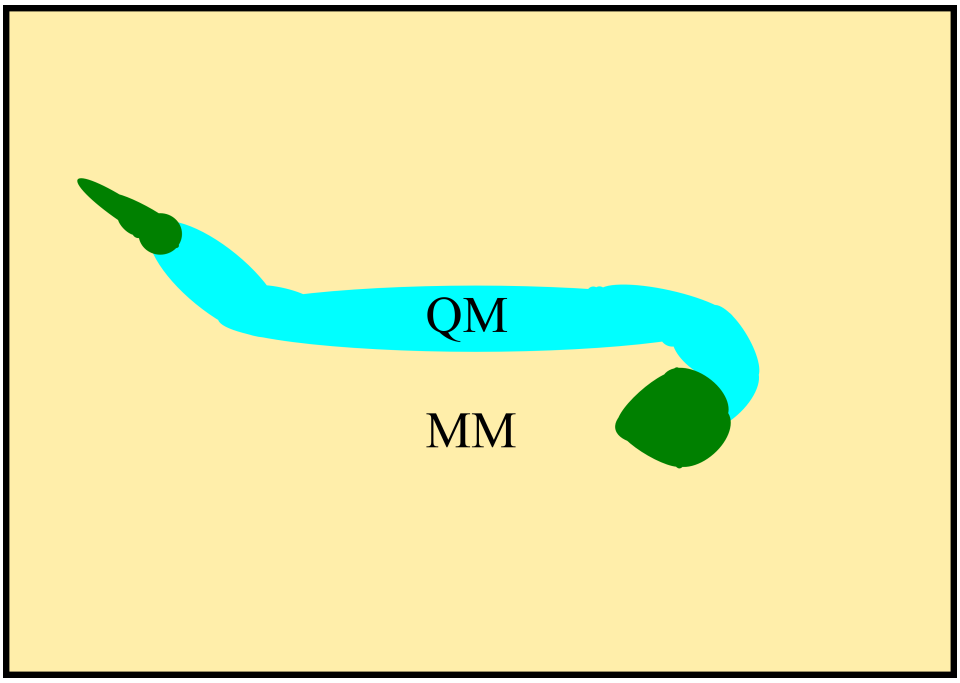
FLUKE: Fields Layered Under Kohn-Sham Electrons



A QMMM Interface for Polarizable Force Fields

Eric G. Kratz, 2014

Contents

Chapter 1

Introduction

1.1 Design Principles

FLUKE is in development with four guiding principles:

1) **FLUKE should primarily be an interface.** In order to make the code function with a variety of software packages and for different versions of the packages, FLUKE should not require modifications to other software packages or tools.
2) **FLUKE should be easy to read and modify.** Most scientists are not experts in computer programming. In order to allow for collaborative development of FLUKE, the source code needs to be easy to read. To that end, the code should be simple, well commented, and avoid linking to esoteric libraries.
3) **FLUKE should be a single executable.** Since FLUKE is an interface, there is no need for a large collection of binaries. Having many different single-use executable files can discourage new users. Furthermore, FLUKE can be statically linked to ease distribution of the software.
4) **FLUKE should be open source.** FLUKE is licensed under the GPLv3. Science can benefit everyone, and hence, should be shared as widely as possible. Additionally, I do not want to see my effort in creating a large software package go to waste in a package that can only be used by a handful of scientists.

1.2 Installation

Since FLUKE is designed to be simple, only a small number of packages are required to compile the code. An approximate list of packages is given below.

FLUKE binary: g++, OpenMP, Eigen3
FLUKE manual: latex, pdflatex, bibtex

Unpack FLUKE_tarbomb.tgz as follows.
user:$ mkdir FLUKE

user:$ cd FLUKE/
user:$ tar -xvf FLUKE_tarbomb.tgz

The FLUKE binary is included with teh source code, however, modified versions can be compiled with the script provided with FLUKE. On Ubuntu boxes the compile script should function without modifications. However, with other operating systems it may be necessary to change the path to the *Eigen3* package.
Default: -I/usr/include/eigen3/

The compile script produces both the documentation and the source code.
user:$ ./compile

1.3 Capabilities

	Gaussian	PSI4	TINKER	Amber	LAMMPS
QM energy	Yes	Yes	No	No	No
MM energy	No	No	Yes	Yes	Yes
QMMM energy	Yes	Yes	Yes	Yes	Yes
QM opt.	Yes	Yes	No	No	No
MM opt.	No	No	Yes	Yes	Yes
QMMM opt.	Yes	No	Yes	Yes	Yes
QM MC	Yes	Yes	No	No	No
MM MC	No	No	Yes	Yes	Yes
QMMM MC	Yes	Yes	Yes	Yes	Yes
QM PIMC	Yes	Yes	No	No	No
MM PIMC	No	No	Yes	Yes	Yes
QMMM PIMC	Yes	Yes	Yes	Yes	Yes
QM RP	Yes	No	No	No	No
MM RP	No	No	No	No	No
QMMM RP	Yes	No	Yes	Yes	Yes

Table 1.1: Wrapper capabilities for single-point energy, geometry optimization, Monte Carlo (MC), path-integral Monte Carlo (PIMC) and reaction path (RP) calculations.

1.3.1 QM and MM wrappers

FLUKE can perform QM, MM, or QMMM calculations via an interface to packages in the user's path. Temporary input files are created for each package and the results are collected from the temporary output files. This is a relatively inefficient procedure, however, reading and writing input/output files is often negligible compared to the computational cost of the QM calculation. Currently, calculations can be performed using Gaussian, PSI4, TINKER, Amber, and LAMMPS. Additional wrappers can be added to FLUKE with relatively little effort.

