

This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1063/5.0219851

PyDFT-QMMM: A Modular, Extensible Software Framework for DFT-Based QM/MM Molecular Dynamics

John P. Pederson¹ and Jesse G. McDaniel^{1, a)}

*School of Chemistry and Biochemistry, Georgia Institute of Technology
Atlanta, Georgia, 30332-0400, United States*

(Dated: 21 June 2024)

PyDFT-QMMM is a Python-based package for performing hybrid quantum mechanics/molecular mechanics (QM/MM) simulations at the density functional level of theory. The program is designed to treat short-range and long-range interactions through user-specified combinations of electrostatic and mechanical embedding procedures within periodic simulation domains, providing necessary interfaces to external quantum chemistry and molecular dynamics software. To enable direct embedding of long-range electrostatics in periodic systems, we have derived and implemented force terms for our previously described QM/MM/PME approach [Pederson and McDaniel, J. Chem. Phys. **156**, 174105 (2022)]. Communication with external software packages Psi4 and OpenMM is facilitated through Python application programming interfaces (APIs). The core library contains basic utilities for running QM/MM molecular dynamics simulations, and plug-in entry-points are provided for users to implement custom energy/force calculation and integration routines, within an extensible architecture. The user interacts with PyDFT-QMMM primarily through its Python API, allowing for complex workflow development with Python scripting, for example interfacing with PLUMED for free energy simulations. We provide benchmarks of forces and energy conservation for the QM/MM/PME and alternative QM/MM electrostatic embedding approaches. We further demonstrate a simple example use case for water solute in water solvent system, for which radial distribution functions are computed from 100 ps QM/MM simulations; in this example we highlight how the solvation structure is sensitive to different basis-set choices due to under- or over-polarization of the QM water molecule's electron density.

PACS numbers: 07.05.Tp, 31.15.-p, 31.15.E-, 31.15.Ne, 02.70.Ns, 02.70.-c, 02.90.+p,
Keywords: Density functional theory, Molecular dynamics, Quantum mechanical/-molecular mechanical calculations, Electrostatics, Computational chemistry, Software

^{a)}Electronic mail: mcdaniel@gatech.edu

I. INTRODUCTION

Hybrid quantum mechanics/molecular mechanics (QM/MM) is a multiscale modeling methodology for investigating condensed-phase, chemical processes with atomistic resolution. It is widely understood that the influence of the condensed-phase environment can have a profound impact on the chemistry of solutes,^{1–6} and a variety of computational methods have been devised to capture such effects.^{3,7–10} For most applications, it is intractable to explicitly model extended solvent environments with full *ab initio* wavefunction or density functional theory (DFT) quantum chemistry (QC) treatments,¹¹ so the interaction between the solute and the solvent must be modeled approximately, at a lower level of theory. Solvent models are typically constructed in one of several ways: implicitly, through reaction field and polarizable continuum models;^{8,12–15} explicitly, through atomistic QM/MM, where solvent molecules are represented by an MM forcefield;^{8,16,17} or as a combination of explicit and implicit representations, such as in QM cluster-continuum models,^{8,12,13,18,19} boundary potential models,^{12,20–24} and QM/MM/Continuum methods.^{7,12,25,26} There is no general heuristic for the choice of solvent representation for a given chemical system,^{8,9,18} and the quality of computations will depend greatly on the parameterizations of the implicit or explicit solvent model;^{6,27,28} however, explicit solvent representation offers the ability to capture dynamic effects of the solvent at an atomistic resolution. Accordingly, QM/MM methodology has been employed to investigate a wide range of phenomena in the condensed phase, including organic chemical reactions,^{6,29–32} excited state dynamics,^{16,33–37} and enzymatic and heterogeneous catalysis.^{10,16,37–41}

The application of dynamic QM/MM simulations combines the methodologies of multiple disciplines, including quantum chemistry, molecular mechanics, and molecular dynamics (MD) simulations. The interdisciplinary nature of multiscale modeling presents both theoretical and technical challenges which must be addressed,^{42,43} often in concert. The basic QM/MM formalism involves a subsystem treated at a QM level of theory, embedded within a solvation environment treated at the MM level of theory. These subsystems of the simulation domain are generally referred to as the “QM region” and “MM region” in the literature; however, we opt for the nomenclature of Truhlar, Lin, and co-workers—where the QM region is referred to as the primary subsystem and the MM region is referred to as the secondary subsystem^{17,44–46}—in order to decouple levels of theory from physical regions of the chemi-

cal system under study. Practical considerations include how to partition the system, which levels of theory to apply in calculations, and how to model the physical coupling between subsystems.^{9,47} Because common applications of QM/MM including molecular dynamics simulations and geometry optimizations require many sequential force and/or energy calculations, several methodological decisions that impact computational speed/tractability must be made.

The QC routines are typically the most computationally expensive part of a QM/MM calculation,^{9,11,48} so both the size of the primary subsystem and the level of electronic structure theory will dictate computational cost. The membership of the primary subsystem should at least include the moieties which are directly involved in the chemical process of interest, e.g. charge transfer or bond formation and fission.^{47,49–51} The choice of primary subsystem will also depend on the system-size scaling of the QC method. Fast-scaling semiempirical (SE) QC methods, including those based on the neglect of diatomic differential overlap (NDDO) approaches,⁵² density functional tight-binding (DFTB),^{53,54} and related empirical valence bond (EVB) formalisms,⁵⁵ have been widely used in QM/MM studies due to their computational efficiency. The steady growth of computational resources and intermittent algorithmic improvements has also led to increasing application of DFT^{11,48,56} and correlated wavefunction^{37,57,58} QC methods in QM/MM studies in recent years. In this regard, it is generally understood that DFT provides a good compromise between speed and accuracy.^{9,37,42,47} While our methodological and software discussion focuses on QC software that utilizes atomic orbital (AO) bases, alternative QM/MM implementations have utilized planewave bases,^{59–62} mixed Gaussian-planewave bases,^{63–65} numerical orbital bases,^{66,67} or real-space grids.^{68–70}

The physical coupling between the secondary and primary subsystems is a key component of QM/MM methods that warrants discussion. Interactions between subsystems are calculated through “embedding” schemes. The most basic form of embedding is mechanical embedding, whereby the interaction between the primary and secondary subsystems is mediated entirely at the MM level of theory.^{16,17,71–73} This clearly has limitations when the primary region is not well described by a classical forcefield,^{9,16,44,72,74} for example going from reactants to transition state to products in a chemical reaction. Electrostatic embedding is the standard embedding practice in QM/MM methodology,^{16,17,72,73} and it allows the charge distribution of the secondary subsystem to polarize the electron density of the

primary subsystem during the self-consistent field (SCF) procedure. Such an approach requires that the QC software provide the option to add the potential from arbitrary point charges into the SCF calculation and that gradients at the embedded point charge coordinates are available for dynamics propagation. Electrostatic embedding schemes differ in their treatment of long-range electrostatics. Some QM/MM approaches utilize a cluster model with finite boundaries, coupled with a reaction field representation of the extended environment.^{15,20–22,25,75} Alternatively, the system may be modeled with periodic boundary conditions (PBC), with different possible treatments of electrostatic embedding; real-space embedding is incorporated up to distance-based cutoff, with longer range interactions truncated through smoothing or shifting functions,^{76–79} accounted for using Ewald summation approaches,^{80–93} or modeled with other techniques.^{94–96}

This manuscript describes our recently developed PyDFT-QMMM software for DFT-based, QM/MM molecular dynamics simulations. Our software development has been motivated and inspired by the ever expanding role of QM/MM methods for investigating complex chemical systems, and the need for versatile multiscale modeling machinery and interoperable software. There is a long history of QM/MM software development;^{97,98} in this regard it is profitable to discuss prior work and implementation paradigms to provide context for our present software development efforts. The first application of QM/MM made use of spectroscopy-derived forcefield parameters and an early SE QC program in 1976,⁹⁹ and the first publicly distributed QM/MM software was developed 10 years later: QUEST, an interface between Gaussian-80 (UCSF) and AMBER.¹⁰⁰ In the subsequent decades, a plethora of QM/MM software implementations were developed,^{61–64,66–70,75,83,101–171} in addition to countless research codes that were never broadly reported or distributed.¹⁷² These QM/MM software implementations can be roughly classified according to their architecture as: standalone packages that contain both QC and MD routines, dedicated interfaces between extant QC and MD packages, and general interfaces between QC and MD packages.^{17,98,162}

Standalone QM/MM implementations have historically grown out of pre-existing QC or MD software,^{17,98} where the package realizes an implementation of the complementary QM/MM component. MD packages that have incorporated QC routines include CHARMM,^{102,113} BOSS,^{104,173} AMBER,^{83,105,108} and GROMACS,¹³⁶ but the implementation of internal QC routines is typically limited to SE QC methods. General electronic structure packages that have incorporated MM or MD routines include Gaussian,^{106,174,175}

HONDO,⁷⁵ NWChem,¹³³ ADF,¹³⁷ PQS,¹³⁴ and GAMESS.¹³⁵ While a greater variety of QC methods are offered by electronic structure packages, the MM implementations within these packages may be limited to a few general forcefield parameterizations or simple dynamics propagation schemes. We further note a contingent of specialized DFT codes which have incorporated MM routines, including CP2K,^{63,64} DeMon2k,^{119,132,143} CP-PAW,^{111,112} and the grid-based DFT software of Takahashi and co-workers.^{69,70} Other standalone QM/MM packages have been developed specifically to employ QM/MM methodology, as in the case of MOPS,¹³¹ Dynamo/pDynamo,^{103,114,130} or Platypus.¹¹⁷ While the standalone architectural approach offers the convenience of utilizing a single, “monolithic” package,^{172,176,177} it requires significant initial and continuous effort to write, test, and maintain duplicate implementations of existing methods. Given that there are many MD and QC programs that are designed and optimized by specialists within their respective fields, it seems natural to realize QM/MM methodology by bridging the functionality of these packages.⁴³

Many QM/MM implementations have been realized through interfacing between existing packages. One such approach is the creation of dedicated interfaces between QC and MD packages,^{61,62,66,67,100,138–148} which benefits from the strengths of the chosen packages. It has been noted that many early interfaces of this type relied on modifications to the respective package source codes, requiring re-compilation and introducing tight coupling between the packages.^{116,129,140,142} Without community and official package support, such interfaces are difficult to maintain or install. The challenges arising from tightly coupled interfaces are bypassed by devising alternate schemes for package interoperability, which may be accomplished through input/output (I/O) parsing within a shared file system;^{115,116,139,142,145,166} inter-process communication schemes, such as piping,^{67,152,168,177} message passing,^{116,144,169–171} remote call procedures,^{120,161,162} or shared memory;^{101,129,168} or in-process schemes, where the different programs are loaded in memory and communicate through local method calls.^{68,121,125,167,168,177} Since these communication schemes do not typically require modifications to the underlying QC or MD packages, QM/MM interfaces can be generalized to support arbitrary pairings of QC and MD packages, and these interoperability schemes have been employed in many general QM/MM interface packages^{149–170} as well as more general computational chemistry programs.^{115,171,177}

