional lattice from a hexagonal (HEX) crystal lattice with lattice constant 0.91, $LJ\,units$ extended in the [1 0 0] crystal direction. For style LJ, all quantities are without units and LAMMPS sets the fundamental quantities mass, sigma, epsilon, and the Boltzmann constant = 1. Atomic interactions are modeled by the Lennard-Jones potential with parameters $\sigma = 1.0\,LJ\,units$, $\epsilon = 1.0$, and a cut-off radius of 2.5; the mass of all atoms is taken as 1.0. Before the actual coupled simulation starts, we minimized the potential energy in the pure atomistic part to equilibrate the system. We used the conjugate gradient (CG) algorithm [103] for the energy minimization.

The coupling of the continuum and atomistic parts is performed within a cubic box of dimensions $65 \times 110 \times 0\,LJunits^2$. The elements which are cut by this box are the bridging elements and the atoms which are located inside bridging elements are the bridging atoms.

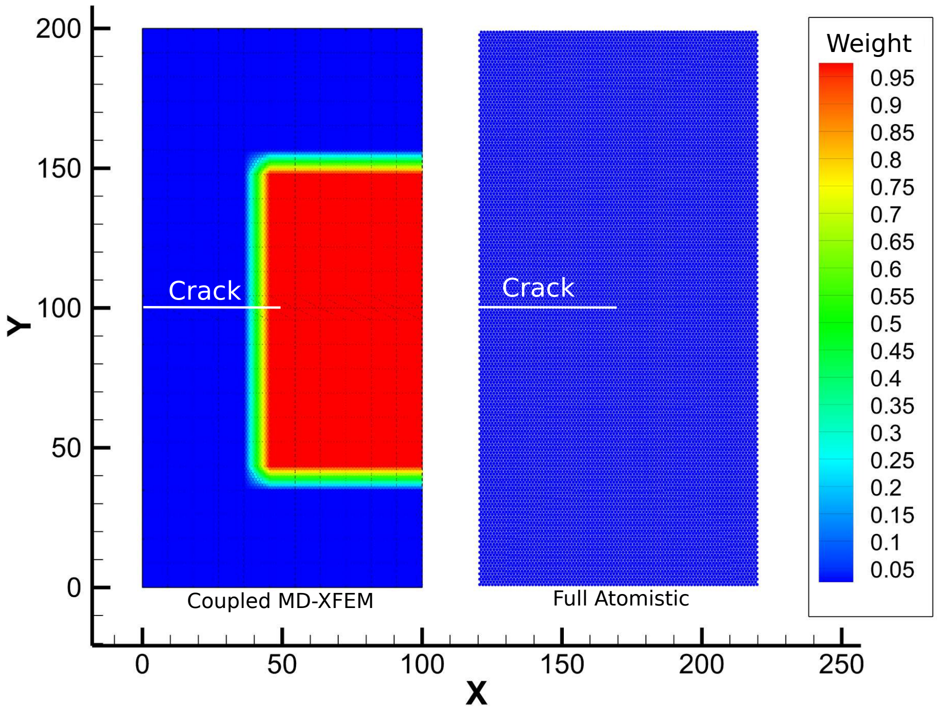

Figure 16: Initial configuration of the coupled model and the full atomistic counter part

Consequently, the coupling region is one element wide. The driving force for the system is introduced through a velocity boundary condition on the top and bottom faces of the continuum region. A velocity of 0.02 and -0.02 is set on all the nodes belonging to the top and bottom boundary of the continuum domain at each time step respectively. The time step is 0.003. The magnitude of the time step is chosen according to the stability criterion in the pure atomistic domain. Fig. 16 shows the initial configuration of the body and the weighting of the atoms and nodes.

Fig. 17 shows the stress contours at four different time steps. The atomistic stress computed here is the Virial stress tensor. The symmetric Virial stress tensor is computed for each atom and for pair potentials such as the one used here is defined in [108] and [126]:

$$\sigma_{ij}^{V} = \frac{1}{V} \sum_{\alpha} \left[\frac{1}{2} \sum_{\beta=1}^{N} \left(R_i^{\beta} - R_i^{\alpha} \right) F_j^{\alpha\beta} - m^{\alpha} v_i^{\alpha} v_j^{\alpha} \right] \tag{38}$$

where (i, j) range over the spatial directions, x, y, z. $\beta \in [1, \ldots, N]$ ranges over the N neighbors of atom α, R_i^{α} is the coordinate of atom α in the i direction, $F_j^{\alpha\beta}$ is the force on atom α from atom β along the j direction, V is the total volume, m^{α} is the mass of atom α and v^{α} is the velocity of atom α. In Fig. 17(a) and (b) a stress concentration is visible, that is initially confined at the crack front; subsequently when propagation occurs, stress waves are emitted from the crack tip. As is clear from Fig. 17 the coupled model can accurately predict the crack propagation behavior with much fewer degrees of freedom.