1.3.2 Types of calculations

Single-point energies, geometry optimizations, reaction pathways, classical Monte Carlo, and path-integral Monte Carlo calculations can be performed using the QM and MM wrappers. Due to limitations of the software packages, not all calculations can currently be performed with all combinations of wrappers. For convenience, the capabilities are summarized in Table 1.1.

Some additional restrictions are also present when there are bonds between the QM and MM regions. Currently only Gaussian can be used as a QM wrapper for calculations where the QM and MM regions are bonded.

1.4 Acknowledgements

The development of FLUKE was supported by funding from the NIH. FLUKE is maintained by the Cisneros research group at Wayne State University.

Chapter 2

Input

2.1 Command line arguments

FLUKE can only be invoked from a command line interface.
user:$ FLUKE -n Ncpus -x xyzfile.xyz -c confile.inp -r regfile.inp -o output.xyz

-n: Number of CPUs used in the calculations.

-x: File name for the input structure in XYZ or PDB format. The XYZ input should be in the standard format and have a blank comment line.

-c: File name for connectivity and force field information.

-r: File name for definitions of QMMM regions, QM wrapper, MM wrapper, and general simulation options.

-o: File name for trajectories and optimized structures.

2.2 XYZ input files

The input structure for FLUKE is a standard XYZ file with a blank comment line.

N

A X_A Y_A Z_A
B X_B Y_B Z_B
...

Here N is the number of atoms and (X_i,Y_i,Z_i) is the position of particle i. Note that the atom types given in the XYZ file need to be the true atom types from the periodic table.

Additionally, FLUKE reads the XYZ file item-by-item, which means that having additional values on a line will cause (silent) errors.

2.3 Connectivity input files

The connectivity of the molecules and the force field information are defined in the connectivity file. The general format is given below.

id MMTyp NumTyp q Nbonds [connectivity]
...

Here id is the index of the atom (0 to N-1), MMTyp is the force field atom type (i.e. Amber atom types), NumTyp is a numerical atom type (i.e. TINKER atom types), q is the force field charge on the atom, Nbonds is the number of bonds to the atom, and [connectivity] is a list for ids for the atoms bonded to the atom. The length of the connectivity list must match Nbonds and the information in the connectivity file must be given for all atoms, even if there are no MM atoms in the system. An additional caveat is that the atoms must be listed in order. This is part of an internal check to make sure correct input files were provided.

2.4 Region input files

The region file contains QMMM regions (QM atoms, pseudo-atoms, boundary atoms, and frozen atoms) as well as miscellaneous input needed to define the type of calculations that will be performed. This is the most complicated input file required by FLUKE and currently the input keywords must be given in the correct order. The keywords themselves are only guides for making the files human readable, only the values after the keywords are read. Blank templates of region files are provided in the documentation directory.
The first line of the input file should read:

Potential_type: <QM or MM or QMMM>

This line tells FLUKE what type of QM and MM input is listed in the lines that follow.

2.4.1 General notes

Input that is not specific to the wrappers is mostly case insensitive, however, users may find exceptions to this rule of thumb. Future versions of FLUKE will have a more robust input reader. In the mean time, please report any bugs that are found.

2.4.2 QM input

If the potential type is QM or QMMMM, several keywords are required to define the QM wrapper and level of theory.

QM_type: <Gaussian or PSI4>
QM_functional: <functional>
QM_basis: <basis set>
QM_memory: <RAM in GB>
QM_charge: <charge on the QM region>
QM_spin: <multiplicity of the QM region>

All of the QM input is read into FLUKE as text, and hence, it must match the correct input for the QM package. For example, Gaussian can only read integer values for the memory.

2.4.3 MM input

If the potential type is MM or QMMMM, several keywords are required to define the MM wrapper.

MM_type: <TINKER or Amber or LAMMPS>

All of the MM input is read into FLUKE as text, and hence, it must match the correct input for the MM package.

2.4.4 QMMM input

QMMM input is essentially just a combination of the QM and MM input described above.

QM_type: <Gaussian or PSI4>
QM_functional: <functional>
QM_basis: <basis set>
QM_memory: <RAM in GB>
QM_charge: <charge on the QM region>
QM_spin: <multiplicity of the QM region>
MM_type: <TINKER or Amber or LAMMPS>

All of the QMMM input is read into FLUKE as text, and hence, it must match the correct input for the QM and MM packages. For example, Gaussian can only read integer values for the memory.

2.4.5 Simulation input

Input for the calculation types directly follows the wrapper input. Each type of calculation requires slightly different input. FLUKE can perform single-point energies (SP), optimizations (Opt), and path-integral Monte Carlo simuations (PIMC). Classical Monte Carlo simulations are performed as PIMC simulations with a single bead. All calculation types are documented below.

Calculation_type: <SP or Energy>

Calculation_type: Opt

Calculation_type: PIMC
Ensemble: <NVT or NPT>
Temperature: <Temperature in Kelvin>
Pressure: <Pressure in atm >
Number_of_eq_steps: <Run N steps before collecting data>
Number_of_MC_steps: <Collect data for N steps>
Number_of_beads: <Number of PI beads>
Acceptance_ratio: <Fraction of attempted moves which succeed>
Traj_steps_before_printing: <Print every N steps>
Print_mode: <All or COM>

2.4.6 Region input

Four different regions of the structure need to be defined even if the regions contain zero atoms.

