Methodology Synergy in CMDF:
Toward Multi-scale Multi-paradigm Simulations

Yi Liu, Andres Jaramillo-Botero, Jef Dodson and William A. Goddard III

Materials and Process Simulation Center
California Institute of Technology
1200 E. California Blvd.
Pasadena, CA 91125

Computational Materials Design Facility (CMDF) has been developed as a general purpose materials simulation framework with capability of mixing theory and method to perform multi-scale and multi-paradigm simulations. Current CMDF integrates multiple physical methods ranging from quantum mechanics (Jaguar, SeqQuest), reactive force field (ReaxFF), non-reactive force fields (Dreiding, EAM), to continuum model (APBS). Moreover, the decoupled valence, electrostatic, and vdW energy/force functions allow one to construct Hamiltonian with customized component. All these methods compute energy and force that enable further geometry optimization (with and without constraints), or molecular dynamics (at NVE, NVT and NPT ensembles).

The nature of multi-scale and multi-paradigm modeling (e.g. spatial domain) can be described suitably by the concept of object; we introduced class into CMDF as basic building block for object orientated programming. Combining local Numpy array and central data model makes data flow more efficient than before. Hybrid coding with multiple languages (Python, Fortran and C/C++) takes advantage of the integration ability of Python while maintaining most efficiency of compiled language.

CMDF has a friendly user interface, and can be used as convenient as monolithic package. On the other hand, the encapsulated function modules in CMDF can be called and resembled easily at customized script level since they are designed to be well decoupled. CMDF is constructed to be flexible and extendable for further methodology development – an ideal platform to test new concepts and theory. CMDF can be found useful for novice, expert users and software developers.

CMDF enhances the code reuse by integrating legacy code into the framework. Once legacy codes are integrated, they can communicate and interact with other function modules through well-defined interface. By such methodology synergy, we can extend the original functionality of legacy codes, and make them more powerful than their standalone counterparts.