Remark: With the current implementation strategy in PERMIX and for concurrent multiscale methods, adaptivity can be applied with minimum effort. This is because, the data structure of different components (FE parts or atomistic parts) are separately stored and the storage strategy within each component is efficient and straight forward. For the case of MD-XFEM coupling, the atomistic region can be arbitrarily displaced, refined or coarse grained according to some measure of error. For example, in crystalline materials the centro-symmetry measure [61] can be used to detect if dislocations or cracks are close to the handshake region and therefore the continuum region is refined down to the atomistic region. The detection of the bridging elements, nodes and atoms currently is implemented for arbitrary FE meshes.

7 Conclusion

Multiscale modeling and multiscale modeling of cracks in particular, are not only difficult in theory but also they demand a huge effort on the implementation. In fact, many of the theoretical methods are hindered by implementation aspects and challenges. Therefore in this manuscript we presented an open source tool that is based on the Fortran 2003 standard. The software tool allows for multiscale modeling of cracks via several multiscale methods. The generality of the design allows several different physical systems to co-exist in the same simulation. We also explained the overall object oriented design of the code which is clearly a novel usage of the Fortran language for the task. We also showed how efficient software interfaces can be generated for coupling various software solutions written in different programming languages. The efficiency and the applicability of the software tool was shown for several multiscale fracture examples, including hierarchical, semi-concurrent and concurrent multiscale methods. The software can be used also for atomistic-continuum coupling with either concurrent or semi-concurrent methods which was shown in the last example. We hope that this tool can remove the limits of the computational methods with the implementation issues and provide the community with reusable and extendible software.

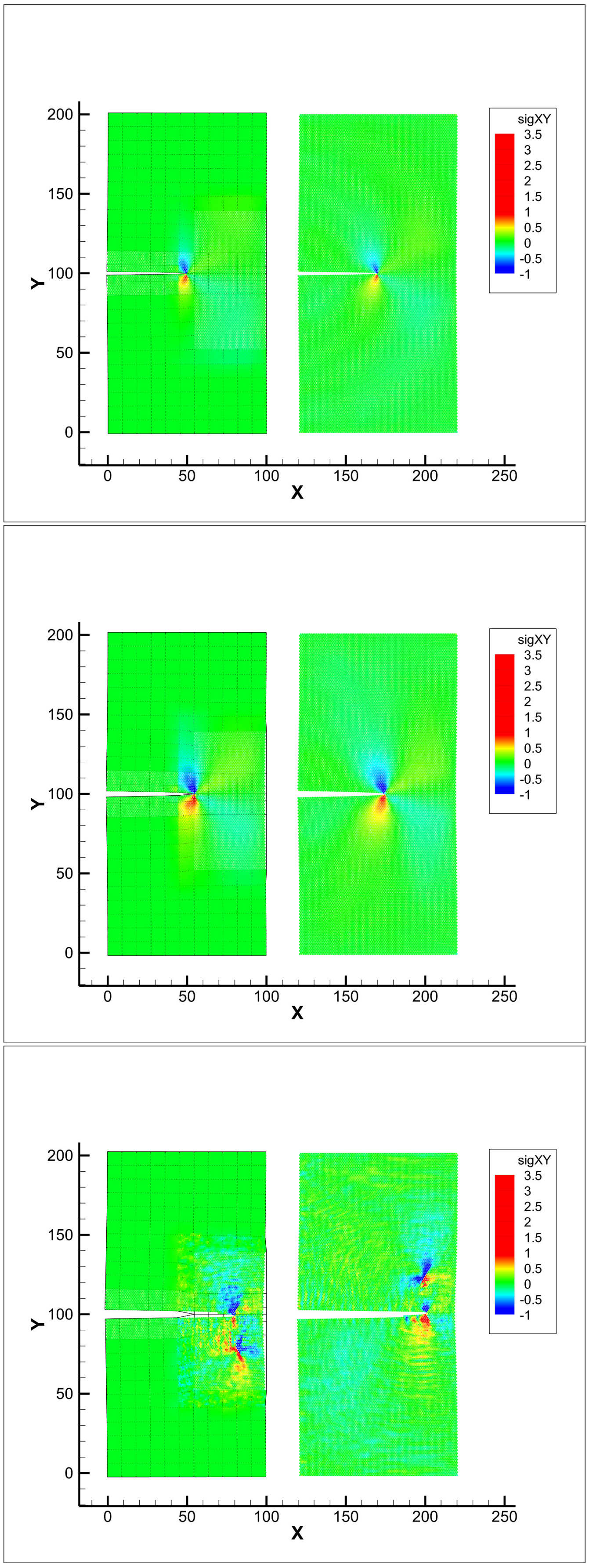

Figure 17: Atomistic σ_{xy} contour around the crack for coupled and full atomistic parts at 48 ps (top), 90 ps (middle) and 127.5 ps (bottom)

8 Acknowledgements

The authors thank the support of the German Research Foundation (DFG). Also, the authors gratefully acknowledge the help of Dr. Amitava Moitra from Pennsylvania State University for providing the WARP code. Stéphane Bordas and Pierre Kerfriden also thank partial funding for their time provided by:

- the EPSRC under grant EP/G042705/1 Increased Reliability for Industrially Relevant Automatic Crack Growth Simulation with the eXtended Finite Element Method.
- the European Research Council Starting Independent Research Grant (ERC Stg grant agreement No. 279578) entitled "RealTCut – Towards real time multiscale simulation of cutting in non-linear materials with applications to surgical simulation and computer guided surgery."