The speed and flexibility of available communication schemes is a central concern when performing QM/MM simulations. The I/O file scheme is simple and relatively maintainable

due to the stability of file templates in packages, but program initialization, program termination, and file system access introduce computational overhead.^{144,158,161,162,167,171} Inter-process communication permits QC and MD programs to run concurrently, avoiding the latency associated with starting and stopping programs. In-process communication additionally benefits from having all programs loaded in memory, where they may access or write data without the duplication required to communicate between concurrent processes.^{167,168,177} In-process communication is typically facilitated by an application programming interface (API), which may be implemented in one or another programming language. APIs utilizing Python bindings have steadily grown in popularity within computational chemistry over the past decade,^{68,125,171,177–181} owing to its human-friendly syntax^{177,180,182} and its extensive ecosystem of other advanced packages.^{125,177,180} These features make Python a scripting language *par excellence*, enabling users to create more advanced workflows.^{115,122,155,156,171,177–183} Several general QM/MM interfaces with Python-based communication have been developed, including ASE,^{109,177} ASH,¹²¹ Janus,¹⁶⁷ and QMCCube.¹⁶⁸ We further note that QM/MM packages employing other communication schemes have implemented Python APIs, including ChemShell,^{155,156} MiMiC,¹²² ADF,¹¹⁵ and the MolSSI Driver Interface.¹⁷¹

While communication techniques have generally improved and language bindings have spread, QM/MM methodological considerations may still be limited by the capabilities of the underlying QC and MD software. The benefit of pairing arbitrary QC and MD packages is apparent: the user is able to access implementations of myriad QC and MD approaches as well as differing implementations of comparable approaches,^{121,168,171,177} however, the generality of interfaces to external software may come at the cost of the availability and extensibility of QM/MM approaches. More general frameworks implement different communication schemes depending on the selected packages, and the full range of QM/MM capabilities are not necessarily shared across programs. Accordingly, the extent and performance of QM/MM functionality is dependent on user input, requiring that the user knows some of the underlying software implementation details. The inconsistency between interfaces also presents a challenge to the extension of QM/MM embedding techniques within these frameworks. Licensing and closed-source distributions of software may further complicate the application and development of QM/MM approaches.^{167,184,185} As an example of an alternate philosophy, the authors of the Janus package only interface their program to open-source packages with Python bindings and explicitly denote the applied QM/MM

schemes.¹⁶⁷ The Janus package further abstracts partitioning and embedding procedures, allowing for facile extension; however, the generation of system data and the propagation of simulations is tightly coupled to the MM interface. This is a natural choice, given that MD software is built for simulation, but the separations between system construction, data containers, energy and force calculations, and dynamics propagation are useful abstractions used in other software.^{177,181,186}

In this work, we present PyDFT-QMMM: a modular, lightweight, and entirely Python-based QM/MM package for performing DFT-based QM/MM MD simulations. The software is built upon the Psi4 electronic structure software^{179,187} and OpenMM molecular dynamics software,^{181,188} acting as QC and MM calculation engines, respectively. PyDFT-QMMM provides a flexible framework for modifying or extending embedding, partitioning, and sampling routines. This flexibility is facilitated through Python-based interfaces to Psi4 and OpenMM for in-process communication, as well as adherence to an object-oriented design philosophy that stresses separation of responsibilities, clear contracts and specifications defined in abstract classes, and provision of hooks and templates for extension. Additionally, we employ a user philosophy comparable to that of OpenMM,^{181,186} privileging scripting over input file parsing. In accord with this philosophy, the data structures of PyDFT-QMMM have parallel behavior to and are able to interoperate with the data structures found in the Python Standard Library and the NumPy library. These features permit finer control over embedding procedures, including those relevant to QM/MM within a periodic simulation domain. The remainder of this manuscript is organized as follows: the mathematical QM/MM formalism and a taxonomy of embedding approaches are presented in Section II, the implementation details of the PyDFT-QMMM package are presented in Section III, and benchmarks and simple example use cases for the package are presented in Section IV.

II. THEORY

A. QM/MM Hamiltonian

QM/MM Formalism. QM/MM calculations begin with the construction of a universe, $\mathbb{U}$, from which an explicit system, $\mathbb{S}$, and an environment, $\bar{\mathbb{S}}$, are defined, as in Figure 1a. The chemical subsystem of interest and some amount of the solvent must be part of the

explicitly-modeled simulation domain $\mathbb{S}$ in order to replicate the structure and behavior of the condensed phase. Conventional MD makes use of PBC, where the simulation domain $\mathbb{S}$ comprises the principal simulation box or cell, and the surrounding environment $\bar{\mathbb{S}}$ consists of periodic replicas of $\mathbb{S}$ (Figure 1c). Various QM/MM schemes have been proposed in which $\mathbb{S}$ comprises the primary and secondary subsystems, and $\bar{\mathbb{S}}$ might be a reflecting wall, a boundary potential, a stochastic boundary, or periodic replicas of $\mathbb{S}$ as modeled utilizing PBC, similar to conventional MD. In this work, we will focus on the specific choice of periodic boundary conditions, such that $\bar{\mathbb{S}}$ consists of periodic replicas of $\mathbb{S}$. While it is straightforward to partition a simulation domain into primary and secondary subsystems, a thorough discussion of long-range interactions necessitates a subtle refinement in our nomenclature. We extend the definitions of primary and secondary subsystem presented in Section I to further partition the “MM region” and complement $\bar{\mathbb{S}}$ of the simulation domain into a secondary (II) and tertiary (III) subsystem, corresponding to the local and extended solvent environments, respectively (Figure 1b). For periodic boundary conditions (our focus), the tertiary III subsystem will generally include both molecules within the principal simulation box, as well as all periodic replicas of the principle simulation box (Figure 1c). These partitioning schemes, starting from the highest level of abstraction, are depicted in Figures 1a–c.

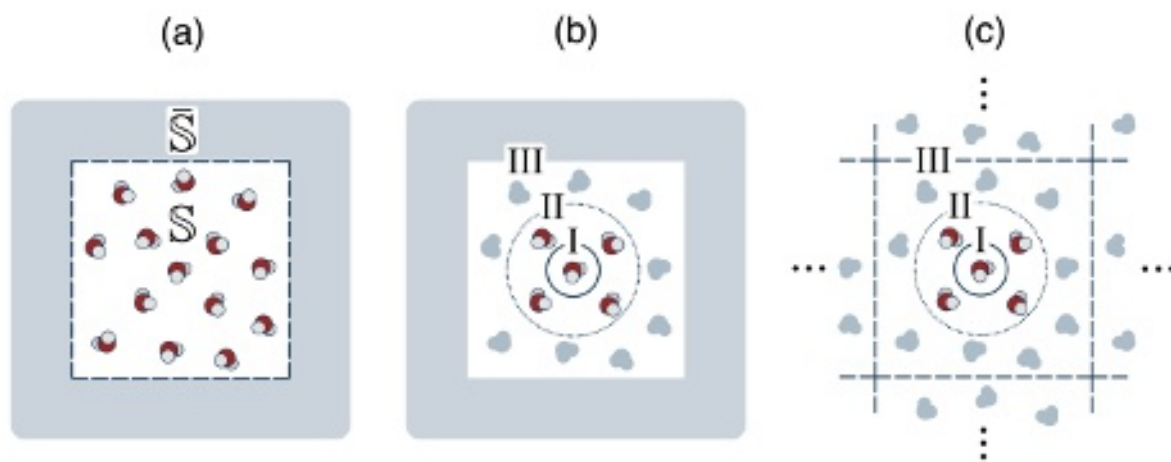


FIG. 1. The QM/MM system may be partitioned according to a) explicit, atomistic representation in the universe of the physical problem or b) proximity to the subsystem whose electronic structure is under study; c) depicts the periodic boundary conditions that are the focus of this work.

Our proposed partition draws a distinction between short/medium-range (region II) and

long-range (region III) intermolecular interactions with the primary subsystem (region I). This division generates a specific I-III term that explicitly represents long-range interactions between the primary and tertiary subsystem, which may otherwise be ambiguous. We note that this partitioning scheme is analogous to that employed in MD codes to evaluate electrostatic interactions, in which I-II and I-III interactions are treated in real space and reciprocal space respectively within Ewald-based methods.¹⁸⁹ The secondary subsystem II thus consists of the atoms or molecules which may be included within a real-space electrostatic embedding procedure, as defined up to some cutoff distance. Among various QM/MM implementations, interactions between the primary (I) and tertiary (III) subsystems may be modeled through reaction field or boundary potential,^{15,20–22,25,75} smoothing or switching functions,^{76–79} real-space summation schemes,^{94–96} reciprocal-space Ewald summation schemes,^{80–93} or entirely neglected.¹⁴⁰ In our approach, termed “QM/MM/PME” as described below, long-range electrostatic interactions between regions I and III are explicitly incorporated utilizing particle-mesh Ewald algorithms.

The total Hamiltonian describing the QM/MM system is

$$\begin{aligned} H^{\text{Total}} &= H^{\text{QM}} + H^{\text{MM}} + H^{\text{QM/MM}} \\ &= H_{\text{I-I}}^{\text{QM}} + H_{\{\text{II} \cup \text{III}\}-\{\text{II} \cup \text{III}\}}^{\text{MM}} + H_{\text{I-II}}^{\text{QM/MM}} + H_{\text{I-III}}^{\text{QM/MM}} \\ &= H_{\text{I}}^{\text{QM}} + H_{\text{II,III}}^{\text{MM}} + H_{\text{I-II}}^{\text{QM/MM}} + H_{\text{I-III}}^{\text{QM/MM}}, \end{aligned} \quad (1)$$

where we introduce our notation in the second line and provide a convenient shorthand in the final line. The superscripts describe the level of theory applied, and the subscripts describe the intra- or inter-subsystem interaction modeled. The QM/MM superscript refers to the coupling of subsystems which may have both QM and MM components. The first term of the final two lines, H_{I}^{QM} , represents the intra-subsystem interactions within the primary subsystem, and we employ Kohn-Sham DFT as the QM level of theory in this work. The second term of the final two lines, $H_{\text{II,III}}^{\text{MM}}$, represents the intra- and inter-subsystem interactions of the secondary and tertiary subsystems and reduces to the energy and force expressions given by an MM (e.g. forcefield) description. The third and fourth terms, $H_{\text{I-II}}^{\text{QM/MM}}$ and $H_{\text{I-III}}^{\text{QM/MM}}$, of Equation 1 represent the interaction between the primary subsystem I and surrounding environment (II and III), as modeled with both QM and MM components.

