

The Computational Materials Design Facility (CMDf): A powerful framework for multi-paradigm multi-scale simulations

Markus J. Buehler^{1,2*}, Jef Dodson², Adri C.T. van Duin², Peter Meulbroek², William A. Goddard III²

¹ Massachusetts Institute of Technology, Cambridge, MA 02139, USA

² Materials and Process Simulation Center, California Institute of Technology, Pasadena, CA, 91125, CA

ABSTRACT

Predicting the properties and behavior of materials by computer simulation from a fundamental, *ab initio* perspective has long been a vision of computational material scientists. The key to achieving this goal is utilizing hierarchies of paradigms and scales that connect macrosystems to first principles quantum mechanics (QM). Here we describe a new software environment, the “Computational Materials Design Facility” (CMDf), capable of simulations of complex materials studies using a variety of simulation paradigms. The CMDf utilizes a Python scripting layer to integrate different computational tools to develop multi-scale simulation applications. We have integrated DFT QM methods, the first principles ReaxFF reactive force field, empirical all atom force fields (FFs), mesoscale and continuum methods. The central data structure Extended OpenBabel (XOB) plays a critical role as glue between applications. We demonstrate the usefulness of CMDf in examples that couple complex chemistry and mechanical properties during dynamical failure processes, as for example in a study of cracking of Ni under presence of O₂.

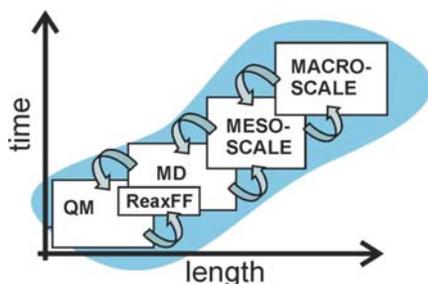


Figure 1: Schematic representation of the Computational Materials Design Facility (CMDf). To calculate materials properties, fundamental information is obtained from the QM level, and then used to train the ReaxFF level, which in turn is used to train ordinary FF and mesoscale levels to inform the macroscale simulations needed for engineering design. In principle, this scheme allows the process to be inverted. CMDf allows re-usage of software for multiscale applications. The concepts developed within the CMDf framework will help with seamlessly integrated multiscale modeling.

INTRODUCTION

Calculating macroscopic properties of materials from first principles or *ab-initio* computations is one of the foremost goals of computational materials science. There have been enormous advances in efficient and accurate quantum mechanical (QM) methods [1], new accurate force fields for molecular dynamics (MD) [2-5], coarse grain descriptions for treating very large systems, and techniques to couple scales and paradigms (e.g. [6-8]). Furthermore, coupling these advances in methods with the enormous growth of computing power has enabled studies with millions to billions of particles [9]. However, remains cumbersome to apply all these scales of simulation to the same problem in order to transcend completely from the electrons of QM to continuum plasticity, within a single model. We report development of the

* Electronic address: mbuehler@MIT.EDU

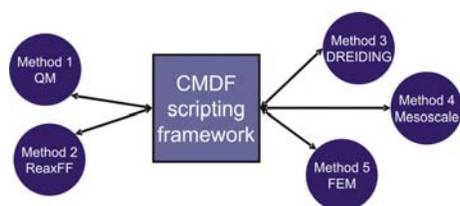


Figure 2: The CMDF framework provides the glue coupling various computational engines operating on different scales and based on a variety of simulation paradigms. The Python scripting environment enables the integration of these highly distinct scales and paradigms. All modules access the XOB data mode, which ensures that new communication channels between various modules can easily be realized. The CMDF is an ideal environment for prototyping new scale-coupling algorithms.

Computational Materials Design Facility (CMDF), a new multi-scale multi-paradigm simulation framework. The CMDF is developed to achieve seamless multi-scale modeling from electrons to continuum. The CMDF was constructed with the objective of providing an extensible and easy to use framework for multi-scale modeling and that enables simulation of complex materials to start at QM and bridge up to the continuum scale. The CMDF is based on a hierarchy of overlapping modeling methods where the parameters at each level are based on a more fundamental theory (schematic shown in Figure 1). We apply our new framework to describe fracture of nickel under presence of O₂ molecules, a problem that could not be solved with existing methods.