References

[1] *A micromechanical damage model for effective elastoplastic behavior of partially debonded ductile matrix composites*, International Journal of Solids and Structures **38** (2001), 6307 – 6332.

[2] *Abaqus 6.11 standard user's manual*, DASSAULT SYSTEMES, 2011.

[3] James Ahrens, Berk Geveci, and Charles Law, *Paraview: An end-user tool for large data visualization*, The Visualization Handbook **717** (2005), 731.

[4] JE Akin, *Object oriented programming via fortran 90*, Engineering Computations: Int J for Computer-Aided Engineering **16** (1999), no. 1, 26–48.

[5] B.J. Alder and TE Wainwright, *Studies in molecular dynamics. i. general method*, The Journal of Chemical Physics **31** (1959), 459.

[6] M.V. Cid Alfaro, A.S.J. Suiker, C.V. Verhoosel, and R. de Borst, *Numerical homogenization of cracking processes in thin fibre-epoxy layers*, European Journal of Mechanics - A/Solids **29** (2010), no. 2, 119 – 131.

[7] M.P. Allen and D.J. Tildesley, *Computer simulation of liquids*, vol. 18, Oxford university press, 1989.

[8] Patrick Amestoy, Iain Duff, Jean-Yves LExcellent, and Jacko Koster, *Mumps: a general purpose distributed memory sparse solver*, Applied Parallel Computing. New Paradigms for HPC in Industry and Academia (2001), 121–130.

[9] P. Aubertin, J. Réthoré, and R. de Borst, *A coupled molecular dynamics and extended finite element method for dynamic crack propagation*, International Journal for Numerical Methods in Engineering **81** (2010), no. 1, 72–88.

[10] Pascal Aubertin, Julien Rthor, and Ren de Borst, *Energy conservation of atomistic/continuum coupling*, International Journal for Numerical Methods in Engineering **78** (2009), no. 11, 1365–1386.

[11] F. Aurenhammer, *Voronoi diagramsa survey of a fundamental geometric data structure*, ACM Computing Surveys (CSUR) **23** (1991), no. 3, 345–405.

[12] Santiago Badia, Michael Parks, Pavel Bochev, Max Gunzburger, and Richard Lehoucq, *On atomistic-to-continuum coupling by blending*, Multiscale Modeling & Simulation **7** (2008), no. 1, 381–406.

[13] W. Bangerth, R. Hartmann, and G. Kanschat, *deal. iia general-purpose object-oriented finite element library*, ACM Transactions on Mathematical Software (TOMS) **33** (2007), no. 4, 24–es.

[14] Zdenek P Bazant, *Can multiscale-multiphysics methods predict softening damage and structural failure*, International Journal for Multiscale Computational Engineering **8** (2010), no. 1, 61–67.

[15] David M Beazley et al., *Swig: An easy to use tool for integrating scripting languages with c and c++*, Proceedings of the 4th USENIX Tcl/Tk workshop, 1996, pp. 129–139.

[16] W. Bellevue, *Tecplot user's manual*, 2005.

[17] T. Belytschko and T. Black, *Elastic crack growth in finite elements with minimal remeshing*, International Journal for Numerical Methods in Engineering **45** (1999), no. 5, 601–620.

[18] T. Belytschko, H. Chen, J. Xu, and G. Zi, *Dynamic crack propagation based on loss of hyperbolicity and a new discontinuous enrichment*, International Journal for Numerical Methods in Engineering **58** (2003), no. 12, 1873–1905.

[19] T. Belytschko, W. K. Liu, and B. Moran, *Nonlinear finite elements for continua and structures*, John Wiley and Sons, Chichester, 2000.

[20] T. Belytschko, Y.Y. Lu, and L. Gu, *Crack propagation by element-free galerkin methods*, Engineering Fracture Mechanics **51** (1995), no. 2, 295 – 315.

[21] Ted Belytschko, Stefan Loehnert, and Jeong-Hoon Song, *Multiscale aggregating discontinuities: a method for circumventing loss of material stability*, International Journal for Numerical Methods in Engineering **73** (2008), no. 6, 869–894.

[22] Ted Belytschko and Jeong-Hoon Song, *Coarse-graining of multiscale crack propagation*, International Journal for Numerical Methods in Engineering **81** (2010), no. 5, 537–563.

[23] Rateau G. Ben Dhia H, *The Arlequin method as a flexible engineering design tool*, International Journal for Numerical Methods in Engineering **62** (2005), 1442–1462.

[24] J. Bonet and R.D. Wood, *Nonlinear continuum mechanics for finite element analysis*, Cambridge Univ Pr, 1997.