The coupling Hamiltonian of the primary subsystem, or the QM/MM interaction between the primary subsystem and the environment, may be rewritten as

$$H^{\text{QM/MM}} = H_{\text{I-II}}^{\text{QM,elst}} + H_{\text{I-II}}^{\text{MM,non-elst}} + H_{\text{I-III}}^{\text{QM or MM,elst}}, \quad (2)$$

where the interaction between the primary and secondary subsystems is decomposed into: an electrostatic embedding term, $H_{\text{I-II}}^{\text{QM,elst}}$; and a mechanical embedding term, $H_{\text{I-II}}^{\text{MM,non-elst}}$, which is a forcefield contribution capturing exchange repulsion and dispersion interactions (for example, using a Lennard-Jones potential). For Ewald-based QM/MM approaches, the interaction between the primary and tertiary subsystems, $H_{\text{I-III}}^{\text{QM or MM,elst}}$, can be modeled with mechanical embedding,⁹¹ direct electrostatic/PME embedding,^{86,92} or hybrid schemes in which interactions with the extended environment are described through fitted auxiliary point charges^{83,87–90,93} or effective potentials.^{80–82}

In our direct QM/MM/PME approach,⁹² $H_{\text{I-III}}^{\text{QM or MM,elst}}$ explicitly incorporates electrostatic interactions of the full periodic system with the electron density of the primary (QM) subsystem. It is assumed that a sufficiently long cutoff distance (e.g. 12–14 Å) is used to define the tertiary (III) subsystem (Figure 1), so that only electrostatic interactions are considered between the primary (I) and tertiary (III) subsystems. In other words, $H_{\text{I-III}}^{\text{QM or MM,elst}}$ is purely electrostatic, since exchange repulsion and dispersion interactions are negligible at such length-scales. The QM and MM terms of Equations 1 and 2 can be grouped together and evaluated with QC and MD software packages. For the interactions treated at the MM level of theory, the energy evaluation reduces to the evaluation of forcefield expressions; the only difficulty here is the “bookkeeping” of different interaction terms, which is a software consideration that is discussed in Section III.

Electronic Energy. The QM terms of Equations 1 and 2 require further elaboration. To enable explicit discussion, we now specifically describe our QM/MM/PME approach⁹² in which $H_{\text{I-III}}^{\text{QM or MM,elst}}$ is treated at a QM level. In this case, the electronic energy of the primary subsystem (QM) including electrostatic interactions with surrounding secondary and tertiary subsystems is

$$E^{\text{QM/MM}} = \langle \Psi | \hat{H}_{\text{I}}^{\text{QM}} + \hat{H}_{\text{I-II}}^{\text{QM,elst}} + \hat{H}_{\text{I-III}}^{\text{QM,elst}} | \Psi \rangle, \quad (3)$$

where the use of the “QM/MM” superscript here signifies the complete electronic energy of the QM/MM system. This usage stands in contrast to that in Equation 2, where the superscript denotes interaction between subsystems. We focus on Kohn-Sham DFT as the QM level of theory and so evaluation of Equation 3 in an atomic orbital basis yields (utilizing

Einstein summation convention)

$$\begin{aligned}
 E^{\text{QM/MM}} = & D_{\mu\nu}(T_{\mu\nu} + V_{\mu\nu}) + \frac{1}{2}D_{\mu\nu}J_{\mu\nu} + E^{\text{XC}}(\rho) + \sum_{i<j}^{N_{\text{I}}} \frac{Z_i Z_j}{|\vec{x}_i - \vec{x}_j|} \\
 & - D_{\mu\nu} \int d\vec{r} \phi_{\mu}(\vec{r}) \phi_{\nu}(\vec{r}) \sum_n^{N_{\text{II}}} \frac{q_n}{|\vec{r} - \vec{x}_n|} + \sum_i^{N_{\text{I}}} \sum_n^{N_{\text{II}}} \frac{Z_i q_n}{|\vec{x}_i - \vec{x}_n|} \\
 & - D_{\mu\nu} \int d\vec{r} \phi_{\mu}(\vec{r}) \phi_{\nu}(\vec{r}) v_{\text{III}}(\vec{r}) + \sum_i^{N_{\text{I}}} Z_i v_{\text{III}}(\vec{x}_i),
 \end{aligned} \tag{4}$$

where the first line of the equation corresponds to the purely Kohn-Sham contribution from the primary subsystem, the second line corresponds to the contribution from the I-II interaction, and the third line corresponds to the contribution from the I-III interaction. In Equation 4, $T_{\mu\nu}$, $V_{\mu\nu}$, and $J_{\mu\nu}$ are the standard kinetic energy, nuclear potential, and Coulomb matrices, and $E^{\text{XC}}(\rho)$ is the exchange-correlation energy, as a functional of the electron density ρ . $D_{\mu\nu}$ denotes the density matrix, ϕ_{μ} are atomic orbitals, and Z_i are nuclear charges. We use notation “ $\vec{r}$ ” and “ $\vec{x}_i$ ” to denote electron and nuclear coordinates respectively in the primary subsystem, and “ $\vec{x}_n$ ” to denote atomic coordinates (with corresponding partial charges q_n) in the secondary subsystem. Summation over index “i” thus runs over all nuclear coordinates N_{I} in the primary subsystem, and summation over index “n” runs over all atomic coordinates N_{II} in the secondary subsystem. For simplicity, we do not consider exact Hartree-Fock exchange in Equation 4, but generalization is straightforward.

The QM/MM energy expression in Equation 4 is minimized with respect to the expansion coefficients (comprising the density matrix, $D_{\mu\nu}$) to define the effective Kohn-Sham matrix utilized within the self-consistent field (SCF) procedure. The external potentials from the secondary (II) and tertiary (III) subsystems correspond to matrices:

$$V_{\mu\nu}^{\text{II}} = - \int d\vec{r} \phi_{\mu}(\vec{r}) \phi_{\nu}(\vec{r}) \sum_n^{N_{\text{II}}} \frac{q_n}{|\vec{r} - \vec{x}_n|}, \tag{5}$$

$$V_{\mu\nu}^{\text{III}} = - \int d\vec{r} \phi_{\mu}(\vec{r}) \phi_{\nu}(\vec{r}) v_{\text{III}}(\vec{r}). \tag{6}$$

Using Equations 5 and 6 and the usual definition of the exchange-correlation matrix, $V_{\mu\nu}^{\text{XC}}$, the effective Kohn-Sham matrix is given as

$$F_{\mu\nu} = T_{\mu\nu} + V_{\mu\nu} + J_{\mu\nu} + V_{\mu\nu}^{\text{XC}} + V_{\mu\nu}^{\text{II}} + V_{\mu\nu}^{\text{III}}. \tag{7}$$

A standard SCF procedure is then utilized to solve for the density matrix $D_{\mu\nu}$ and thus evaluate the QM/MM electronic energy, Equation 4. In terms of software implementation, the

matrix $V_{\mu\nu}^{\text{II}}$ (as well as any required gradients, Section II B) is constructed analytically with standard integral libraries, as either implemented or readily feasible within most quantum chemistry software. Implementation of integrals over arbitrary potentials, as in $V_{\mu\nu}^{\text{III}}$, can be generally done utilizing numerical quadrature. In our PyDFT-QMMM implementation, we evaluate $V_{\mu\nu}^{\text{III}}$ numerically within the DFT module of Psi4, utilizing the intrinsic DFT quadrature.⁹²

Long-Range v_{III} Electrostatic Interaction. In Equation 4, $v_{\text{III}}(\vec{r})$ is the long-range electrostatic potential from the tertiary (III) subsystem. For an arbitrary position, $\vec{r}$, the I-III potential is

$$v_{\text{III}}(\vec{r}) = v_{\text{U}}^{\text{recip}}(\vec{r}) - \sum_i^{N_{\text{I}}} \frac{q_i \text{erf}(\beta|\vec{r} - \vec{x}_i|)}{|\vec{r} - \vec{x}_i|} - \sum_n^{N_{\text{II}}} \frac{q_n \text{erf}(\beta|\vec{r} - \vec{x}_n|)}{|\vec{r} - \vec{x}_n|}, \quad (8)$$

where $v_{\text{U}}^{\text{recip}}(\vec{r})$ corresponds to the electrostatic potential arising from sums over all point charges in the “universe”, U (Figure 1), and the second and third terms correspond to exclusions from regions I and II. Since we specifically consider the case in which the “universe”, U , corresponds to an infinite system modeled with PBC, $v_{\text{U}}^{\text{recip}}(\vec{r})$ is computed in Fourier/reciprocal space exploiting the periodicity of the system. We have previously described such an approach, QM/MM/PME, in which $v_{\text{U}}^{\text{recip}}(\vec{r})$ is computed utilizing particle-mesh Ewald (PME) algorithms in a computationally efficient and straightforward way.⁹² In QM/MM/PME, $v_{\text{U}}^{\text{recip}}(\vec{r})$ corresponds directly to $v_{\text{MM}}^{\text{recip}}(\mathbf{r})$ defined in Equation 11 of reference ⁹². The QM/MM/PME method is thus defined by the QM/MM energy expression Equation 4, which invokes Equation 11 of reference ⁹² to define/compute $v_{\text{U}}^{\text{recip}}(\vec{r})$ and thus $v_{\text{III}}(\vec{r})$.

While the reader is referred to reference ⁹² for complete details of the QM/MM/PME approach, we briefly summarize the major points here. With PME algorithms, $v_{\text{U}}^{\text{recip}}(\vec{r})$ is computed efficiently for real-space positions that correspond to the PME grid points. Because numerical evaluation of the one-electron integrals involving $v_{\text{III}}(\vec{r})$ in Equation 4 is done utilizing the intrinsic DFT quadrature, $v_{\text{U}}^{\text{recip}}(\vec{r})$ must be interpolated from the PME grid to the DFT quadrature grid points. Interpolation must also be done to evaluate $v_{\text{U}}^{\text{recip}}(\vec{r})$ at the nuclear positions for region I atoms, required for the nuclear contribution to I-III interaction (Equation 4). In reference ⁹² it was demonstrated that high numerical accuracy is achieved with simple trilinear interpolation, with standard (or slightly more dense) choices of the PME and DFT quadrature grids. Furthermore, trilinear interpolation enables straightforward evaluation of gradients of terms involving $v_{\text{III}}(\vec{r})$, so that atomic forces are accessible

in the QM/MM/PME approach (Section II B). Once $v_{\text{U}}^{\text{recip}}(\vec{r})$ is computed and interpolated onto the DFT quadrature grid, exclusions from atoms in regions I and II (Figure 1) must be applied to convert $v_{\text{U}}^{\text{recip}}(\vec{r})$ to the I-III potential, $v_{\text{III}}(\vec{r})$ (Equation 8). The second term in Equation 8 represents exclusions added for atoms in region I with partial charges “ q_i ”, and the third term represents exclusions added for atoms in region II with partial charges “ q_n ”. Here “ β ” is the Gaussian width parameter utilized in the PME calculation of $v_{\text{U}}^{\text{recip}}(\vec{r})$, and “erf” denotes the error function; the error function arises because $v_{\text{U}}^{\text{recip}}(\vec{r})$ is computed with Gaussian charge distributions, as is typical in Ewald approaches.¹⁸⁹ In practice, exclusions in Equation 8 are computed specifically for each DFT quadrature grid point and nuclear position required in evaluation of $v_{\text{III}}(\vec{r})$ containing energy terms (Equation 4); this is accomplished in PyDFT-QMMM at the python level utilizing vectorized NumPy array broadcasting operations.