QM_atoms: Nqm [list of ids]
Pseudo_atoms: Npseudo [list of ids]
Boundary_atoms: Nbound [list of ids]
Frozen_atoms: Ninact [list of ids]

Definitions of the QM, pseudo-atoms, and boundary atoms can be found in Chapter 3. Frozen atoms are atoms that should remain stationary during optimizations or Monte Carlo simulations. Frozen atoms are useful optimizing small regions of large periodic systems.

Note that in pure QM and pure MM simulations, {Nqm,Npseudo,Nbound} can all be set to zero. This is because these regions are not relevant unless FLUKE is performing a QMMM calculation. For all of the regions, the lists of atoms can be separated by either spaces or newlines.

2.5 Input generation

While FLUKE does not handle atom typing or structure generation, it is possible to use the native input/output functions to create FLUKE input from input used for MM packages. In all cases the converter produces three files: xyzfile.xyz connect.inp regions.inp

TINKER: The file converter for TINKER can be used on standard TINKER XYZ files, TINKER QMMM XYZ files (Q-CHEM), and TINKER XYZ files with periodic boundary conditions.

user:$ FLUKE -convert -t tinker.xyz -k tinker.key (-p Yes)

The -p flag is optional and tells FLUKE to read the lattice constants from the second line of the XYZ file.

Amber:

LAMMPS:

2.6 FLUKE output

Most of the FLUKE output is sent through the c++ std output stream. However, trajectories and optimized structures are printed into the file defined by the -o flag. In single-point energy calculations the output XYZ file is deleted after the calculation is completed since the geometry does not change during the calculation.

2.7 Additional input for wrappers

The FLUKE wrappers automatically look for a couple "extra" input files. These files depend on which wrappers are used in the calculation, however, in most cases the files contain force field and pseudopotential information. Some of the extra files are detailed below in no particular order and examples can be found in the documentation directory.

2.7.1 TINKER

The TINKER wrapper checks the working directory for a file called tinker.key and will read additional force field information (e.g. atom classes, multipoles, etc) based on the keywords listed in that file. Any keyword defined in tinker.key will be passed into the tinker key file used by the wrapper. For somewhat obvious reasons, the user must also provide the force field parameter files defined in the tinker.key file.

2.7.2 Gaussian

The Gaussian wrapper will look for a file called BASIS in the working directory. This file contains basis and pseudopotential information for QMMM calculations with bonds between the QM and MM regions (see below). While the file is read in automatically if it is found, the user still needs to tell Gaussian to use this information:

QM_basis: GEN

GEN is a Gaussian keyword which causes the package to check for extra basis set information below the structure.

FLUKE automatically tells Gaussian to look for pseudo potentials, however, these must be given below the basis set information in the BASIS file. Both the basis set and pseudopotential information must be given in Gaussian format.

Chapter 3

Theoretical Background

3.1 Preface

The following chapter is intended to serve as a general introduction to computational chemistry and the methods used in FLUKE. There are numerous places users can find more detailed and up-to-date discussions of these topics. This chapter is only intended to give a broad review of the concepts and terminology.

3.2 Classical Methods

3.2.1 Newton's laws of motion

Much of classical physics can be described in terms of Newtons three laws of motion.

1) An object in motion tends to stay in motion unless it is acted upon by a force.
2) The force on an object is equal to its mass times its acceleration $F = ma$.
3) For every action there is an equal and opposite reaction.

Newton's second law can be employed in calculations to predict the position and velocity of objects moving in a potential field. A simplifed alogorithm for the discritized equations of motion can be written as

$$F_n = -\nabla V_n = m\ddot{r}_n \ , \tag{3.1}$$

$$\dot{r}_{n+1} = \dot{r}_n + \ddot{r}_n \Delta t \ , \tag{3.2}$$

and

$$r_{n+1} = r_n + \dot{r}_n \Delta t + \frac{\ddot{r}_n \Delta t^2}{2} \ , \tag{3.3}$$

where F is the force, V is the potential energy, r is a vector representing the potions of all atoms in the system, and Δt is the time interval between time step n and $n+1$. The natural ensemble for molecular dynamics is the mircocanonical ensemble (NVE), where the number of particles and energy are conserved inside a constant volume simulation box. Although the NVE simulations are the native result of propagating Newton's equations, most experiments

are typically performed in the canonical (NVT) or isothermal-isobaric (NPT) ensembles. In the NVT ensemble, the temperature is held constant instead of the energy, and the NPT ensemble also holds the pressure to be constant by adjusting the size of the simulation box.

3.2.2 Force fields

3.3 Quantum Methods

3.3.1 Schrödinger equation

Quantum behavior can be described using the Schrödinger wave equation and the Hamiltonian operator, $\hat{H}$,

$$\hat{H} = \hat{T} + \hat{V} \ , \tag{3.4}$$

where $\hat{T}$ is the kinetic energy operator and $\hat{V}$ is the potential energy operator. The Schrödinger equation is given by

$$\hat{H}\psi = E\psi \rightarrow E = \int \psi^* \hat{H} \psi d\tau \ , \tag{3.5}$$

where ψ is the wavefunction, E is the energy, * denotes the complex conjugate, and $\int d\tau$ is the integral over all space. The Schrödinger equation can also be written in terms of a determinant,

$$|H - ES| = 0 \ , \tag{3.6}$$

where S is the overlap matrix. Another common shorthand is bra-ket notation.

$$\hat{H}|\psi\rangle = E|\psi\rangle \rightarrow E = \langle\psi|\hat{H}|\psi\rangle \ , \tag{3.7}$$

where $\langle i|\hat{O}|j\rangle$ is the integral of operator $\hat{O}$ on state j.

