DIGITAL MATERIAL

MOLECULAR DYNAMICS (MD) FRAMEWORK

Design Philosophy: We develop a molecular dynamics framework with an eye for flexibility and efficiency. We design and implement our numerical kernels using C++ and control and integrate the kernels using Python. The functionalities of our framework are decomposed into 6 major base classes (see class diagram on the right). We identify class interfaces and inheritance hierarchies that are suitable for MD simulations. We develop patterns to organize the interactions among computational objects and encapsulate those aspects of a program that are likely to change.

DIGITAL MATERIAL is an extensible modeling and software infrastructure to support the representation and simulation of material structure and evolution across multiple length and time scales. Such an environment must balance the need for high performance against the need for flexible, lightweight prototyping and interoperation. It must be able to integrate a variety of programs and tools into a consistent and seamless framework for multiscale materials simulation. And it must support code reuse across scales through appropriate decomposition of functionality among collections of collaborating objects and modules.

Objectives: Digital Material is intended to serve several roles:
- A material representation
- A software environment
- A problem-solving environment
- A problem-solving environment

Strategies: The strategies we are pursuing to build this environment are several:
- The use of a two-level software architecture, combining low-level numerical kernels written in compiled languages and a high-level interpreted control and integration layer written in the interpreted programming language Python
- The use of Design Patterns and related object-oriented programming techniques to decompose simulation functionality into collaborating objects, in order to support flexible program composition
- The development of object models for the hierarchy of structures existing in materials, separated into explicitly modeled geometric components and implicitly modeled attributes, and to support the migration of information across scales.

Advantages:
- Code sharing and reuse among projects
- Flexible coupling and decoupling of computational components
- Collaboration from researchers
- Direct benefit from world-wide development of Python modules
- A set of interoperable components to create a whole that is much greater than the sum of its parts
- Software adaptivity in a responsive way to match the volatility of changing software and hardware requirements

MOLECULAR DYNAMICS (MD) FRAMEWORK

Molecular Dynamics Framework

DFT
MD
QC
Hysteresis
Phase Field
XMM
PMF

Parallelization
Visualization
Mesh
GUI


test

QUASICONTINUUM (QC) FRAMEWORK

Introduction: Quasicontinuum is a mixed atomistic-continuum approach to blend the atomistic realism with finite element interpretation. One quasicontinuum framework is a natural outgrowth of our molecular dynamics approach. Our quasicontinuum framework has gained the atomistic point of view while the finite element method has gained the continuum point of view. One key benefit of this approach is that a Quasicontinuum-based MD simulation is barely altered and extended our quasicontinuum


Error Estimation and Adaptive Meshing: To optimally represent the atomistic degrees of freedom we compute an error indicator in the form of the residual energy via a posteriori error estimation from energy and force calculations. The meshing scheme is then updated to refine the mesh size at elements identified to have critical error indicators and to coarsen the mesh where the residual energy is low.