We briefly comment on the partial charges “ q_i ” on QM atoms in region I for the exclusions added in Equation 8. In the QM/MM/PME method, the q_i are static charges as defined on region I atoms when constructing the forcefield defining MM interactions. Specifically, the q_i are separate from and not derived from the QM electron density resulting from the DFT-based SCF. It is important to realize that in many applications, definition of static charges q_i on QM atoms will inevitably be unphysical. For example, if QM/MM/PME is utilized for molecular dynamics simulation of a chemical reaction (e.g. with biased/enhanced sampling), static charges “ q_i ” clearly cannot be a good representation of simultaneously the reactant, product, and transition state of the reaction. In this regard, it should be clarified exactly how the charges q_i affect the QM/MM energy and forces. First, the static charges q_i dictate the long-range electrostatic interaction between the QM atoms in region I (described by its DFT computed electron density) and periodic replicas of the QM atoms (described by q_i) outside of the primary simulation cell, i.e. region III. Such interactions between periodic replicas of the QM region are not expected to be important as they are very long range for standard simulation boxes, and will be screened by the solvent; hence unphysical, static q_i should be inconsequential in this regard.⁸⁶ The second effect is that forces on region III “MM” atoms from the I-III electrostatic interaction are typically computed with the static q_i charges (Section II B). For a long cutoff distance (e.g. 12-14 Å) defining region III (Figure 1), these forces are expected to be relatively small and any force errors due to unphysical q_i may be acceptable (since region III is far away from the chemistry of focus in region I).⁸⁶ A

more detailed discussion of force treatments/approximations in various QM/MM schemes is given in Section II B.

Finally, we discuss double-counting issues with electrostatic interactions that need to be corrected given usual PME/Ewald treatments in the MM Hamiltonian evaluation. The MM energy evaluation involves a PME/Ewald calculation of the electrostatic energy of the entire system, or “universe” $\mathbb{U}$, which we denote $E_{\mathbb{U}}^{\text{MM,elst}}$. Because electrostatic interactions between regions I-II and I-III are computed utilizing the QM electron density in Equation 4, the MM equivalent of these interactions must be subtracted out from $E_{\mathbb{U}}^{\text{MM,elst}}$ to avoid double counting. The electrostatic energy contained in $H_{\text{II,III}}^{\text{MM}}$ of Equation 1 then corresponds to

$$\begin{aligned} E_{\text{II,III}}^{\text{MM,elst}} &= E_{\mathbb{U}}^{\text{MM,elst}} - E_{\text{I-II}}^{\text{MM,elst}} - E_{\text{I-III}}^{\text{MM,elst}} \\ &= E_{\mathbb{U}}^{\text{MM,elst}} - \sum_i^{N_{\text{I}}} \sum_n^{N_{\text{II}}} \frac{q_i q_n}{|\vec{x}_i - \vec{x}_n|} - \sum_i^{N_{\text{I}}} q_i v_{\text{III}}(\vec{x}_i). \end{aligned} \quad (9)$$

Here it is assumed that electrostatic interactions within region I, e.g. I-I interactions, have been excluded in the MM PME/Ewald calculation and are not present in $E_{\mathbb{U}}^{\text{MM,elst}}$. The notation in Equation 9 is identical to that defined and discussed previously. In our PyDFT-QMMM implementation, the second term (region I-II double counting correction) is evaluated with the MM software (OpenMM), with implementation details described in Section III. The third term (region I-III double counting correction) is easily evaluated at the Python level once $v_{\text{III}}(\vec{x}_i)$ is computed at all of the nuclear/partial charge positions for atoms in region I. As discussed previously, interpolation of v_{III} from the PME grid to region I nuclear coordinates was already done in evaluating term(s) in Equation 4, so no additional interpolation is required. While force evaluation is discussed in Section II B, we briefly discuss how forces are handled with regard to Equation 9. In principle, there should be no force contribution from the double-counting correction terms in Equation 9, since they simply subtract out energy contributions from $E_{\mathbb{U}}^{\text{MM,elst}}$. However, this requires an MM software/interface that enables separation of electrostatic force terms resulting from $E_{\mathbb{U}}^{\text{MM,elst}}$ that can simply be neglected. In our PyDFT-QMMM implementation this is the case, as OpenMM enables separation of force terms as discussed in Section III, so no forces are added from gradients of the double counting corrections. For an alternative MM software/interface in which such force separation isn't possible, one would have to do careful bookkeeping to ensure that the force terms from double counting corrections in Equation 9 appropriately cancel.

B. QM/MM Forces

Nuclear/atomic forces derived from the QM/MM Hamiltonian are required for many applications, such as molecular dynamics simulations or geometry optimizations. Force expressions are derived by taking the negative gradient of the QM/MM energy, but the gradients will differ depending on whether the gradient coordinate corresponds to the primary (I), secondary (II), or tertiary (III) subsystem. In principle, all such gradients are needed for molecular dynamics simulations or full (i.e. unconstrained) geometry optimizations. The bookkeeping of these different gradient terms is often not explicitly discussed in the PBC QM/MM literature, even though it can be subtle depending on the level of theory used to treat the different subsystems/regions. To aid in explicit bookkeeping of the different gradient terms (required for forces), we propose the following “force matrix” expression,

$$\begin{aligned} & - \nabla_{\text{I}} \cdot \begin{bmatrix} E_{\text{I-I}} & E_{\text{I-II}} & E_{\text{I-III}} \end{bmatrix} \\ & - \nabla_{\text{II}} \cdot \begin{bmatrix} E_{\text{I-II}} & E_{\text{II-II}} & E_{\text{II-III}} \end{bmatrix} \\ & - \nabla_{\text{III}} \cdot \begin{bmatrix} E_{\text{I-III}} & E_{\text{II-III}} & E_{\text{III-III}} \end{bmatrix} \end{aligned} = \begin{bmatrix} \vec{F}_{\text{I-I}} & \vec{F}_{\text{I-II}} & \vec{F}_{\text{I-III}} \\ \vec{F}_{\text{II-I}} & \vec{F}_{\text{II-II}} & \vec{F}_{\text{II-III}} \\ \vec{F}_{\text{III-I}} & \vec{F}_{\text{III-II}} & \vec{F}_{\text{III-III}} \end{bmatrix} \Rightarrow \begin{bmatrix} \text{QM} & \text{QM/MM} & \text{X} \\ \text{QM/MM} & \text{MM} & \text{MM} \\ \text{Y} & \text{MM} & \text{MM} \end{bmatrix}, \quad (10)$$

where the first matrix, the “energy matrix,” describes the symmetric intra- and inter-subsystem interaction energies. Each row of the energy matrix is multiplied by a set of gradients corresponding to a subsystem, such as ∇_{I} for gradients taken with respect to members of the primary subsystem. This operation yields the force matrix, where $\vec{F}_{\text{I-II}}$ represents the contribution of the interaction with the secondary subsystem on the force acting on the primary subsystem and $\vec{F}_{\text{II-I}}$ represents the (theoretically) equal and opposite contribution of the interaction to the force acting on the secondary subsystem. The third matrix is an abstraction of the force matrix which contains the levels of theory applied to calculate each force contribution. This abstract force matrix follows the form of a block matrix: the intra-subsystem force contribution for the primary subsystem is treated with the QM level of theory, and the intra- and inter-subsystem force contributions for the secondary and tertiary subsystems are treated with the MM level of theory. The inter-subsystem forces for the primary and secondary subsystems, labeled “QM/MM,” follow the electrostatic embedding scheme described by Equation 2, where the MM force contributions are defined by the forcefield and the QM force contributions are derived from analytic gradients of the energy

with respect to nuclei and embedded point charges. The QM force contributions for the I-II interaction are further discussed in Appendix A.

Method	X	Y	Reference
Additive QM/MM	0	0	¹⁴⁰ *
QM/MM/Cutoff	0	MM	This work
QM/MM+PME	MM	MM	⁹¹
QM/MM/PME	QM	MM	⁸⁶ , This work
QM/MM/SC-PME	QM	QM	-

TABLE I. QM/MM method classification by level of theory of I-III force terms

* Neglecting I-III interactions by additive QM/MM typically involves having no forcefield representation of the charge density of region I. Reference ¹⁴⁰ uses a reaction field method to represent long-range electrostatics in regions II and III within the principal cell. In a simulation using Ewald or PME summation, no forcefield charges on region I implies that region III in the principal cell experiences a charge cavity from region I, and that regions II and III experience periodic charge cavities in the mirror images of region I.

The force contributions arising from the interaction between the primary and tertiary subsystems warrant further discussion. Varying treatments of X, the force acting on the primary subsystem from the I-III interaction, and Y, the force acting on the tertiary subsystem from the I-III interaction, are denoted by specific QM/MM nomenclature as presented in Table I. With the nomenclature in Table I, we have tried to be consistent with the most standard usage of these terms in the literature; inevitably, some authors may use different nomenclature than what we employ here. The “Additive QM/MM” approach does not account for interactions between the primary and tertiary subsystems in any way, when computing either energies or forces. This approach avoids parameterization of forcefield charges for the primary subsystem,^{17,190} which may be convenient for solutes undergoing significant charge redistribution or chemical reactions. The “QM/MM/Cutoff” approach utilizes forcefield charges on the primary subsystem for propagating forces on the tertiary subsystem, and for incorporating the I-III interaction energy at a fully MM level, but the reciprocal force on the primary subsystem is generally not included. The reciprocal force on the primary subsystem from the charge/MM model is included in so-called “QM/MM+PME” approaches,

in which the long-range electrostatics computed from a PME/Ewald sum are mechanically embedded.