3.3.2 Perturbation theory

If there is a complicated system and we want to solve the Schrödinger equation, a good starting point is often to use a system for which the exact solution is known. In chemistry, this is often the hydrogen atom and in solid state physics, it is often the "homogeneous electron gas". For example, helium can be approximated as two electrons interacting with a nucleus, but not with each other (i.e. solving the hydrogen problem twice with a nuclear charge of +2).

Mathematically, the Hamiltonian can be divided into the simple parts and the ones we wish we did not have to solve,

$$\hat{H} = \hat{H}_0 + \hat{H}_1 + \hat{H}_2 + \cdots \ , \tag{3.8}$$

where $\hat{H}$ is the full Hamiltonian, $\hat{H}_0$ is the simple (zeroith-order) Hamiltonian that we can solve analytically, and $\hat{H}_i$ (i=1,2...) are the ith-order Hamiltonians for interactions that have been neglected in $\hat{H}_0$. The Schrödinger equation can then solved for the simple system,

$$\hat{H}_0\psi_0 = E_0\psi_0 \ . \tag{3.9}$$

Perturbation theory adds the ith-order Hamiltonians to the zeroith-order energy as the average interactions (based on the probability from the wavefunction). This is easiest to understand for first-order perturbation theory, however, the higher-order corrections are systematic and straight-forward. The perturbed Schrödinger equation is given as

$$\hat{H}\psi = E\psi \rightarrow (\hat{H}_0 + \hat{H}_1)\psi = E\psi \ , \tag{3.10}$$

and

$$E \approx E_0 + E_1 \approx E_0 + \langle \psi_0 | \hat{H}_1 | \psi_0 \rangle \ . \tag{3.11}$$

Note that first-order perturbation theory uses the zeroith-order wave function for the averages. This can be seen as both an advantage and a disadvantage in the calculations. It speeds up the calculation to use a lower-order wavefunction since the higher order terms have no effect on the solution of the simpler system. The disadvantage is that wavefunction itself does not respond to the higher-order interactions. For helium with a first-order perturbation, the electrons are too close to the nucleus since there is a strong nuclear attraction with no repulsion due to the second electron.

It should be noted that perturbation theory can correct both the wavefunction and the energy. However, the corrections to the wavefunction can be two orders lower than the corrections to the energy. So a third-order correction to the energy requires a first-order correction to the wavefunction.

3.3.3 Variational theorem

Another method for solving the Schrödinger equation involves approximating the wavefunction for a complex system. If the Hamiltonian for a system is known, but not the wavefunction, we can often make an educated guess. Since the ground-state of a system is the one with the lowest energy, any guess-wavefunction, ϕ, must have an energy that is greater-than or equal-to the ground-state energy. If the energy calculated using ϕ is equal-to the true ground-state energy, then ϕ is the true ground-state wavefunction. If $\hat{H}\phi = E\phi$ predicts an energy lower than the ground-state energy, then the ground-state is not the lowest energy state (impossible by definition). This theorem can be proven mathematically, however, it is conceptually intuitive.

While the variational theorem may sound useful since a reasonable guess wavefunction must be provided, we can often employ the properties of linear combinations and the solutions of $\hat{H}_0\psi_0 = E_0\psi_0$. When solving differential equations, any linear combination of the solutions to the equation is also a solution. For example, the 1s ground-state of the hydrogen atom is a solution of the Schrödinger equation for the hydrogen atom. The 2s excited state is another solution. Therefore,

$$\phi = c_1\psi_{1s} + c_2\psi_{2s} \ , \tag{3.12}$$

is also a solution for the hydrogen atom. We call this a linear combination of atomic orbitals (LCAO) wave function.

3.3.4 Basis sets

In principle, any set of functions (a basis set) can be chosen as a guess then the linear combination coefficients (c_1 and c_2 above) can be adjusted to lower the energy. The more functions that are included in the basis set, the closer the guess-wavefunction can get to the true wavefunction of the system. A logical choice would be to use hydrogen-like wavefunctions for the individual atoms in a molecule as the basis set, however, the integrals are not easy to calculate.

A more practical solution is to choose a large set of functions that we can easily integrate, for instance, Gaussian functions have great properties that simplify integration. The STO-3G basis set provides a good example of how this is done. A Slater-type orbital (STO) is a systematic hydrogen-like orbital for any atomic state. The STO-3G basis set represents the STOs of the atoms as the combination of 3 Gaussian functions. This produces approximate "hydrogen-like" orbitals that have still have the mathematical properties of Gaussians. For the hydrogen molecule, the wavefunction would be given by

$$\psi \approx c_a \phi_{a,1s} + c_b \phi_{b,1s} \ , \tag{3.13}$$

where $\phi_{a,1s}$ and $\phi_{b,1s}$ are the 1s orbitals on the first and second hydrogen atom, respectively. If ϕ_n is represented by the STO-3G basis set, the wavefunciton is given by

$$\psi \approx c_a(G_1^a + G_2^a + G_3^a) + c_b(G_1^b + G_2^b + G_3^b) \ , \tag{3.14}$$

where G_i^j is the ith Gaussian functions centered on atom j. This approximation produces a lot of Gaussian integrals, but only has two coefficients that can be changed to lower the energy. Hence, the basis set has two relatively easy to integrate orbitals made of three Gaussians each.

