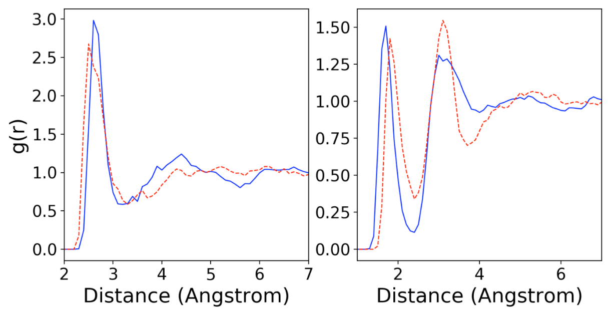
MDI

The MolSSI Driver Interface (MDI) project provides a standardized interface for fast, on-the-fly communication between computational chemistry codes. By simplifying the process of implementing and running methods that require the cooperation of multiple software packages, MDI enables researchers to use these software packages in a highly modular manner. This modularity greatly expands the options available to users when running complex simulations, such as quantum mechanics / molecular mechanics (QM/MM) multiscale techniques, *ab initio* molecular dynamics (AIMD), machine learning, advanced sampling, and path integral MD.

Overview

The MolSSI Driver Interface (MDI) project provides a standardized interface for fast, on-the-fly communication between computational chemistry codes. This greatly simplifies the process of implementing and running methods that require the cooperation of multiple software packages and enables developers to write a single implementation that works across many different codes. The interface is sufficiently general to support a wide variety of techniques, including QM/MM, AIMD, machine learning, advanced sampling, and path integral MD, while also being straightforwardly extensible.

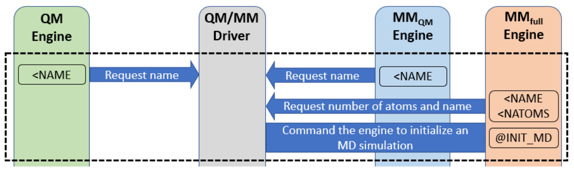
Types of Calculations



MDI is designed to support a wide variety of different calculation types, with the MDI Standard defining a variety of simple commands that can be used to develop many drivers for many purposes. Potential application areas for MDI include AIMD, path integral molecular dynamics, advanced sampling, forcefield development, and QM/MM. More specifically, here are some examples of what can be accomplished using MDI:

- Run an AIMD calculation in which a quantum mechanics (QM) code evaluates nuclear forces and a molecular mechanics (MM) code performs time integration. A proof-of-concept is available here.
- Run an MD simulation in which the nuclear forces include contributions from two different MM codes.
- Implement an advanced sampling technique (*i.e.*, metadynamics, replica exchange, etc.) in a self-contained driver that can use other codes to compute forces. An MDI-powered metadynamics example is available.
- Implement a new forcefield in a self-contained engine that is compatible with multiple MM codes.
- Perform electric field analysis of molecular dynamics trajectories. An example is available.
- Run nudged elastic band (NEB) simulations. A proof-of-concept is available.

- Run QM/MM simulations. A proof-of-concept is available.



More generally, MDI can support any scientific simulation methods that can be implemented using the command set defined by the MDI Standard. For simulation types that require additional commands, unofficial extensions to the MDI Standard can be straightforwardly developed. Feel free to contact us if your use case requires additions to the MDI Standard, and we may be able to provide advice about the process.

A rapidly growing ecosystem of codes can be used in a modular manner through MDI, including:

- CP2K
- deMonNano
- DFT-D3
- DFTB+
- entos
- FHI-aims
- LAMMPS
- Molpro
- MOPAC
- OpenMM
- PetaChem
- Psi4
- QCEngine
- QE
- RDKit
- Siesta
- TBE
- Tinker
- TorchANI
- Yaff

Approach

MDI uses a driver/engine paradigm in which drivers orchestrate complex simulations by controlling engines using an API-like command set that is defined by the MDI Standard. A driver will typically implement one or more high-level methods, such as advanced sampling or QM/MM, while relying on one or more engines to perform lower-level operations, such as energy and force evaluation. In total, MDI consists of the following components:

- Drivers, which are codes that control the high-level program flow of one or more other codes.
- Engines, which are codes capable of responding to commands from an external driver. A list of currently functional engines is available here.
- The MDI Standard, which is an API-like definition of a set of commands that can be sent from a driver to an engine, and that cause the engine to respond in a clearly defined way.
- The MDI Library, which is a library that enables inter-code communication in compliance with the MDI Standard.

More information about MDI can be found at:

https://molssi-mdi.github.io/MDI_Library/html/index.html