MOTIVATION: THE NEED FOR MULTI-PARADIGM MODELING

To obtain a fundamental understanding of macroscale behavior of materials important in engineering (and physics, biology, and chemistry), it is essential to relate it to the atomic scale interactions with macroscopic scales involving more than 10^{23} atoms. However, the relationship between the microscopic, atomic scale and macroscopic properties is generally too complicated for any sort of analytical relationship. Thus, we must develop numerical simulations capable of relating atomic-scale processes with macroscale behavior. For example, the overall macroscopic behavior of materials is governed by the response to microscopic material defects such as cracks, dislocations, or grain boundaries. Thus, whether a material exhibits brittle (like glass) or ductile (like copper) behavior depends on the nature of the interatomic interactions, which essentially depends on the chemistry of interacting atoms. In brittle failure, the stress ahead of the crack is localized causing individual bond breaking leading to cleavage of the material [10]. In ductile failure, the stress ahead of the crack causes local annealing of atomic distortions so that atomic planes to slip on top of each other leading to nucleation of dislocations along directions of largest shear. Whereas most previous studies have neglected the complexities of chemical interactions on the mechanics of materials, the CMDF framework enables such conglomerated studies, taking into consideration the competing mechanisms of shear, chemical events changing the bonding, or cleavage of atomic bonds leading to brittle fracture, while modelling the continuum constraints imposed on a moving crack. Using CMDF, it is possible to develop fundamental understanding of how reaction mechanisms at the crack tip control macroscopic behavior.

TECHNICAL SOFTWARE DESIGN OF CMDF

Various modeling paradigms are coupled via a Python scripting environment. Scripting driven frameworks have been developed and applied previously to enable complex materials simulations [11-13]. In contrast to earlier work, the goal of CMDF is to fully integrate state-of-the-art first principles QM methods, the first principles based ReaxFF [2] reactive force field

```

1   ModDynamicsCPP.IntegrateNVE1 ( OBtot, deltat, cellx,
2                                     celly, cellz, 1, 1, 1 )
3   sys = OBSystem ()
4   sys.AddMol ( OB1 )
5   sys.AddMol ( OB2 )
6   ModMultiCPP.xyzupdate ( OBtot, sys )
7
8   REAXTOOLS.REAXForces ( OB1, modreax, cellx, celly, cellz )
9   IMDTOOLS.IMDFORCES ( OB2, IMD_EAM )
10
11  sys = OBSystem ()
12  sys.AddMol ( OB1 )
13  sys.AddMol ( OB2 )
14  ModMultiCPP.forcecombine ( OBtot, sys, 1 )
15
16  ModDynamicsCPP.IntegrateNVE2 ( OBtot, deltat )

```

Figure 3: Example CMDF Python script. In lines 1, the first step of the Velocity Verlet integration is performed. In lines 3-5, a new system object is created that contains the collection of atoms OB1 and OB2 (XOB objects). This system object is used to update the new positions of atoms in the collection of atoms OB1 and OB2. The integration is always done at the system level. In lines 8 and 9, new forces are calculated with two distinct computational engines (ReaxFF and ITAP IMD/EAM). The forces calculated by these engines are then combined into the global collection of atoms OBtot in line 14 after creating a system object in lines 11-13. Finally, in line 16 the second step of the Velocity Verlet integration is performed.

method, all available non-reactive and empirical force fields, mesoscale, and continuum methods into a framework where all of these can be coupled together as suitable for a particular application, while providing all available computational tools for setting up and analyzing the results for simulations over a broad range of materials. CMDF is designed to:

- Provide a general, extensible approach of a simulation environment utilizing a library of a variety of computational tools spanning scales from quantum mechanics to continuum theories.
- Establish a re-usable library of highly complex computational tools that can be used as black boxes for most applications, while being initialized with standard parameters for easy usage in standard cases.
- Enable atomistic applications to be used by engineers and experimental scientists, while retaining the possibility of building highly complex simulations and models.
- Close the gap in coupling fundamental, quantum mechanical methods such as DFT to the ReaxFF reactive force field, to nonreactive force field descriptions (e.g. DREIDING, UFF).
- Provide a test bed for developing new model and algorithms, making it simple to develop new communication channels between computational engines (e.g. developing a new force fields combining distinct methods as QEq, Morse potentials, ReaxFF or M/EAM).

Since all simulation tools and engines can be called from a Python scripting language level, scale agnostic combinations of various modeling approaches can easily be realized (see Figure 2). This strategy [14] enables complex simulations to be simplified to a series of calls to various modules and packages, whereas communication between the packages is realized through the CMDF central data structures that are of no concern to the applications scientist. An excerpt of a CMDF script is shown in Figure 3.

Individual computational engines are wrapped using automated wrapping tools (e.g. SWIG, PyFort, F2PY or others). This strategy allows rapid integration of low-level codes with scripting languages. Scripting languages provide interfaces to compiled code, and rely significantly on external libraries for important subsystems. The CMDF builds upon this concept of wrapping and providing bindings to each code to provide the environment in which a “complete set” of highly complex, but validated simulation and modeling codes, which are embedded into a common, flexible, extensible framework that allows new applications to be developed on the fly

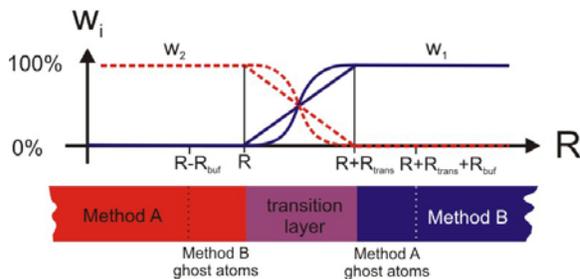


Figure 4: Schematic showing coupling of different interatomic potentials using the concept of mixed Hamiltonians (using linear or smooth interpolation schemes). The transition layer serves as the handshake region between the two distinct methods (this can be extended to describe multiple regions treated with different numerical accuracy). In the example discussed here, A=EAM and B=ReaxFF.

by an applications scientist without having to understand or deal with the complexities inherent in each of the computational engines. We find that the CMDf provides a unique solution allowing simulation applications to be focused on very challenging problems. The CMDf consists of four classes of modules and methods:

- **Class I: Data storage and central data model XOB:** Generic data model (central place of data storage, these modules involve set and get functions for inserting and applying data).
- **Class II: System wide operations related to the generic data model XOB:** Methods that operate on the generic data model (integrators and dynamics modules, I/O routines that write different file types, such as XYZ, BGF, or PDB files, pre-processing for visualization, analysis methods, as well as crystal and molecule builders). The set of methods solely operates on the data model.
- **Class III: Force and energy engines (e.g. various force fields) and related methods (e.g. charge equilibration QEq):** Methods that provide functions to calculate forces on atoms and energy of atoms, molecules or clusters. These engines all operate on XOB objects.
- **Class IV: Interfaces:** Interfaces form the critical parts of CMDf since they connect different force engines, for example using the weighting method based on the safe buffer layer concept. Interfaces play an important role in coupling methods across scales and paradigms.

The ReaxFF first principles-based reactive force field plays an integral role in CMDf, since it provides the versatility required to predict catalytic processes in complex systems nearly as accurate as QM at computational costs closer to that of simple force fields (see also Figure 1).