Our direct QM/MM/PME approach,⁹² as well as similar variants,⁸⁶ allows the electron density of the primary subsystem to be polarized by the static charge distribution of the tertiary subsystem. QM forces from the v_{III} potential (Equation 8) acting on the primary subsystem are derived in Appendix A, giving the “X=QM” contribution in Table I, and Equation 10. Due to practical considerations, the reciprocal force on the tertiary subsystem is computed at the MM level, i.e. “Y=MM” utilizing the forcefield charges of the primary subsystem. The practical difficulty is that computing these forces from the QM electron density (i.e. “Y=QM”) would require a very dense FFT/PME grid for numerical solution of the Poisson equation utilizing the QM electron density in combination with PBC. In practice, a fully “self-consistent” approach termed “QM/MM/SC-PME”, in which the QM electron density is utilized to compute all relevant electrostatic forces (i.e. “Y=QM”) in a self-consistent manner, is possible if core electrons are described by pseudopotentials, and only valence electrons are treated explicitly.⁶⁵ This is because for valence electrons only, a much coarser (and more standard) FFT grid can be utilized for the QM electron density in both the PME calculation and subsequent numerical quadratures. Such a “QM/MM/SC-PME” approach will necessarily provide a more accurate treatment of the forces on the tertiary subsystem from the I-III interaction, compared to our “QM/MM/PME” method. However, it is important to keep in mind the primary “goals” of a QM/MM calculation, in which the focus is typically on a chemical process occurring within the primary/QM subsystem. Due to the long interaction range, forces on region III atoms resulting from the I-III electrostatic interaction are expected to be relatively small and any associated force errors on region III atoms due to unphysical region I forcefield charges (previously denoted q_i) may be acceptable. This may be particularly true when a thermostat is applied in an MD simulation (e.g. NVT ensemble), such that stochastic forces modify the true Newtonian dynamics anyway. Regardless, our focus on the “QM/MM/PME” method rather than (a more complex) “QM/MM/SC-PME” approach is motivated by the practical numerical and software challenges discussed.

It is clear that several of the QM/MM approaches listed in Table I violate Newton’s third law for I-III force terms, and furthermore some of the force terms do not correspond to the negative gradient of the QM/MM energy (i.e. are not conservative). For example, this is

the case for QM/MM/PME, in which the energy expression Equation 4 incorporates a full QM treatment of I-III electrostatic interactions, yet the force term $\vec{F}_{\text{III-I}}$ in Equation 10 is computed from an alternate energy expression at the MM level (e.g. “Y=MM”). Clearly then, molecular dynamics simulations utilizing these force expressions will not rigorously conserve energy, and the quality of energy conservation or energy drift due to a particular QM/MM convention will have to be numerically evaluated. Energy drift for both the “QM/MM/PME” and “QM/MM/Cutoff” approaches will be discussed in Section IV. For now, we will simply note that classical molecular dynamics simulations employing forcefields commonly do not rigorously conserve energy either, due to use of truncations or cutoffs e.g. in computing VDWs interactions, lack of fully self consistent induced dipoles in polarizable forcefields, and/or other numerical effects. The extent of energy drift that can be tolerated for physically valid predictions (e.g. ensemble averages) is a quite subtle and complex question, but it is clear that there are many practically useful MD approaches/implementations that do not obey rigorous energy conservation.

Having established a description of the energy and forces arising from the I-III interaction, we outline abstracted, sequential algorithms for applying different combinations of I-II and I-III embedding schemes in PBC QM/MM in Table II that the MD software must be able to perform. The rows correspond to different I-III embedding procedures, and the columns correspond to different I-II embedding procedures. Note that the approaches discussed in Table I assume a I-II electrostatic embedding scheme, and so the rows of the final column of Table II correspond to the Additive QM/MM, QM/MM+PME, QM/MM/Cutoff, and our direct QM/MM/PME approaches, respectively. Table II follows the form of an upper triangular matrix when embedding schemes are sorted by level of sophistication along the axis because we assume that it is reasonable to select a I-III embedding scheme at a comparable or lower level of sophistication to that of the I-II scheme. When energies are added, subtracted, or “zeroed” in Table II, this is also propagated to the gradients of the subsystems involved in the interaction (with the exception of $E_{\text{I-III}}^{\text{elst}}$ subtraction, which is needed to avoid double counting interaction energies in the QM/MM/PME). Several schemes on the bottom half of Table II also require that only the forces on subsystem I are zeroed so that the corresponding force on III and energy from the I-III interaction is maintained at the MM level, while other force contributions to subsystem I at the MM level, like $\vec{F}_{\text{I-II}}^{\text{non-elst}}$ or $\vec{F}_{\text{I-II}}^{\text{elst}}$, must be added back onto the primary subsystem forces to account for the truncated

I-III Ixn \ I-II Ixn	None	Mechanical Embedding	Electrostatic Embedding
	None	Mechanical Embedding	Electrostatic Embedding
None	1. Zero E_{I-I} 2. Zero $E_{I-II,III}^{\text{elst}}$ 3. Zero $E_{I-II,III}^{\text{non-elst}}$	1. Zero E_{I-I} 2. Zero $E_{I-II,III}^{\text{elst}}$ 3. Add E_{I-II}^{elst}	1. Zero E_{I-I} 2. Zero $E_{I-II,III}^{\text{elst}}$ *Additive QM/MM
Mechanical Embedding		1. Zero E_{I-I}	1. Zero E_{I-I} 2. Subtract E_{I-II}^{elst} *QM/MM+PME
“Cutoff” Embedding		1. Zero E_{I-I} 2. Zero $\vec{F}$ on I 3. Add $\vec{F}_{I-II}^{\text{elst}}$ on I 4. Add $\vec{F}_{I-II}^{\text{non-elst}}$ on I	1. Zero E_{I-I} 2. Subtract E_{I-II}^{elst} 3. Zero $\vec{F}$ on I 4. Add $\vec{F}_{I-II}^{\text{non-elst}}$ on I *QM/MM/Cutoff
Electrostatic Embedding			1. Zero E_{I-I} 2. Subtract E_{I-II}^{elst} 3. Zero $\vec{F}$ on I 4. Add $\vec{F}_{I-II}^{\text{non-elst}}$ on I 5. Subtract E_{I-III}^{elst} *QM/MM/PME

TABLE II. Bookkeeping of MM energy and force terms for the different combinations of embedding schemes

interaction. In order to accommodate these different embedding approaches in a straightforward way, the MD software needs to be able to separate out different energy and force contributions. It is worth noting that many of the operations in Table II are comparable and may be able to be generalized in implementation. We expand on the realization of these embedding procedures in the PyDFT-QMMM package in Section III.

C. Additional Considerations

We have discussed approaches and algorithms for QM/MM employing electrostatic embedding within PBC, but we note that further considerations often arise in specialized QM/MM applications. Several important topics within QM/MM methods that have not been discussed here include: 1) polarizable embedding, 2) bond-capping or link atoms, and 3) adaptive partitioning schemes. Polarizable embedding allows for mutual polarization between primary and secondary subsystems through fluctuating charge, fluctuating dipole, many-body expanded, or drude/shell models.^{23,191–201} Such approaches may require additional or specialized SCF procedures in order to variationally optimize the mutual polarization of both the primary and secondary subsystems. Interactions between subsystems can be further refined by considering charge transfer between the primary and secondary subsystems through flexible embedding,⁴⁵ though charge-transfer interactions are typically neglected²³ for a well-chosen primary subsystem. A common complication arises when the partition boundary between the primary (QM) and secondary (MM) subsystems bisects a covalent bond. This often occurs in QM/MM applications to enzymes and proteins, materials^{202–204} and catalysis^{16,205}, and in general when a chemically active molecule/macro-molecule is too large to fully treat at a QM level. In such cases, the “dangling bond” between the primary and secondary subsystem may be treated with one of several boundary schemes, including frozen orbitals,^{99,110,206–213} capping or group potentials,^{214–225} or link atoms^{71,100,190,205,226–228} and their more sophisticated variations.^{203,204,229–240} Hydrogen link atom treatments are among the most common boundary schemes,^{16,205} but care must be taken to minimize energy and force artifacts¹⁹⁰ and ensure correct electron polarization near the boundary.²³⁸

In systems where solvent molecules participates in the essential chemistry of interest, the exact subsystem partition may be dynamic and thus difficult to determine and define. A concrete example is a hydronium ion in water, for which the excess proton shuttles between numerous water molecules with constantly changing bond topology.²⁴¹ In such cases, it is infeasible to predict which solvent molecules will participate in reactivity *a priori*, so alternative partitioning schemes are necessary. Constrained partitioning schemes prevent molecules from diffusing into or out of a defined primary subsystem, either through a well-parameterized biasing potential,^{242–247} or through scattering at the subsystem

boundary.²⁴⁸ Adaptive partitioning schemes, on the other hand, allow molecules to diffuse into and out of the primary subsystem region, leading to a dynamic definition of subsystem membership.^{46,123,241,249–274} Adaptive partitioning approaches determine primary subsystem membership based on criteria such as spatial distance,^{241,249–253,256,259–261,264,265,267,271,272} temporal distance,²⁶⁸ coordination,^{254,263} or interaction^{255,266,269} with an “active center,” which is typically defined *a priori* or determined through system properties.^{270,273,274} In practice, the application of finite biasing potentials in constrained schemes may require small time steps^{242,243} and can lead to structural and statistical artifacts due to incomplete separation,^{245,247,248,275} and adaptive schemes generally require multiple QM/MM evaluations during a single step to capture a smooth transition between QM and MM potentials.^{251,258,275}

III. IMPLEMENTATION

PyDFT-QMMM is a Python package for implementing QM/MM approaches, comprising a lightweight library of core capabilities and a pluggable architecture. A list of important features for both users and developers and a sample script is provided:

1. **Simple installation.** The package has default interfaces to Psi4 ≥ 1.9 and OpenMM ≥ 8.0 , which may be installed, along with the PyDFT-QMMM, through Python’s `pip` protocol (no compilation is required). Use of the QM/MM/PME approach currently requires customized versions of OpenMM and Psi4,⁹² but work is underway to allow this approach to run with the standard distributions.
2. **Pluggable architecture.** Custom behavior may be injected into energy/force calculation and integration routines through a decorator design pattern²⁷⁶ that modifies the `calculate` and `integrate` methods.
3. **Choice of embedding schemes.** Arbitrary pairings of short-range (I-II) and long-range (I-III) embedding schemes, as in Table II of Section II B, can be selected through the high-level API.
4. **Complex workflows through scripting.** The package focuses on the facile development of QM/MM simulation workflows through its Python API, where Python scripts take the place of input files, as in Code Example 1; however, we also provide a

basic commandline interface (CLI) with reduced functionality, allowing for batch job submission with input files in the *.ini configuration file format.

5. **Internal type-safety.** The package includes full Python type annotations, promoting type-safety during development and partly serving as code documentation, in addition to docstrings and comments.