[25] S. Bordas, V.P. Nguyen, C. Dunant, H. Nguyen-Dang, and A. Guidoum, *An extended finite element library*, **71** (2007), no. 6, 703–732, 10.1002/nme.1966.

[26] S. Bordas, T. Rabczuk, and G. Zi, *Three-dimensional crack initiation, propagation, branching and junction in non-linear materials by an extended meshfree method without asymptotic enrichment*, Engineering Fracture Mechanics **75** (2008), no. 5, 943–960.

[27] PO Bouchard, F. Bay, and Y. Chastel, *Numerical modelling of crack propagation: automatic remeshing and comparison of different criteria*, Computer methods in applied mechanics and engineering **192** (2003), no. 35, 3887–3908.

[28] I. Chivers and J. Sleightholme, *Compiler support for the fortran 2003 and 2008 standards revision 6*, ACM SIGPLAN Fortran Forum, vol. 29, ACM, 2010, pp. 26–34.

[29] T. Christman, A. Needleman, and S. Suresh, *An experimental and numerical study of deformation in metal-ceramic composites*, Acta Metallurgica **37** (1989), no. 11, 3029 – 3050.

[30] MA Crisfield, *An arc-length method including line searches and accelerations*, International journal for numerical methods in engineering **19** (1983), no. 9, 1269–1289.

[31] L. Dagum and R. Menon, *Openmp: an industry standard api for shared-memory programming*, Computational Science & Engineering, IEEE **5** (1998), no. 1, 46–55.

[32] C. Dascalu, G. Bilbie, and E.K. Agiasofitou, *Damage and size effects in elastic solids: A homogenization approach*, International Journal of Solids and Structures **45** (2008), no. 2, 409 – 430.

[33] Dibyendu K Datta, Catalin Picu, and Mark S Shephard, *Composite grid atomistic continuum method: an adaptive approach to bridge continuum with atomistic analysis*, International Journal for Multiscale Computational Engineering **2** (2004), no. 3.

[34] V.K. Decyk, C.D. Norton, and B.K. Szymanski, *Expressing object-oriented concepts in fortran 90*, ACM SIGPLAN Fortran Forum, vol. 16, ACM, 1997, pp. 13–18.

[35] S Eckardt and C Könke, *Adaptive damage simulation of concrete using heterogeneous multiscale models*, Journal of Algorithms & Computational Technology **2** (2008), no. 2, 275–297.

[36] Thomas GW Epperly, Gary Kumfert, Tamara Dahlgren, Dietmar Ebner, Jim Leek, Adrian Prantl, and Scott Kohn, *High-performance language interoperability for scientific computing through babel*, International Journal of High Performance Computing Applications **26** (2012), no. 3, 260–274.

[37] F. Feyel and J-L Chaboche, *FE^2 multiscale approach for modeling the elastoviscoplastic behavior of long fiber SiC/Ti composite materials*, Computer Methods in Applied Mechanics and Engineering **183** (2000), 309–330.

[38] J. Fish and Z. Yuan, *Multiscale enrichment based on partition of unity*, International Journal for Numerical Methods in Engineering **62** (2005), no. 10, 1341–1359.

[39] 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** (2007), no. 45-48, 4548 – 4560.

[40] Jacob Fish, Kamlun Shek, Muralidharan Pandheeradi, and Mark S. Shephard, *Computational plasticity for composite structures based on mathematical homogenization: Theory and practice*, Computer Methods in Applied Mechanics and Engineering **148** (1997), no. 1-2, 53 – 73.

[41] Jacob Fish, Qing Yu, and Kamlun Shek, *Computational damage mechanics for composite materials based on mathematical homogenization*, International Journal for Numerical Methods in Engineering **45** (1999), no. 11, 1657–1679.

[42] Bruce WR Forde and Siegfried F Stiemer, *Improved arc length orthogonality methods for nonlinear finite element analysis*, Computers & structures **27** (1987), no. 5, 625–630.

[43] C. Geuzaine and J.F. Remacle, *Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities*, International Journal for Numerical Methods in Engineering **79** (2009), no. 11, 1309–1331.

[44] Somnath Ghosh, Kyunghoon Lee, and Suresh Moorthy, *Two scale analysis of heterogeneous elastic-plastic materials with asymptotic homogenization and voronoi cell finite element model*, Computer Methods in Applied Mechanics and Engineering **132** (1996), no. 1-2, 63 – 116.

[45] Somnath Ghosh, Kyunghoon Lee, and Prasanna Raghavan, *A multi-level computational model for multi-scale damage analysis in composite and porous materials*, International Journal of Solids and Structures **38** (2001), no. 14, 2335 – 2385.

[46] I.M. Gitman, H. Askes, and L.J. Sluys, *Representative volume: Existence and size determination*, Engineering Fracture Mechanics **74** (2007), no. 16, 2518 – 2534.