Next, let's consider the Pople basis sets which are written as w-xyzG. This notation indicates how many Gaussians are used to represent the atomic orbitals. For now, we can just use the lithium atom as the example for the wavefunction,

$$\psi = c_1\phi_{1s} + c_2\phi_{2s} + c_3\phi_{2p,x} + c_4\phi_{2p,y} + c_5\phi_{2p,z} \ . \tag{3.15}$$

Lithium has five orbitals and five variational coefficients if we consider ϕ_n to be an STO, but we are going to use more complicated basis sets.

The w in w-xyzG represents the number of Gaussian functions used to represent the core orbitals (filled inner shells) and (x,y,z) represent the number of Gaussian functions in the valence orbitals. For the moment let's ignore the p orbitals (ψ') and consider the 3-21G basis set (w=3,x=2,y=1,z=0) for the s orbitals.

$$\psi' = c_{1w}(G_1 + G_2 + G_3) + c_{2x}(G_4 + G_5) + c_{2y}G_6 \ . \tag{3.16}$$

The first three Gaussian functions represent the 1s orbital and have a single variational coefficient. The second three Gaussians split the valence orbital into a two Gaussian and one Gaussian function with two additional variational parameters. These split valence basis

sets are more flexible and perform better compared to the STO valence states. The full 3-21G basis set for lithium has nine variational parameters and a total of fifteen Gaussian functions. The larger 6-311G basis set splits the valence states into three parts (thirteen variational parameters, twenty total Gaussian functions). The number of "parts" representing the valence orbitals is referred to as the zeta level of the basis set. A single zeta basis set has one variational parameter per valence orbital (e.g. STO-3G), a double zeta basis set has two variational parameters per valence orbital (e.g. 6-31G), and a triple zeta basis set has three parameters per valence orbital (e.g. 6-311G).

The Pople basis sets are often used with polarization and/or diffuse Gaussian functions. Polarization functions are additional p, d, f, etc. orbitals on top of the standard functions. The 6-31G(d) basis set adds d orbitals to the heavy atoms (i.e. not hydrogen) and 6-31G(d,p) also adds p orbitals to the hydrogens. This can be made much more complicated (i.e. 6-31G(df,pd)) but the notation is straight-forward. There is also an older notation for the polarization functions using * instead of the orbitals (6-31G*=6-31G(d) and 6-31G**=6-31G(d,p)). Diffuse functions are very large s orbitals (e.g. 5s, 6s) added to the atomic basis sets. The notation is 6-31+G and 6-31++G for adding diffuse functions to the heavy atoms and all atoms respectively. The polarization and diffuse functions are often vital for small molecules, but can be problematic in large systems.

Dunning basis sets have a simpler notation, but are mathematically much more complicated. The notation is cc-pVnZ, where n is the zeta level. The notation stands for "correlation consistent polarized valence n zeta" with n=D,T,Q,5,6,etc. A cc-pVDZ basis set is approximately the same as the 6-31G(d,p) basis set. Diffuse functions are added to all atoms with aug- (augmented), i.e. aug-cc-pVnZ. The Dunning basis sets are designed to systematically converge to the infinite basis set, however, they are much more computationally demanding than the Pople basis sets.

3.3.5 Pseudopotentials

Large or complicated systems can be simplified by discarding core electrons and modifying the electron-nuclei interactions for the valence states. A simple (albeit extremely inaccurate) example would be to combine the core electrons of an atom with the nucleus (e.g. oxygen would have a nuclear charge of six and four valence electrons). The modified potentials are referred to as pseudopotentials. Pseudopotentials are useful for a variety of reasons, they reduce the number of electrons, the least important electrons are removed, basis functions have a difficult time modeling the core region, and they can be designed to include relativistic effects.

Real pseudopotentials do not change the nuclear charge, but rather add an electrostatic potential with the same spatial distribtion as the electrons they replace. A futher advantage of pseudopotentials, which is employed in QMMM calculations, is that the potentials can replace entire atoms. For instance, a methyl group can be modeled as a fluorine atom combined with a pseudopotential.

3.3.6 Orbital-free density functional theory

Density functional theory (DFT) is an elegant solution to problems in quantum theory and is distinct from the other formalisms (matrix mechanics (Heisenberg), wave mechanics (Schrödinger), and the path-integral approach (Feynman)). The idea is that you do not need to know the wavefunction, only the electron density of the system, to calculate the energy. It can be proven that there is a one-to-one relationship between a given electron density and the energy of the system. This greatly simplifies the problem the problem since electrons contribute three degrees of freedom (each) to the Schrödinger equation, while the density, ρ, always has a total of three degrees of freedom.

$$\hat{H}\psi = E\psi \rightarrow E[\rho] = T[\rho] + V[\rho] \; . \tag{3.17}$$

DFT, in its original construction, does not require orbitals or basis sets. The entire problem can be solved as a function of $\rho(x, y, z)$. Unfortunately, the exact functional $E[\rho]$ is unknown and the functional must be approximated. Functionals are often based on the homogeneous electron gas, and the nuclei are treated as an external field which perturbs the electrons. Current functionals are only approximate, however, they often provide a computationally efficient solution for large systems. In general the electron-electron and electron-nucleus parts of the functional are much more accurate than the kinetic energy functional. In fact, the simplest functional for the potential (LDA) is still used, but the simplest kinetic energy functional (Thomas-Fermi, TF) cannot even predict chemical bonding. A more accurate approximation for the kinetic energy are discussed below.