CODE EXAMPLE 1. A Basic Workflow

```

1 from pydft_qmmm import *
2 from pydft_qmmm.plugins import SETTLE
3
4 # Provide directories to PDB and topology/forcefield
5 # XML files to generate the system.
6 system = System(...)
7
8 # Define Hamiltonians to apply to the energy and
9 # force evaluations
10 H_QM = QMHamiltonian(...)
11 H_MM = MMHamiltonian(...)
12 H_QMMM = QMMMHamiltonian(...)
13
14 # Define the first 3 atoms of the system as
15 # belonging to the primary subsystem and define
16 # the remainder as belonging to the secondary and
17 # tertiary subsystems.
18 H_tot = H_QM[0:3] + H_MM[3:] + H_QMMM
19
20 # Define the integrator to apply in generating
21 # dynamics.
22 integrator = Verlet(...)
23
24 # Instantiate a plugin. In this case, the
25 # the plugin applies the SETTLE algorithm on
26 # water residues during the integration routine.
27 settle = SETTLE(...)
28
29 # Provide preceding objects to the simulation
30 # object.
31 simulation = Simulation(
32     system,
33     H_tot,
34     integrator,
35     plugins=[settle],
36     ...,
37 )
38
39 # Run dynamics for 10 timesteps whose length was
40 # provided to the integrator object
41 simulation.run_dynamics(10)

```

The primary concern of the package is to ensure that QM/MM embeddings are performed in a manner that is consistent with user specifications. Following the approach of OpenMM,¹⁸⁶ PyDFT-QMMM provides several layers of abstraction to ensure a robust translation of the physics specified by the user into the myriad details of implementation. While the abstractions in OpenMM are introduced for the sake of performance across hardware platforms,¹⁸⁶ our abstractions serve to organize the embedding procedure across QC and MD packages during computation, which could be any one of those presented in Table II of Section II B. This “implementation-hiding” approach is informed by the object-oriented programming paradigm and allows the user to select arbitrary embedding schemes without needing to consider all details of the underlying implementation. In this regard, PyDFT-QMMM abstracts QM/MM algorithms in a manner that is comparable to Janus, for example;¹⁶⁷ however, PyDFT-QMMM also maintains an internal representation of the state of the system, like ASE or OpenMM,^{177,186} and internal molecular dynamics integrator routines are provided in the core library. Accordingly, PyDFT-QMMM is also a simulation engine. The QM/MM calculation and force integration routines serve as entry-points for users to inject custom behavior into the program through plug-ins, e.g. modifications to the partitioning scheme, *post hoc* corrections to energy and force calculations, or the maintenance of constraints during integration. To facilitate the array of QM/MM embedding, calculation, and simulation tasks, the classes of the PyDFT-QMMM package are structured to appropriately separate responsibilities in accordance with object-oriented principles,^{277,278} minimizing coupling between internal data structures and maximizing the reusability of the code. A simplified Unified Modeling Language (UML) representation of the package class structure is provided in Figure 2.

The central **Simulation** object is effectively a wrapper that organizes the operations performed during molecular dynamics simulation. The MD simulation protocol involves computing forces from the QM/MM Hamiltonian for given system coordinates, and then integrating the Newtonian equations of motion to propagate the dynamical trajectory possibly utilizing thermostat(s), constraints, biasing potentials, etc. These processes inherently have low coupling, and so these tasks are represented through separate objects: the **Calculator** object and the **Integrator** object, respectively. Additional QM/MM simulation tasks, including dynamic system partitioning, link-atom placements, and the maintenance of geometry constraints, are orchestrated by QM/MM software before, between, or after the

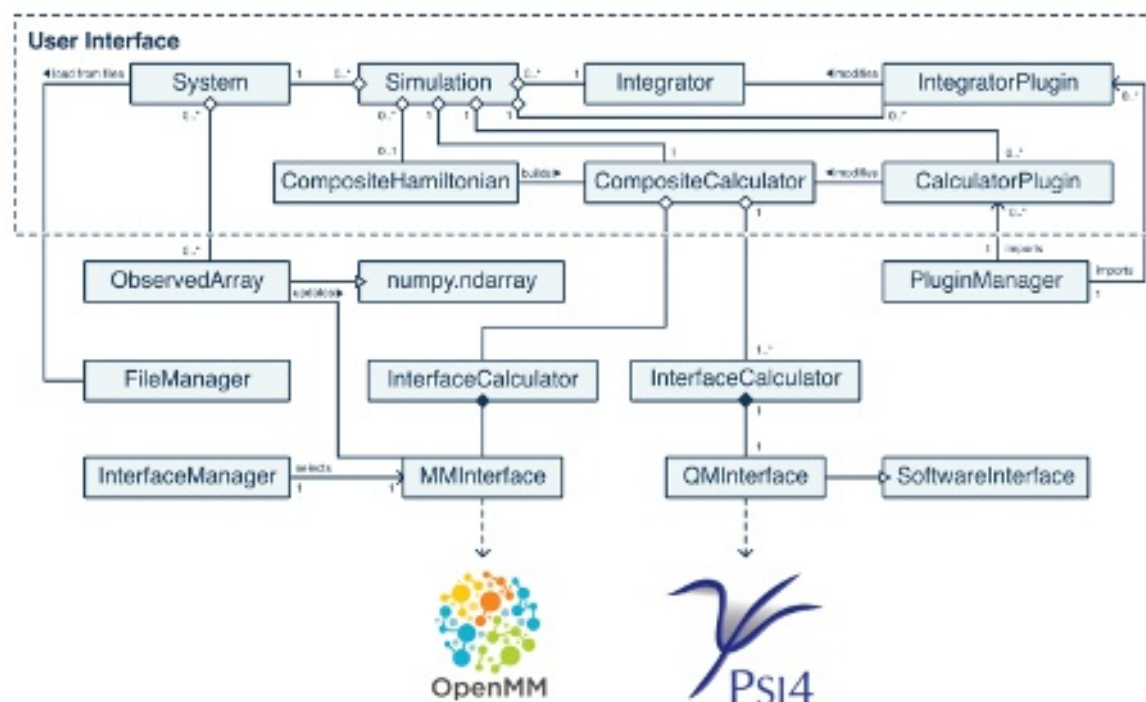


FIG. 2. The class structure of PyDFT-QMMM.

fundamental force calculations and integration steps. Accordingly, PyDFT-QMMM allows the behavior of calculation or integration routines to be readily modified or extended on-the-fly with **Plugin** objects. Plugins are applied using a decorator design pattern²⁷⁶ to inject functionality, and multiple plugins may be used sequentially to modify **Calculator** or **Integrator** behavior in a predictable manner. Several plugins are provided in the base package, for altering the subsystem partitioning scheme or implementing SETTLE or frozen constraints. Additionally, a separate PLUMED²⁷⁹ plugin is available for installation, which enables access to a comprehensive suite of biased-sampling approaches that are utilized for free energy simulations within QM/MM (or other) MD approaches. The **PluginManager** searches the Python environment for packages containing metadata for the PyDFT-QMMM plugin entry point, allowing for dynamic loading of local or third-party plugins with no additional installation beyond what is required to install a Python package. We provide templates for developers to write third-party plugins.

The application of various embedding schemes and the versatility afforded by the plug-in architecture are facilitated through several layers of abstraction. The abstracted **Hamiltonian** class stores settings for the external QC and MD packages, implemented in

the concrete `QMHamiltonian` and `MMHamiltonian` classes, respectively. Subsets of atoms are assigned to be represented by the `QMHamiltonian` or `MMHamiltonian` objects, defining the primary, and secondary and tertiary subsystems. The interactions between subsystems are represented by a coupling `Hamiltonian` class, `QMMMHamiltonian`, which configures the level of theory to be applied in energy and force calculations associated with interactions according to the force matrix presented in Equation 10. The `Hamiltonian` classes are independent of specific instances of the `System` or `Simulation` classes, and so they can be applied independently to multiple `Simulation` objects regardless of the concrete system under study—the correct subsystem membership need only be applied to the `Hamiltonian` for each system. The total QM/MM `Hamiltonian` can then be constructed in a manner analogous to the canonical form of Equation 1, as in Code Example 1. The `Calculator` is an intermediate level of abstraction between the Python API and the QC and MD packages. The `Calculator` object might be: a standalone `InterfaceCalculator` for pure QC or MD calculations; a `CompositeCalculator` to collate calculations from aggregated `InterfaceCalculator` objects; or a user-defined `Calculator` derived class. In addition to managing distinct types of calculation, the `Calculator` abstraction also serves as an entry-point within the plug-in architecture.

The `SoftwareInterface` class underpins the energy and force calculation procedures, mediating communication between PyDFT-QMMM and objects of external QC and MD packages constructed with data provided by the `Hamiltonian` and `System` objects. PyDFT-QMMM currently provides default interfaces to Psi4 ≥ 1.9 and OpenMM ≥ 8.0 through their native Python APIs,^{179,181} and interfaces to modified versions of Psi4 and OpenMM implementing our QM/MM/PME approach are also provided.⁹² In contrast to prior approaches,^{121,168,177} we do not adapt the `SoftwareInterface` to handle different forms of communication between packages with varying degrees of source availability at the expense of QM/MM methodological diversity. In addition to avoiding the latency incurred by file-based and inter-process communication,^{17,167,171} in-process communication through Python API allows for interaction with the internal data structures of the QC and MD programs at runtime. Many well-implemented APIs provide bindings for a variety of low-level utilities,^{178,179,182,188} allowing facile extension of the software to QM/MM embedding procedures. For example, versatile energy and force decomposition is a convenient and powerful feature of OpenMM that allows PyDFT-QMMM to implement the various embedding procedures.

The general bookkeeping of the various QM/MM energy and force terms for the different embedding choices in Table II is a major focus of our PyDFT-QMMM software. The MM contributions listed in Table II are largely related, and are thus evaluated using several general functions defined in the `MMInterface` class. We emphasize encapsulation in our implementation, requiring in the `MMInterface` class definition that interfaced MD programs provide the utilities for removing doubly-counted interactions. For example, removal of MM contributions to $E_{\text{I-I}}$ in the `OpenMMInterface` involves setting “exclusions” for bonded interactions and “exceptions” for non-bonded interactions for all atoms within the primary subsystem. This operation is common to all combinations of embedding procedures in Table II. The additive QM/MM schemes of the first row of Table II also variously require removal of non-bonded interactions $E_{\text{I-II,III}}^{\text{elst}}$ or $E_{\text{I-II,III}}^{\text{non-elst}}$ with the primary subsystem, which is accomplished by setting the region I charges and Lennard-Jones parameters to appropriately zero interactions and updating the `OpenMM Context` object.

Operations involving adding or subtracting $E_{\text{I-II}}^{\text{elst}}$ require further attention, since PyDFT-QMMM makes use of molecular centroid or “group-based” cutoffs by default for determining subsystem II membership. Accordingly, a separate `CustomNonbondedForce` is constructed to only represent I-II Coloumbic electrostatic interactions using the `NoCutoff` method, adding a binary parameter to the potential expression to dynamically zero I-III interactions with the `updateParametersInContext` method as the system evolves. The centroid of region I is kept at the center of the simulation domain to implicitly incorporate the minimum image convention in this group-based scheme. Because the standard treatment of OpenMM PBC electrostatics either incorporates a reaction field procedure in the `CutoffPeriodic` non-bonded method or Ewald summation in the `Ewald` and `PME` non-bonded methods, the I-II interaction modeled through a `CustomNonbondedForce` employing `NoCutoff` must be attached to a separate OpenMM `Context` object, called the `ixn_context` in the `OpenMMInterface` object of PyDFT-QMMM, and the base `Context` utilizing a PBC electrostatic approach is referred to as the `base_context`. Such a dual-`Context` approach requires that changes to system data be propagated to the two different OpenMM objects with some degree of overhead, though this overhead is negligible when compared with the QC calculation, and it also requires OpenMM to compute and remove doubly-counted interactions (e.g., subtracting $E_{\text{I-II}}^{\text{elst}}$). The forces and energies corresponding to $E_{\text{I-II}}^{\text{elst}}$ in the `ixn_context` can then be added or subtracted with those calculated in the `base_context`.