[47] I.M. Gitman, H. Askes, and L.J. Sluys, *Coupled-volume multi-scale modelling of quasi-brittle material*, European Journal of Mechanics - A/Solids **27** (2008), no. 3, 302 – 327.

[48] R. Gracie and T. Belytschko, *Concurrently coupled atomistic and xfem models for dislocations and cracks*, International Journal for Numerical Methods in Engineering **78** (2009), no. 3, 354–378.

[49] Robert Gracie and Ted Belytschko, *An adaptive concurrent multiscale method for the dynamic simulation of dislocations*, International Journal for Numerical Methods in Engineering **86** (2011), no. 4-5, 575–597.

[50] JosMiranda Guedes and Noboru Kikuchi, *Preprocessing and postprocessing for materials based on the homogenization method with adaptive finite element methods*, Computer Methods in Applied Mechanics and Engineering **83** (1990), no. 2, 143 – 198.

[51] P.A. Guidault, O. Allix, L. Champaney, and J.P. Navarro, *A two-scale approach with homogenization for the computation of cracked structures*, Computers & Structures **85** (2007), no. 17-18, 1360 – 1371.

[52] Z. Hashin, *Analysis of composite materials*, J. appl. Mech **50** (1983), no. 2, 481–505.

[53] Thomas Hettich, Andrea Hund, and Ekkehard Ramm, *Modeling of failure in composites by x-fem and level sets within a multiscale framework*, Computer Methods in Applied Mechanics and Engineering **197** (2008), no. 5, 414 – 424.

[54] R. Hill, *The essential structure of constitutive laws for metal composites and polycrystals*, Journal of Mechanics Physics of Solids **15** (1967), 79–95.

[55] Claudia Britta Hirschberger, Natarajan Sukumar, and Paul Steinmann, *Computational homogenization of material layers with micromorphic mesostructure*, Philosophical Magazine **88** (2008), no. 30-32, 3603–3631.

[56] W.G. Hoover, *Smooth particle applied mechanics: the state of the art*, World Scientific Publishing Co., Inc., 2006.

[57] MF Horstemeyer, *Multiscale modeling: A review*, Practical Aspects of Computational Chemistry (2010), 87–135.

[58] Adnan Ibrahimbegovi and Damijan Markovi, *Strong coupling methods in multi-phase and multi-scale modeling of inelastic behavior of heterogeneous structures*, Computer Methods in Applied Mechanics and Engineering **192** (2003), no. 28-30, 3089 – 3107.

[59] Intel, *Math kernel library*, `http://developer.intel.com/software/products/mkl/`.

[60] Jayesh R Jain and Somnath Ghosh, *Damage evolution in composites with a homogenization-based continuum damage mechanics model*, International Journal of Damage Mechanics **18** (2009), no. 6, 533–568.

[61] C.L. Kelchner, SJ Plimpton, and JC Hamilton, *Dislocation nucleation and defect structure during surface indentation*, Physical Review B **58** (1998), no. 17, 11085.

[62] S. Kohlhoff, P. Gumbsch, and H. F. Fischmeister, *Crack propagation in b.c.c. crystals studied with a combined finite-element and atomistic model*, Philos. Mag.˜A **64** (1991), 851–878.

[63] V. Kouznetsova, *Computational homogenization for the multi-scale analysis of multi-phase materials*, PhD Thesis, Netherlands Institute for Metals Research, The Netherlands (2002).

[64] V Kouznetsova, WAM Brekelmans, and FPT Baaijens, *An approach to micro-macro modeling of heterogeneous materials*, Computational Mechanics **27** (2001), no. 1, 37–48.

[65] VG Kouznetsova, MGD Geers, and WAM Brekelmans, *Multi-scale second-order computational homogenization of multi-phase materials: a nested finite element solution strategy*, Computer Methods in Applied Mechanics and Engineering **193** (2004), no. 48, 5525–5550.

[66] Mohan G Kulkarni, Philippe H Geubelle, and Karel Matouš, *Multi-scale modeling of heterogeneous adhesives: Effect of particle decohesion*, Mechanics of Materials **41** (2009), no. 5, 573–583.

[67] Fredrik Larsson and Kenneth Runesson, *On two-scale adaptive fe analysis of micro-heterogeneous media with seamless scale-bridging*, Computer Methods in Applied Mechanics and Engineering **200** (2011), no. 37-40, 2662 – 2674.

[68] Fredrik Larsson, Kenneth Runesson, and Fang Su, *Variationally consistent computational homogenization of transient heat flow*, International Journal for Numerical Methods in Engineering **81** (2010), no. 13, 1659–1686.

[69] Jean Lemaitre, Rodrigue Desmorat, and Maxime Sauzay, *Anisotropic damage law of evolution*, European Journal of Mechanics-A/Solids **19** (2000), no. 2, 187–208.

[70] F. Lene and D. Leguillon, *Homogenized constitutive law for a partially cohesive composite material*, International Journal of Solids and Structures **18** (1982), no. 5, 443 – 458.