3.3.7 Self-consistent field methods

The self-consistent field (SCF) or Hartee-Fock (HF) formalism takes advantage of the topics discussed above to find a simple but relatively accurate solution to the Schrödinger equation. The method iteratively solves the equations using the Fock operator, F,

$$E = \langle\phi|\hat{F}|\phi\rangle \; , \tag{3.18}$$

$$\hat{F} = \hat{H}_0 + \langle\phi|\hat{H}_1|\phi\rangle \; , \tag{3.19}$$

until the variational wavefunction no-longer changes. The problem converges since the Fock operator depends on the average electron-electron repulsion from the previous iteration. The result is an approximate first-order energy and a first-order wave function.

The HF method can be improved by adding additional perturbations. A second-order correction to the HF energy is referred to as MP2, and up to MP4 are often discussed in the literature. The MP2 energy, E_{mp2}, is given by

$$E_{mp2} = E_{hf} + \langle E_2\rangle \; , \tag{3.20}$$

where E_{hf} is the HF energy and $\langle E_2\rangle$ is the second-order perturbation.

There is a bit of terminology that has thus far been missing from the discussions. The first is "electron correlation", which refers to the interactions and motions of the electrons (i.e. an electron wants to move away from another electron). In the literature, the "correlation energy" is the electron-electron interaction energy not captured by the HF method (second-order and higher perturbations). The SCF formalism is only a mean-field approach, and hence, it gives the average interactions not the true interactions. The second bit of terminology is the "exchange interaction", which is included in $\hat{F}$ through $\hat{H}_1$. The solution of the Schrödinger equation for electrons should be anti-symmetric with respects to the exchange of two electrons. The exchange interaction is added to the $\hat{H}_1$ operator to force the variational wavefunction to have the correct symmetry.

Hartree-Fock calculations capture the exchange, but miss the higher-order correlation. Density functional theory calculations, on the other hand, capture some of the higher-order correlation, but miss some of the exchange interaction.

3.3.8 Kohn-Sham DFT

Kohn-Sham (KS) DFT is a SCF approach similar to HF theory, however, it does not obey the variational theorem. To overcome the failures of the orbital-free kinetic energy functionals, orbitals are reintroduced to the problem so that the kinetic energy can be calculated "exactly" from a basis set. As with the HF method, the electrons are moving in a mean-field of all other electrons, but are not interacting directly. Since KS-DFT has orbitals, the exchange interaction can also be calculated "exactly" with only a modest increase in computational cost. However, since most functionals include an exchange term, hybrid functionals add in a percentage of the exact (HF level) exchange and remove the same percentage of DFT exchange. Researchers can make entire careers out of developing better functionals (there are currently thousands) and the mathematical structure of the functionals can be quite complicated. However, some of the most common functionals are LDA (based on the electron gas, used in solid state physics), PBE (contains no empirical parameters, used in physics), PBE0 (a hybrid form of PBE with 25% HF exchange, used in physics), and B3LYP (a hybrid functional with three empirical parameters, used heavily by chemists).

3.3.9 Dispersion corrections

KS-DFT does a reasonably good job for describing bonded interactions, metals, geometries, and energies. However, DFT poorly describes van der Waals interactions, such as dispersion. Modern functionals are often dispersion corrected by adding empirical corrections to the nuclei-nuclei interactions. For the most part, the dispersion energy, E_d, takes the form

$$E_d = \sum_{i,j} f(R_{ij}) \frac{C_6}{R_{ij}^6} \,, \tag{3.21}$$

where C_6 is an emprical parameter, $f(R_{ij})$ is a damping function, and R_{ij} is the distance between atoms i and j.

3.4 QMMM

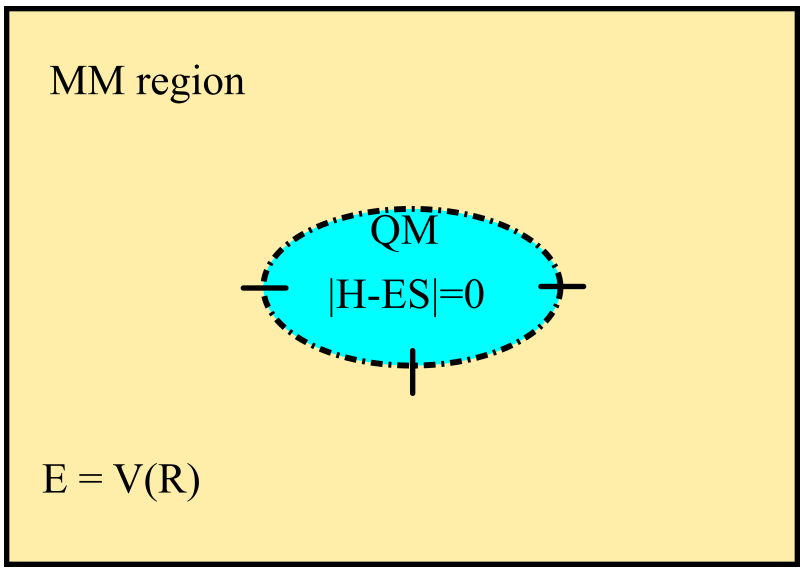


Figure 3.1: Schematic of a QMMM simulation box with bonds between the QM and MM regions.