Duin *et al.* have demonstrated that ReaxFF reproduces QM results for both reactive and non-reactive systems, including hydrocarbons, nitramines, ceramics, metal alloys, and metal oxides (see, e.g. [2, 15, 16]). ReaxFF enables studies that lead to an improved atomic level understanding of the mechanisms of complex reactive processes critical to designing new energetic materials, catalysts for fuel cells, and nanoscale systems. Current results suggest that ReaxFF has the versatility required to simulate a wide variety of complex chemical systems, providing a bridge between various other, more classical force fields. ReaxFF parameters are determined by fitting against QM derived data. Due to the complexity of the underlying mathematical expressions, and the necessity to perform a charge equilibration (QEq) at each iteration, ReaxFF is ~10-100 times more expensive computationally than simple FFs such as DREIDING, or Tersoff. However, it is several orders of magnitude faster than QM.

APPLICATION: CONCURRENT MODELING OF OXIDATION ASSISTED FRACTURE OF NICKEL

We couple between domains described with disparate force fields using handshake regions. The role of the handshake regions is to ensure that the correct boundary conditions are applied to each side, and that each side senses proper continuation in distribution or density and forces.

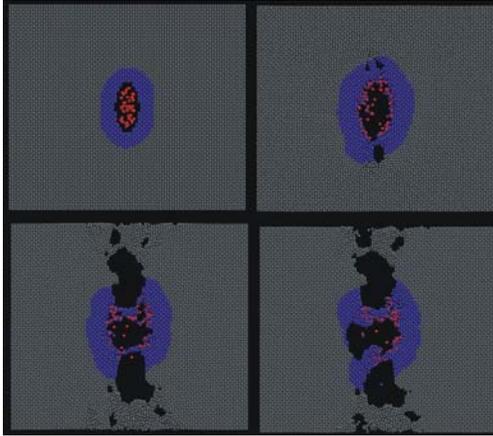


Figure 5: Nano-scale elliptical penny-shaped crack in nickel filled with O_2 . The system is under 10% strain in the x -direction (orthogonal to the long axis of the elliptical defect). Oxidative processes leading to formation of an oxide layer are competing with extension of the crack. In later stages of the simulations, the oxide layer still remains, keeping the Ni half spaces together, indicating that it involves strong Ni-O bonds. The reactive region can expand or shrink during the simulation and is determined by the positions of the oxygen atoms. Failure initiates by formation of nano-voids in the Ni bulk phase. Classical modelling schemes, for example based on the EAM method, cannot describe such organic-metallic systems.

Figure 4 illustrates the approach [5]. The transition region is described by two parameters, R_{trans} for the width of the transition region, and R_{buf} for the width of the ghost atom region. In bridging such distinct computational engines we find it useful to use spatially varying weights w_i to determine the weighting of the force and energy contribution from different simulation engines. Every computational engine i has a specific weight w_i associated with it. The N weights always add up to one, as $\sum_{i=1..N} w_i = 1$. The force on each atom is then given by the weighted sum of all force contributions of each method $F = \sum_{i=1..N} w_i F_i$. Different simulation regimes are coupled to one another by smoothly interpolating between different engines by using smoothly varying weighting functions. The width of the transition region R_{trans} depends on the nature of the system, but it should generally be larger than the typical atomic distance in a lattice or in an organic molecule. The width of the buffer layer R_{buf} describing the ghost atoms should be about 10% larger than any long range cutoffs to rule out possible boundary effects. The important point here is that the atoms at the interface to the ghost atoms (which still contribute a small amount to the total force) should not sense the existence of the boundary of the buffer layer at any point, thus $R_{\text{buf}} > R_{\text{cut}}$. The frequency at which the calculation regimes are updated (provided it is actually updated) depends on the nature of the problem.

Complicated chemical oxidative processes and their influence on the mechanical properties of materials have been a long-standing research interest. Previously, the only method to model these mechanisms was to perform QM calculations to describe formation and breaking of bonds between oxygen and the metal surface. The CMDF framework provides an alternative, new and very promising approach in describing complex interactions between chemistry and mechanical properties. The main feature of our models is that we confine accurate interactions to regions close to the crack tip and use computationally less expensive methods in regions further away from the crack tip. ReaxFF provides a suitable bridge between organic chemistry and inorganic metallic systems. Figure 5 depicts the results of a CMDF calculation for the Ni-NiO system under mechanical load. This also illustrates one of the advantages of CMDF in that it allows to easily transition to new potentials, geometries or boundary conditions. The fact that the overall flow of simulations is controlled at the Python scripting level makes it easy to add new functionality and test or develop new algorithms.