[71] Jae Hyuk Lim, Dongwoo Sohn, Jun Ho Lee, and Seyoung Im, *Variable-node finite elements with smoothed integration techniques and their applications for multiscale mechanics problems*, Computers & Structures **88** (2010), no. 7-8, 413 – 425.

[72] G.R. Liu and MB Liu, *Smoothed particle hydrodynamics: a meshfree particle method*, World Scientific Pub Co Inc, 2003.

[73] O. Lloberas-Valls, D.J. Rixen, A. Simone, and L.J. Sluys, *Multiscale domain decomposition analysis of quasi-brittle heterogeneous materials*, International Journal for Numerical Methods in Engineering **89** (2012), no. 11, 1337–1366.

[74] B. Q. Luan, S. Hyun, J. F. Molinari, N. Bernstein, and Mark O. Robbins, *Multiscale modeling of two-dimensional contacts*, Phys. Rev. E **74** (2006), 046710.

[75] A. Markus, *Design patterns and fortran 2003*, ACM SIGPLAN Fortran Forum, vol. 27, ACM, 2008, pp. 2–15.

[76] T. J. Massart, R. H. J. Peerlings, and M. G. D. Geers, *An enhanced multi-scale approach for masonry wall computations with localization of damage*, International Journal for Numerical Methods in Engineering **69** (2007), no. 5, 1022–1059.

[77] TJ Massart, RHJ Peerlings, and MGD Geers, *Structural damage analysis of masonry walls using computational homogenization*, International Journal of Damage Mechanics **16** (2007), no. 2, 199–226.

[78] Karel Matou, Mohan G. Kulkarni, and Philippe H. Geubelle, *Multiscale cohesive failure modeling of heterogeneous adhesives*, Journal of the Mechanics and Physics of Solids **56** (2008), no. 4, 1511 – 1533.

[79] Ted Belytschko Mei Xu, *Conservation properties of the bridging domain method for coupled molecular/continuum dynamics*, International Journal for Numerical Methods in Engineering **76** (2008), 278–294.

[80] T. Menouillard, J. Réthoré, A. Combescure, and H. Bung, *Efficient explicit time stepping for the extended finite element method (x-fem)*, International Journal for Numerical Methods in Engineering **68** (2006), no. 9, 911–939.

[81] Sinisa Dj Mesarovic and Jagan Padbidri, *Minimal kinematic boundary conditions for simulations of disordered microstructures*, Philosophical Magazine **85** (2005), no. 1, 65–78.

[82] M. Metcalf, J.K. Reid, and M. Cohen, *Fortran 95/2003 explained*, vol. 416, Oxford University Press New York, 2004.

[83] Christian Miehe, Jrg Schrder, and Jan Schotte, *Computational homogenization analysis in finite plasticity simulation of texture development in polycrystalline materials*, Computer Methods in Applied Mechanics and Engineering **171** (1999), no. 3-4, 387 – 418.

[84] R.E. Miller and EB Tadmor, *A unified framework and performance benchmark of fourteen multiscale atomistic/continuum coupling methods*, Modelling and Simulation in Materials Science and Engineering **17** (2009), 053001.

[85] N. Moes, J. Dolbow, and T. Belytschko, *A finite element method for crack growth without remeshing*, International Journal for Numerical Methods in Engineering **46** (1999), no. 1, 133–150.

[86] E. Monteiro, J. Yvonnet, and Q.C. He, *Computational homogenization for nonlinear conduction in heterogeneous materials using model reduction*, Computational Materials Science **42** (2008), no. 4, 704 – 712.

[87] T. Nakamura and S. Suresh, *Effects of thermal residual stresses and fiber packing on deformation of metal-matrix composites*, Acta Metallurgica et Materialia **41** (1993), no. 6, 1665 – 1681.

[88] S. Nemat-Nasser and M. Hori, *Micromechanics: Overall properties of heterogeneous materials*, Elsevier, Amsterdam (1993).

[89] Vinh Phu Nguyen, Oriol Lloberas-Valls, Martijn Stroeven, and Lambertus Johannes Sluys, *Computational homogenization for multiscale crack modeling. implementational and computational aspects*, International Journal for Numerical Methods in Engineering **89** (2012), no. 2, 192–226.

[90] Vinh Phu Nguyen, Oriol Lloberas-Valls, Martijn Stroeven, and Lambertus Johannes Sluys, *On the existence of representative volumes for softening quasi-brittle materials - a failure zone averaging scheme*, Computer Methods in Applied Mechanics and Engineering **199** (2010), no. 45-48, 3028 – 3038.

[91] Vinh Phu Nguyen, Oriol Lloberas-Valls, Martijn Stroeven, and Lambertus Johannes Sluys, *Homogenization-based multiscale crack modelling: From micro-diffusive damage to macro-cracks*, Computer Methods in Applied Mechanics and Engineering **200** (2011), no. 9, 1220–1236.

[92] Vinh Phu Nguyen, Oriol Lloberas-Valls, Martijn Stroeven, and Lambertus Johannes Sluys, *Multiscale continuous and discontinuous modeling of heterogeneous materials: A review on recent developments*, Journal of Multiscale Modelling **3** (2011), no. 04, 229–270.