QMMM calculations divide the system into regions which require accurate quantum treatment and regions that can safely be approximated by classical models (see Figure 3.1). The full effective Hamiltonian, $\hat{H}_{eff}$, can be written as

$$\hat{H}_{eff} = \hat{H}_{qm} + \hat{V}_{mm} + \hat{V}_{qmmm} \ , \tag{3.22}$$

where $\hat{H}_{qm}$ is the Hamiltonian for the QM region, $\hat{V}_{mm}$ is the classical potential for the MM region, and $\hat{V}_{qmmm}$ is the potential for the interaction of the QM and MM regions.

Since the MM and QMMM interface are included only through the potential, solving the Schrödinger equation is essentially trivial.

$$E \approx \langle \psi | \hat{H}_{eff} | \psi \rangle \tag{3.23}$$

3.4.1 QM-MM interactions

When the QM and MM regions are not connected by covalent bonds, the QM-MM interactions are reasonably straight-forward. The interaction between the electric field of the MM region (point charges or multipoles) and the QM region is calculated by adding point charges to the QM calculations. This allows for electrostatic repulsion and polarization, but neglects exchange and dispersion interactions. These missing interactions are added using the vdW potentials taken from the MM force field.

For a single-point energy, the QM+charges calculation is performed with the QM wrapper, while the MM wrapper calculates the MM energy with no charges or bonded potentials on

any of the QM atoms. These two energies are then added together to get the total energy. This approach splits $\hat{V}_{qmmm}$ between the MM calculation ($\hat{V}_{mm}$) and the QM calculation ($\hat{H}_{qm}$).

3.4.2 Pseudo-bond method

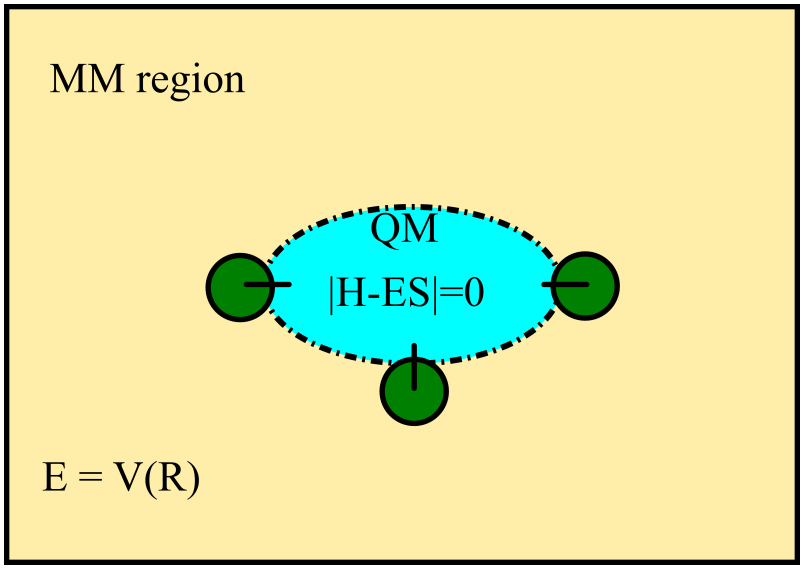


Figure 3.2: Schematic of a QMMM simulation box with the boundary region show in green between the QM and MM regions.

The calculations are more complicated when there are bonds between the QM and MM regions. Covalent bonds require two additional regions of the QMMM system, a pseudo-atom (PA) region and a boundary atom (BA) region. Pseudo-atoms are shared by the QM and MM regions, and handle the bulk of the covalent interactions. The severed bonds from the MM region are treated by adding a pseudopotential to the pseudo-atoms that allows a fluorine atom to mimic the behavior of an sp^3 hybridized carbon or nitrogen atom. Boundary atoms correct for two additional errors. First, having point charges close to the QM region can cause the electron density to be over-polarized. The second error is introduced by the fact that the QM region must have an integer number of electrons. Both of these problems can be mitigated by absorbing the point charges on the atoms bonded to the pseudo-atoms into the QM region during the QM calculation. This produces atoms with zero-charge on the boundaries between the QM and MM regions.

For the MM part of the calculation, the pseudo-atoms and boundary atoms retain their force field charges, and the MM bonding interactions are added for the pseudo-atoms. The treatment of different interaction types are given in Table 3.1 for single-point energy calcuations. In the pseudo-bond approach, only some of the bonded interactions are calculated for the QMMM interface. Further detials on which bonded interactions are included can be found in the literature.

	Wrapper calculation	
Int. type	MM	QM
MM-MM	FF	Zero
QM-QM	Zero	ρ_{el}
PA-PA	FF	PP+ρ_{el}
BA-BA	FF	Zero
QM-MM	FF-$\{q_{qm}\}$	ρ_{el}+$\{q_{mm}\}$
QM-PA	Zero	PP+ρ_{el}
QM-BA	FF-$\{q_{qm}\}$	Zero
MM-PA	FF	PP+ρ_{el}+$\{q_{mm}\}$
MM-BA	FF	Zero
PA-BA	FF	Zero

Table 3.1: The treatment of region-region interactions in the MM and QM wrappers during single-point energy calculations. Interactions were abreviated as follows: force field (FF), electron density (ρ_{el}), pseudopotential (PP), no interaction (Zero), QM point charges ($\{q_{qm}\}$), and MM point charges ($\{q_{mm}\}$). A "-" sign is used if the interaction is removed instead of added (i.e. FF-$\{q_{qm}\}$ denotes the force field with no charges on the QM atoms).