DISCUSSION AND CONCLUSION

The CMDF provides an easy and fruitful means to realize such simulations with high computational efficiency, while including the effect of chemistry including charge transfer into modeling of mechanics of materials. By composing simple scripts as shown in Figure 3, new simulation strategies or algorithms can easily be constructed, by combining many different force fields and modeling paradigms. We have recently extended this scheme to other materials and materials combinations, as done for the case of cracking of silicon [5], a H₂O-magnesium system or nano-voids of Al/Al₂O₃ interacting with water molecules. Such studies would not be possible without CMDF.

ACKNOWLEDGEMENTS

We thank Dr. Carey Schwartz and Dr. Steve Wax of DARPA for encouraging this development by funding first a special CMDF project and then the CMDF part of the DARPA-PROM project. Additional funding for this project was provided by DOEASC and NSF-ITR. Helpful discussions with Dr. Paul van Allmen (JPL), Dr. Art Voter (LANL), and members of the MSC are gratefully acknowledged.

REFERENCES

- [1] M. Springborg, Density-functional methods in chemistry and materials science, Wiley research series in Theoretical Chemistry, 1997.
- [2] A. C. T. v. Duin, S. Dasgupta, F. Lorant, W. A. Goddard, *J. Phys. Chem. A* 105 (2001) 9396-9409.
- [3] D. W. Brenner, O. A. Shenderova, J. A. Harrison, S. J. Stuart, B. Ni, S. B. Sinnott, *Journal Of Physics-Condensed Matter* 14 (2002) 783-802.
- [4] S. J. Stuart, A. B. Tutein, J. A. Harrison, *Journal Of Chemical Physics* 112 (2000) 6472-6486.
- [5] M. J. Buehler, A. C. T. v. Duin, W. A. Goddard, *Phys. Rev. Lett.*, in press.
- [6] W. K. Liu, E. G. Karpov, S. Zhang, H. S. Park, *Computer Methods in Applied Mechanics and Engineering* 193 (2004) 1529-1578.
- [7] X. T. Li, E. Weinan, *Journal Of The Mechanics And Physics Of Solids* 53 (2005) 1650-1685.
- [8] W. E. Curtin, R. E. Miller, *Model. Sim. Mat. Science and Engr.* 11 (2003) R33-E68.
- [9] F. F. Abraham, R. Walkup, H. Gao, M. Duchaineau, T. D. d. L. Rubia, M. Seager, *P. Natl. Acad. Sci. USA* 99 (2002) 5788-5792.
- [10] M. J. Buehler, H. Gao, *Nature* 439 (2006) 307-310.
- [11] J. K. Norskov, J. Schiotz, K. W. Jacobsen.
- [12] S. J. Zhou, D. M. Beazly, P. S. Lomdahl, B. L. Holian, *Phys. Rev. Lett.* 78 (1997) 479-482.
- [13] J. A. Hauch, D. Holland, M. Marder, H. L. Swinney, *Future Generation Computer Systems* 19 (2003) 599-609.
- [14] S. G. Parker, C. R. Johnson, D. Beazley, *IEEE Computational Science and Engineering* 4 (1997) 50-599.
- [15] A. Strachan, A. C. T. van Duin, D. Chakraborty, S. Dasgupta, W. A. Goddard, *Physical Review Letters* 91 (2003).
- [16] A. C. T. van Duin, K. Nielson, W. Q. Deng, J. Oxgaard, W. A. Goddard, *Abstracts Of Papers Of The American Chemical Society* 227 (2004) U1031-U1031.