Our implementation also makes extensive use of the force group architecture of OpenMM, where different energy and force contributions can be calculated separately. This allows forces and energy to be calculated for one group of `Force` objects and only forces with no associated energy to be calculated for another force group, which is especially useful for the QM/MM/Cutoff and QM/MM/PME embedding procedures. The QM/MM/Cutoff schemes require that forces on region I from the I-III interaction are removed without affecting the associated interaction energy or reciprocal forces on region III, and this is accomplished through application of a masking array to zero specified forces. These procedures typically require that some component of the force be added back onto the region I, typically $\vec{F}_{\text{I-II}}^{\text{non-elst}}$ or, less commonly, $\vec{F}_{\text{I-II}}^{\text{elst}}$. These interactions are modeled with `CustomNonbondedForce` in the `ixn_context`, but because the interaction energy is already calculated in the `base_context`, only the forces are calculated for this interaction. For QM/MM/PME, subtracting the doubly-counted $E_{\text{I-III}}^{\text{elst}}$, as in Equation 9 and Table II, does not take place in the `SoftwareInterface` layer due to the processing required to construct v_{III} . With these routines and data structures, all of the procedures in Table II are implemented in PyDFT-QMMM.

One difficulty presented by using Python APIs is that objects from the external dependencies persist throughout a simulation, and so some mechanism is necessary to maintain a consistent representation of the state of the system across the external packages. In the approach taken by several packages, including Janus, this involves privileging the MD software, allowing it to store relevant data, propagate dynamics, and provide system data to other parts of the package. PyDFT-QMMM, on the other hand, maintains an internal representation of the system and enforces this representation across the external packages. State management is enforced through an observer design pattern,²⁷⁶ which registers `SoftwareInterface` objects to be notified when system data changes. While the observer design pattern²⁷⁶ is often used to report results of calculations to the user,^{125,177} we use this pattern to report to registered interfaces whenever pertinent system data is altered, either through internal routines, such as integration, or through on-the-fly user edits. All system data is stored in the `ObservedArray`, which sub-classes from the NumPy `ndarray`, adding functionality to register notification functions that will be called if any of the elements in the array are edited. In this way, all system data is stored centrally, and the user may interact with this data with all of the capabilities afforded to NumPy arrays.

All system data in `ObservedArray` objects originates from `System` object, which manages the internal representation of the system in PyDFT-QMMM. The most basic representation of an atomistic system is the `Atom` object—a derivative of the `dataclass` in the Python standard library—which merely stores data pertaining to a particular atom. This data comprises: vectors, including `position`, `velocity`, and `force`; scalars, including `mass` and `charge`; and topological information, including the two-character `element` symbol and four-character `name` of the atom, the index and four-character name of the molecule to which the atom belongs (compare with the “residue” of MD packages), and the `subsystem` membership of the atom. The vector and scalar quantities are relevant to atoms of any subsystem, though in the case of primary subsystems atoms, it would be more rigorous to say that vector quantities represent the position of, velocity of, and force acting on the nuclei. The net charge of primary subsystem atoms should not be confused with the charge chosen in the `QMHamiltonian`, though these values should be equal for rigorous embedding. Most of the topological information, outside of the subsystem designation, originate from the structure and conventions of the Protein Database (PDB) file. We note that no data is required to instantiate an `Atom` object, and the user may subsequently add or edit data. The user may define and append `Atom` objects to a `System` object, which behaves like a Python sequence. Alternatively, we provide utilities to load system data from PDB and forcefield extensible markup language (XML) files, and we plan to add support for additional formats. The `System` object concatenates data from `Atom` objects and stores them in buffers within `ObservedArray` objects. The `SoftwareInterface` objects register notifiers to relevant `ObservedArray` objects in the `System` object upon instantiation, tying each instance of external software to a single system. We note, however, that this does not preclude the `System` object from interaction with an arbitrary number of interfaces.

IV. EXAMPLES AND BENCHMARKS

We present several benchmarks to demonstrate the capabilities of the PyDFT-QMMM package, each of which makes use of a cubic simulation box of water, with a single water molecule chosen as the primary QM region, and the remaining water molecules constituting the secondary and tertiary (MM) subsystems. The systems consisted of either 895 or 7160 total water molecules. The water systems were equilibrated for 1 ns in the NPT ensemble at

300 K with the SPC/E forcefield to a final equilibrium box length of 29.9 Å (1.00 g cm^{-3}) and 59.9 Å (0.996 g cm^{-3}), respectively. The classical MD equilibrations were performed with OpenMM version 8.0 using a Monte Carlo barostat and Langevin integrator at 1 fs timestep. The thermostat used a friction coefficient of 1 ps^{-1} and the barostat set to 1 atm was applied every 100 steps. All DFT calculations within the QM/MM simulations were performed with the PBE functional and a Lebedev-Laikov quadrature of 302 spherical points and 75 radial points; unless otherwise stated, the def2-SVP basis set was employed.¹⁷⁹ The QM/MM/Cutoff simulations made use of a sharp 14.0 Å cutoff for electrostatic and mechanical embedding, and membership to the secondary subsystem was evaluated by comparing the distance between the centroid of the solute and the centroid of the solvent molecule against the cutoff distance. The QM/MM/PME calculations used a PME grid spacing of 1.0 Å and a Gaussian smearing coefficient of 5.0 nm^{-1} , with numerical accuracy of these settings benchmarked in our previous work;⁹² a similar cutoff distance (14.0 Å) was used to define the secondary and tertiary subsystem regions.

As our first benchmark, we test the gradients implemented in the PyDFT-QMMM software for both the QM/MM/Cutoff and QM/MM/PME approaches, as benchmarked against finite difference energy evaluation calculations. This benchmark serves three purposes: 1) to validate the PyDFT-QMMM software implementation; 2) to validate the correctness of formulas and implementation of QM/MM/PME force terms, which are derived (for the first time) in Appendix A; and 3) to compare fundamental differences between the QM/MM/Cutoff and QM/MM/PME approaches. For this benchmark, gas-phase DFT, QM/MM/Cutoff, and QM/MM/PME gradients were calculated for 10 different water molecules chosen to comprise the primary subsystem in-turn, within the 59.9 Å cubic box, comprising different solute orientations and solvent configurations. Gradients and corresponding finite-difference gradients were computed for all Cartesian coordinates of each atom within the primary/QM system; in this case, there are nine Cartesian coordinates for the three water atoms (QM), requiring 18 energy evaluations per configuration for the finite difference benchmarks. Each displacement utilized in the finite difference benchmark was 0.0025 Å, which is comparable to the default displacement of 0.005 atomic units (a.u.) in the Psi4 numerical gradient implementation.¹⁸⁷ Gradient calculations were performed with energy and density convergence thresholds of $1\text{e} - 8 \text{ a.u.}$, and energy calculations were performed with energy and density convergence thresholds of $1\text{e} - 6 \text{ a.u.}$ Finite differences were then converted to gra-

dients through central difference, and compared as benchmarks against the direct gradient calculation.

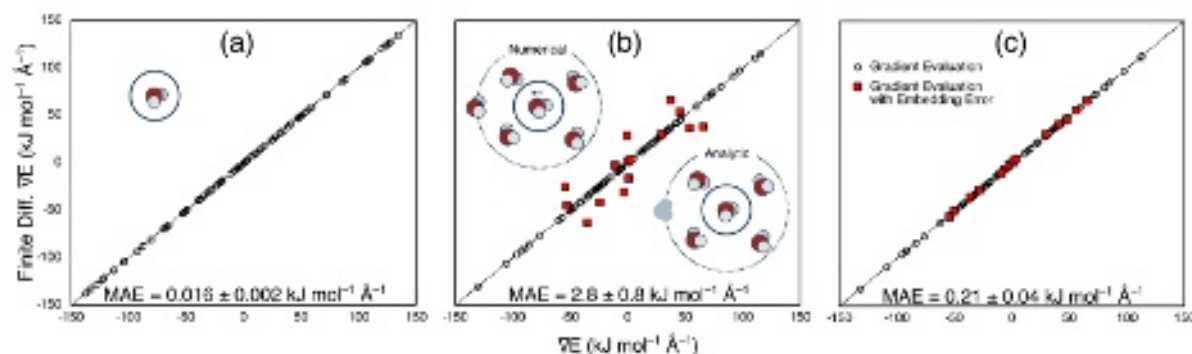


FIG. 3. Parity plots for the QM gradient vs finite difference benchmarks of the primary subsystem (QM water molecule) shown for 10 solute/solvent configurations for a) gas-phase DFT, b) QM/MM/Cutoff, and c) QM/MM/PME approaches. Cartesian gradients for both oxygen and hydrogen atoms are aggregated on the plot. The schematic inset in b) depicts how electrostatic embedding environments may differ in the gradient vs finite-difference evaluation, leading to outliers with significant numerical error (red data points). A back of the envelope estimate indicates that this scenario is not uncommon, e.g. $4\pi (14 \text{ \AA})^2 * 0.0025 \text{ \AA} * 0.033 \frac{\text{H}_2\text{O molecules}}{\text{\AA}^3} \approx 0.2 \text{ H}_2\text{O molecules}$.

Parity plots comparing the direct gradient evaluation vs finite difference benchmarks, for the gas-phase DFT, QM/MM/Cutoff, and QM/MM/PME approaches are shown in Figure 3. The mean absolute error (MAE) of the gradients span two orders of magnitude across the different methods. As a reference, the MAE for the gas-phase DFT gradient calculation compared to finite difference is $0.016 \pm 0.002 \text{ kJ mol}^{-1} \text{ \AA}^{-1}$ (Figure 3a) which represents the “best case” agreement (i.e. QM only). For the QM/MM approaches, we first discuss the gradient benchmarks for “QM/MM/Cutoff”, as shown in Figure 3b. The MAE for the gradients vs finite-difference for QM/MM/Cutoff is substantially larger, $2.8 \pm 0.8 \text{ kJ mol}^{-1} \text{ \AA}^{-1}$, but as evident in Figure 3b the clear culprit for the substantially larger MAE are the “outlier” points highlighted in red. The substantial error in these outlier points results from an intrinsic aspect of the finite difference procedure, related to selection of MM atoms for electrostatic embedding. For the electrostatic embedding procedure, water molecules within the secondary subsystem are included in the embedding if their centroid is within $14.0 \text{ \AA}$ of the centroid of the QM water molecule (primary subsystem). When the QM

molecules/atoms are displaced during the finite difference energy evaluation, it is possible that a water molecule will move in/out of the secondary subsystem if it is very close to the boundary defined by the cutoff. We have verified by inspection that this is indeed the source of error for the red-labeled configurations in Figure 3b; e.g. there is a different (differing by one molecule) electrostatic embedding defined (by the stated algorithm) for the two energy evaluations in the finite-difference procedure, and only the first electrostatic embedding is consistent with the direct gradient call (inset schematic, Figure 3b). As shown in Figure 3b, 15 gradient calculations suffered from this numerical error, and the MAE reduces to $0.039 \pm 0.004 \text{ kJ mol}^{-1} \text{ \AA}^{-1}$ when these outliers were removed from the dataset. It is important to note that such numerical errors resulting from inconsistent electrostatic embeddings will manifest broadly within QM/MM/Cutoff utilization. For example, during the course of a geometry optimization, the electrostatic embedding may change if defined relative to the centroid of the QM region which changes during the optimization. In a molecular dynamics simulation, utilizing e.g. a leapfrog integrator, forces computed at different timesteps (and thus different coordinates) are utilized to propagate the Newtonian dynamics, which may be computed with different electrostatic embeddings for the reason stated above.