[93] Vinh Phu Nguyen, Martijn Stroeven, and Lambertus Johannes Sluys, *An enhanced continuous-discontinuous multiscale method for modeling mode-i cohesive failure in random heterogeneous quasi-brittle materials*, Engineering Fracture Mechanics **79** (2012), no. 0, 78 – 102.

[94] JH Nie, DA Hopkins, YT Chen, and HT Hsieh, *Development of an object-oriented finite element program with adaptive mesh refinement for multi-physics applications*, Advances in Engineering Software **41** (2010), no. 4, 569–579.

[95] C.D. Norton, V.K. Decyk, and B.K. Szymanski, *High performance object-oriented scientific programming in fortran 90*, (1997).

[96] I. Özdemir, W. A. M. Brekelmans, and M. G. D. Geers, *Computational homogenization for heat conduction in heterogeneous solids*, International Journal for Numerical Methods in Engineering **73** (2008), no. 2, 185–204.

[97] I. Özdemir, W.A.M. Brekelmans, and M.G.D. Geers, *computational homogenization for the thermo-mechanical analysis of heterogeneous solids*, Computer Methods in Applied Mechanics and Engineering **198** (2008), no. 3-4, 602 – 613.

[98] M.L. Parks, R.B. Lehoucq, S.J. Plimpton, and S.A. Silling, *Implementing peridynamics within a molecular dynamics code*, Computer Physics Communications **179** (2008), no. 11, 777–783.

[99] B. Patzák and Z. Bittnar, *Design of object oriented finite element code*, Advances in Engineering Software **32** (2001), no. 10, 759–767.

[100] Heinz E. Pettermann and Subra Suresh, *A comprehensive unit cell model: a study of coupled effects in piezoelectric composites*, International Journal of Solids and Structures **37** (2000), no. 39, 5447 – 5464.

[101] S. Plimpton, *Fast parallel algorithms for short-range molecular dynamics*, Journal of Computational Physics **117** (1995), no. 1, 1–19.

[102] Steve Plimpton, *Atomistic stress simulator (warp)*, 2001.

[103] W.H. Press, *Numerical recipes in fortran: the art of scientific computing*, vol. 1, Cambridge Univ Pr, 1992.

[104] Dong Qian, Gregory J. Wagner, and Wing Kam Liu, *A multiscale projection method for the analysis of carbon nanotubes*, Computer Methods in Applied Mechanics and Engineering **193** (2004), no. 17-20, 1603 – 1632.

[105] T. Rabczuk and T. Belytschko, *Cracking particles: A simplified meshfree method for arbitrary evolving cracks*, International Journal for Numerical Methods in Engineering **61** (2004), no. 13, 2316–2343.

[106] P. Raghavan and S. Ghosh, *A continuum damage mechanics model for unidirectional composites undergoing interfacial debonding*, Mechanics of Materials **37** (2005), no. 9, 955 – 979.

[107] A. Rahman, *Correlations in the motion of atoms in liquid argon*, phys. Rev **136** (1964), no. 2A, 405–411.

[108] J.S. Robert, *Comments on virial theorems for bounded systems*, American Journal of Physics **51** (1983), 940–942.

[109] D.W.I. Rouson, J. Xia, and X. Xu, *Object construction and destruction design patterns in fortran 2003*, Procedia Computer Science **1** (2010), no. 1, 1495–1504.

[110] R.E. Rudd and J.Q. Broughton, *Concurrent coupling of length scales in solid state systems*, physica status solidi (b) **217** (2000), no. 1, 251–291.

[111] Olaf Schenk, Klaus Gärtner, Wolfgang Fichtner, and Andreas Stricker, *Pardiso: a high-performance serial and parallel sparse linear solver in semiconductor device simulation*, Future Generation Computer Systems **18** (2001), no. 1, 69–78.

[112] Jörg Schröder and Marc-André Keip, *A framework for the two-scale homogenization of electro-mechanically coupled boundary value problems*, Computer Methods in Mechanics (2010), 311–329.

[113] TANG Shao-Qiang, Wing K Liu, Eduard G Karpov, and Thomas Y Hou, *Bridging atomistic/continuum scales in solids with moving dislocations*, Chinese Physics Letters **24** (2007), no. 1, 161.

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

[115] MS Shephard, MA Nuggehally, B Franz Dale, CR Picu, J Fish, O Klaas, and MW Beall, *Component software for multiscale simulation*, Multiscale methods: bridging the scales in science and engineering (J. Fish, ed.), Oxford University Press, USA, 2009.

[116] J. Shewchuk, *Triangle: Engineering a 2d quality mesh generator and delaunay triangulator*, Applied Computational Geometry Towards Geometric Engineering (1996), 203–222.

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

[118] L.E. Shilkrot, Ronald E. Miller, and William A. Curtin, *Multiscale plasticity modeling: coupled atomistics and discrete dislocation mechanics*, Journal of the Mechanics and Physics of Solids **52** (2004), no. 4, 755 – 787.