So far we have discussed systems similar to the one shown in Figure 3.2, where the boundary atoms only surround the covalent bonds. This is due to the primative treatment of the QM-MM boundaries. A more complicated scheme can be employed where the boundary atoms are represented by a polarizable frozen density force field. The QM and frozen density interactions are more natural than interactions between the QM and point charges, and thus, this approach avoids the over-polarization. Boundary atoms represented by frozen electron density can be extended further from the QM region and can create smoother boundaries between the QM and MM regions. Ideally, a system could be constructed with a large boundary region between the QM and MM regions (see Figure 3.3).

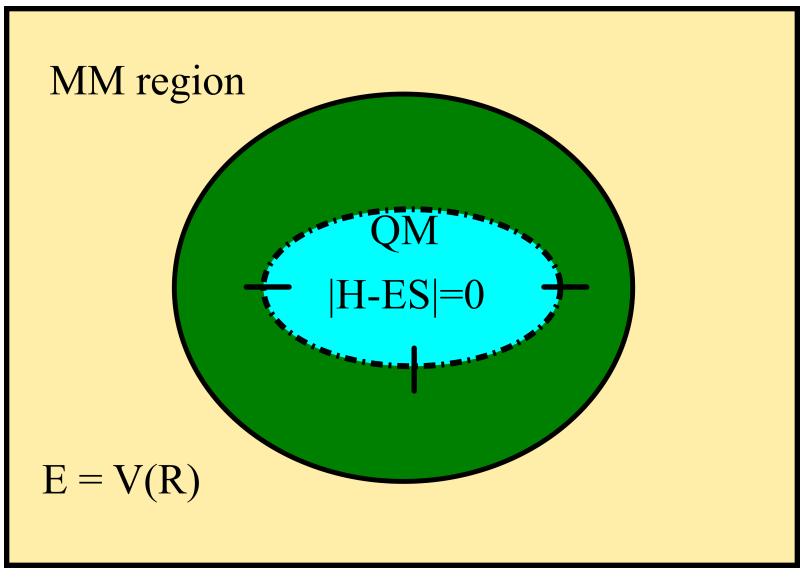


Figure 3.3: Schematic of a QMMM simulation box with a large boundary region between the QM and MM regions.

3.5 Monte Carlo Simulations

3.5.1 Stochastic sampling

Monte Carlo simulations sample phase space by generation random changes to the system. Completely random sampling could eventually find a global minimum, however, majority of the random structures would have energies well above kT and contribute very little to the statistical averages.

Almost any type of change can be made to the system in a Monte Carlo simulation. The primary limitations are the creativity of the programmer and the efficiency of making the changes. This allows Monte Carlo simulations to be tailored to the system and focus on the randomly changing the most important features of a system.

If the moving atoms and/or type of random change are chosen randomly, this proceedure satisfies "detailed balance." I.e. the move from configuration i to configuration j can be reversed by a move from configuration j to configuration i.

3.5.2 Canonical ensemble

A more efficient approach is to accept or reject the random moves based on the Boltzmann weight of the configuration.

$$P_{acc} \propto e^{-\Delta E \beta} \ , \tag{3.24}$$

where P_{acc} is the probability of accepting the move, ΔE is the change in energy, and β is the inverse temperature,

$$\beta = \frac{1}{kT} \ . \tag{3.25}$$

This proceedure leads to biased sampling of the lower energy states.

Unlike molecular dynamics simulations, the natural ensemble for Monte Carlo simulations is the canonical enemble (NVT). This is advantageous since there is no need to couple the simulation to a thermostat.

3.5.3 Isobaric-isothermal ensemble

Although NVT simulations are natural for Monte Carlo, it is relatively easy to change to the NPT ensemble. NPT simulations are performed by randomly changing the volume of the simulation box. This proceedure produces a slightly different expression for the probabilities,

$$P_{acc} \propto e^{-(P\Delta V+\Delta E)\beta+N\Delta ln(V)} \ , \tag{3.26}$$

where P is the pressure, ΔV is the change in volume, and N is the number of atoms.

3.6 Path-Integral Monte Carlo

3.6.1 Path-integral formalism

3.6.2 PIMC simulations

Path-integral Monte Carlo simulations are performed by simulating a large number of classical systems which are then coupled together via harmonic bonds. Allowed Monte Carlo moves include displacements of single beads, translations of centroids, and changes of the volume. The spring energy between the beads represents the quantum kinetic energy of the centroid. Moves are accepted based on the effective energy, E_{eff}

$$E_{eff} = E_{spring} + \frac{1}{N_p}\sum_{i}^{N_p} E_{pot,i} \ , \tag{3.27}$$

where the spring energy is given by

$$E_{spring} = \sum_{j}^{N_a} \frac{m_j N_p}{\hbar^2\beta^2} \sum_{i}^{N_p} (r_i - r_{i-1})^2 \ . \tag{3.28}$$

Here the N_a is the total number of atoms, N_p is the number of beads, r_i is the position of bead i, and E_{pot} is the potential energy. The acceptence probability is given by

$$P_{acc} \propto e^{\Delta E_{eff}\beta} \ . \tag{3.29}$$

The path-integral total energy is slightly different from the effective potential.

$$E_{pi} = \frac{3N_aN_p}{2\beta} - E_{spring} + \frac{1}{N_p}\sum_{i}^{N_p} E_{pot,i} \ , \tag{3.30}$$

where E_{pi} is the total energy. The first two terms in Eq. 3.30 represent the classical kinetic energy for all particles and the amount of kinetic energy absorbed by centroid harmonic bonds.

3.6.3 Ergodicity