[119] H. Si, *Tetgen: A quality tetrahedral mesh generator and three-dimensional delaunay triangulator, research group: Numerical mathematics and scientific computing*, Weierstrass Institute for Applied Analysis and Stochastics (http://tetgen. berlios. de/) (2007).

[120] S.A. Silling, *Reformulation of elasticity theory for discontinuities and long-range forces*, Journal of the Mechanics and Physics of Solids **48** (2000), no. 1, 175–209.

[121] S.A. Silling, M. Epton, O. Weckner, J. Xu, and E. Askari, *Peridynamic states and constitutive modeling*, Journal of Elasticity **88** (2007), no. 2, 151–184.

[122] R.J.M. Smit, W.A.M. Brekelmans, and H.E.H. Meijer, *Prediction of the mechanical behavior of nonlinear heterogeneous systems by multi-level finite element modeling*, Computer Methods in Applied Mechanics and Engineering **155** (1998), no. 1-2, 181 – 192.

[123] Flavio V. Souza and David H. Allen, *Multiscale modeling of impact on heterogeneous viscoelastic solids containing evolving microcracks*, International Journal for Numerical Methods in Engineering **82** (2010), no. 4, 464–504.

[124] Flavio V. Souza and David H. Allen, *Modeling the transition of microcracks into macrocracks in heterogeneous viscoelastic media using a two-way coupled multiscale model*, International Journal of Solids and Structures **48** (2011), no. 22-23, 3160 – 3175.

[125] T. Strouboulis, K. Copps, and I. Babuška, *The generalized finite element method: An example of its implementation and illustration of its performance*, International Journal for Numerical Methods in Engineering **47** (2000), no. 8, 1401–1417.

[126] A.K. Subramaniyan and CT Sun, *Continuum interpretation of virial stress in molecular simulations*, International Journal of Solids and Structures **45** (2008), no. 14-15, 4340–4346.

[127] W.C. Swope, H.C. Andersen, P.H. Berens, and K.R. Wilson, *A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters*, The Journal of Chemical Physics **76** (1982), 637.

[128] S.P. Xiao T. Belytschko, *Coupling methods for continuum model with molecular model*, Internation Journal for Multiscale Computational Engineering **1** (2003), 115–126.

[129] Ellad B Tadmor, Michael Ortiz, and Rob Phillips, *Quasicontinuum analysis of defects in solids*, Philosophical Magazine A **73** (1996), no. 6, 1529–1563.

[130] H. Talebi, C. Samaniego, E. Samaniego, and T. Rabczuk, *On the numerical stability and mass-lumping schemes for explicit enriched meshfree methods*, International Journal for Numerical Methods in Engineering **89** (2012), no. 8, 1009–1027.

[131] H. Talebi, G. Zi, M. Silani, E. Samaniego, and T. Rabczuk, *A simple circular cell method for multilevel finite element analysis*, Journal of Applied Mathematics **2012** (2012).

[132] Ä. Temizer and P. Wriggers, *An adaptive multiscale resolution strategy for the finite deformation analysis of microheterogeneous structures*, Computer Methods in Applied Mechanics and Engineering **200** (2011), no. 37-40, 2639 – 2661.

[133] O. van der Sluis, P.J.G. Schreurs, W.A.M. Brekelmans, and H.E.H. Meijer, *Overall behaviour of heterogeneous elastoviscoplastic materials: effect of microstructural modelling*, Mechanics of Materials **32** (2000), no. 8, 449 – 462.

[134] Clemens V. Verhoosel, Joris J. C. Remmers, Miguel A. Gutirrez, and Ren de Borst, *Computational homogenization for adhesive and cohesive failure in quasi-brittle solids*, International Journal for Numerical Methods in Engineering **83** (2010), no. 8-9, 1155–1179.

[135] Gregory J. Wagner and Wing Kam Liu, *Coupling of atomistic and continuum simulations using a bridging scale decomposition*, Journal of Computational Physics **190** (2003), no. 1, 249 – 274.

[136] G. N. Wells and L. J. Sluys, *A new method for modelling cohesive cracks using finite elements*, International Journal for Numerical Methods in Engineering **50** (2001), no. 12, 2667–2682.

[137] SP Xiao and T. Belytschko, *A bridging domain method for coupling continua with molecular dynamics*, Computer methods in applied mechanics and engineering **193** (2004), no. 17-20, 1645–1669.

[138] X.-P. Xu and A. Needleman, *Numerical simulations of fast crack growth in brittle solids*, Journal of the Mechanics and Physics of Solids **42** (1994), 1397–1434.

[139] Zheng Yuan and Jacob Fish, *Toward realization of computational homogenization in practice*, International Journal for Numerical Methods in Engineering **73** (2008), no. 3, 361–380.

[140] G. Zi, T. Rabczuk, and W. Wall, *Extended meshfree methods without the branch enrichment for the cohesive crack model*, Computational Mechanics **40** (2007), no. 2, 367–382.