We now discuss the QM/MM/PME approach, with corresponding gradient benchmarks shown in Figure 3c. The MAE for the gradients vs finite-difference for QM/MM/PME is $0.21 \pm 0.04 \text{ kJ mol}^{-1} \text{ \AA}^{-1}$, which is substantially smaller than in the QM/MM/Cutoff approach. The major reason for enhanced accuracy is the treatment of the “outlier configurations” shown in Figure 3b. In the QM/MM/PME approach, when a water/solvent molecule is transferred between the secondary and tertiary subsystem in the defined embedding (e.g. due to displacements of the molecule in the primary region), the only difference is whether the electrostatic embedding from this molecule is computed in real space (secondary subsystem) or reciprocal space (tertiary subsystem). Thus the corresponding errors for such outlier configurations that were observed in the QM/MM/Cutoff approach are not observed in the QM/MM/PME approach. Due to numerical differences in the real-space and reciprocal space treatment, the corresponding outlier configurations (red data points, Figure 3c) do have slightly larger gradient errors, with MAE reduced to $0.10 \pm 0.02 \text{ kJ mol}^{-1} \text{ \AA}^{-1}$ when these configurations are removed from the dataset. We note that further improvement in the QM/MM/PME gradient accuracy can be realized with finer PME grids, compared to the $1.0 \text{ \AA}$ grid size utilized in the Figure 3c benchmark. For example, QM/MM/PME gradients com-

puted using a grid spacing of $0.25 \text{ \AA}$ yielded an MAE of $0.045 \pm 0.008 \text{ kJ mol}^{-1} \text{ \AA}^{-1}$, with a further reduction to $0.023 \pm 0.004 \text{ kJ mol}^{-1} \text{ \AA}^{-1}$ when discarding the outlier points shown in red. This is comparable accuracy to the “best case” agreement for the gas-phase DFT gradient benchmarks (Figure 3a), and validates the numerical accuracy of the QM/MM/PME gradient equations (Appendix A) and our PyDFT-QMMM software implementation.

For the next benchmark, we analyze energy conservation/energy drift for the QM/MM molecular dynamics simulations within the NVE ensemble. Molecular dynamics simulations for a simulation box of water were conducted utilizing both the QM/MM/Cutoff and QM/MM/PME approaches. Following 50 ps of equilibration in the NVT ensemble, a 100 ps trajectory was generated in the NVE ensemble for the $29.9 \text{ \AA}$ cubic box of water as previously equilibrated with classical MD (*vide supra*). The NVE and NVT simulations were performed with PyDFT-QMMM implementations of the leap-frog Verlet and Langevin integration schemes, respectively. The time step was 1 fs for all simulations, and the NVT simulations used a friction coefficient of 1 ps^{-1} . The SETTLE algorithm was applied to the secondary and tertiary subsystems at each time step as an integrator plugin.²⁸⁰ For comparison, simulations were performed using the STO-3G, def2-SVP, and 6-31G* basis sets for the DFT component of the QM/MM Hamiltonian. Figure 4 depicts the energy drift over 100 ps for both QM/MM/Cutoff and QM/MM/PME simulations using the def2-SVP basis set, for representative simulation trajectories. To determine statistical uncertainty of the energy drift, three independent simulations were run with each approach, resulting in uncertainty estimates given in Figure 4.

In Figure 4a, it is observed that the energy drift from QM/MM/Cutoff and QM/MM/PME MD simulations with def2-SVP basis set are $0.34 \pm 0.11 \text{ kJ mol}^{-1} \text{ ps}^{-1}$ and $2.35 \pm 0.22 \text{ kJ mol}^{-1} \text{ ps}^{-1}$ respectively. Because energy drift will depend on the specific system studied, in addition to the algorithms and software implementation of the method, it is difficult to make an “apples-to-apples” comparison with literature values. As a “loose comparison”, other DFT-QM/MM implementations have reported energy drifts on the order of $\sim 1 \text{ kcal mol}^{-1} \text{ ps}^{-1}$ ($\sim 4 \text{ kJ mol}^{-1} \text{ ps}^{-1}$), but it is important to note that these benchmarks were performed on different systems.^{90,101} It is interesting that the QM/MM/PME simulation exhibits poorer energy conservation as compared to the QM/MM/Cutoff simulation, given the previously discussed gradient benchmarks in Figure 3. Before discussing possible reasons for this, we discuss/analyze the basis-set dependence of energy conservation for the

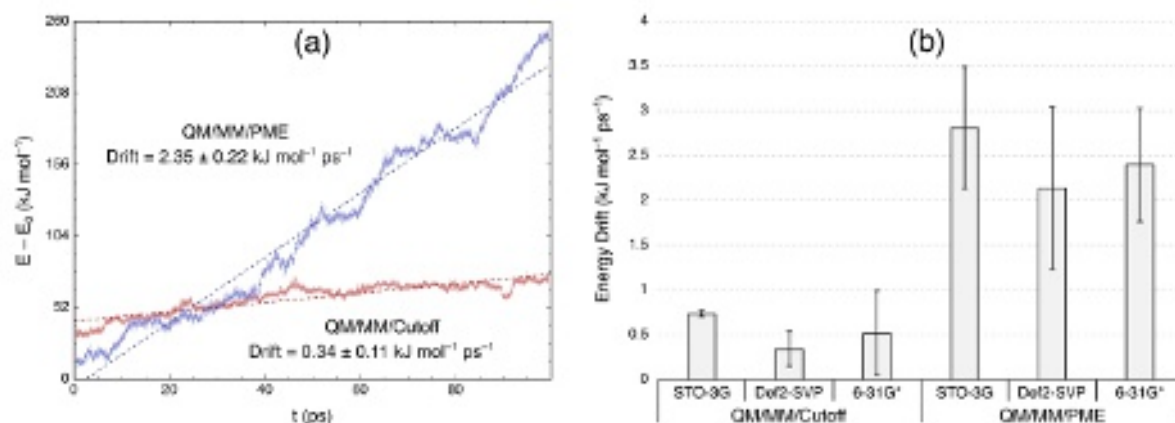


FIG. 4. Representative energy drift over a 100 ps trajectory using a) the QM/MM/Cutoff and QM/MM/PME approaches with the def2-SVP basis set, and b) the drift quantification for several combinations of basis set and approach. The y-axis of a) the representative trajectory graph is the total energy less the energy of the initial MD frame. The trajectories do not appear to start at the zero of energy due to an immediate jump in the energy during the first time-step, with the jump being ~ 34 kJ/mol for the QM/MM/Cutoff trajectory and ~ 13 kJ/mol for the QM/MM/PME trajectory. The error bars of b) the drift quantification correspond to the 95% confidence interval of the distribution of means.

different QM/MM approaches. In Figure 4b, the energy drift from QM/MM/Cutoff and QM/MM/PME MD simulations are reported, as conducted with either STO-3G, def2-SVP, or 6-31G* basis sets. As is seen, the energy drift has only a minor dependence on basis set, relative to the underlying error estimates, and systematic performance difference of the QM/MM/Cutoff and QM/MM/PME approaches. In all cases, the QM/MM/Cutoff simulations exhibit energy drifts on the order of several $0.1 \text{ kJ mol}^{-1} \text{ ps}^{-1}$, while QM/MM/PME simulations exhibit energy drift on the order of several $\text{kJ mol}^{-1} \text{ ps}^{-1}$.

To understand the difference in energy drift of the QM/MM/Cutoff and QM/MM/PME approaches, it is essential to consider the split of the total system into QM (primary) and MM (secondary and tertiary) regions. For the benchmarked system of 895 water molecules, only 1 water molecule was in the QM (primary) region and the remaining 894 water molecules (as well as all periodic replicas) were in the MM (secondary and tertiary) region. Simply put, energy drift is thus dominated by the MM system, and only has a small contribution from the QM system. The treatment/accuracy of forces/gradients on QM atoms, such as the bench-

marks shown in Figure 3, will simply not be reflected in energy conservation benchmarks (Figure 4) that are dominated by the large MM subsystem. Rather, the energy conservation benchmarks in Figure 4 largely reflect accuracy of forces/gradients on MM atoms, which makeup the majority of the system. The force matrix in Equation 10 reveals that neither the QM/MM/Cutoff nor the QM/MM/PME approaches have fully conservative forces. We speculate that the poorer energy drift in the QM/MM/PME approach is due to non conservative forces on the tertiary subsystem (MM) atoms, from the region I-III electrostatic interactions, utilizing our algorithm/protocol. We note that a DFT-QM/MM implementation utilizing the NAMD software reported poorer energy drift utilizing QM/MM+PME compared to QM/MM/Cutoff algorithms for electrostatic embedding,¹⁰¹ although this comparison is different than the QM/MM/PME comparison presented in this work.

To summarize, the QM/MM/PME approach exhibits more accurate gradients on QM atoms (Figure 3), but poorer overall energy conservation (Figure 4) than the QM/MM/Cutoff approach. The latter is simply a metric of the MM (secondary and tertiary) subsystem treatment, and has little-to-no evaluatory merit for the QM (primary) subsystem treatment, because the QM region is only a small fraction of the total system. The conclusion is that energy conservation may not be the best metric to gauge the predictive accuracy of a given QM/MM approach, when the focus is on predicting the properties of the primary/QM subsystem. For example, although QM/MM/Cutoff exhibits relatively good energy conservation for the benchmarked system (Figure 4), there are many chemical applications in which truncated electrostatic interactions lead to severe errors in predicted properties of the primary/QM subsystem.^{92,281} Alternatively, a more rigorous treatment of long-range electrostatic interactions for the QM/MM energy and QM forces as in the QM/MM/PME approach, may lead to higher accuracy predictions of target properties, despite the introduction of non-conservative forces on the tertiary subsystem leading to somewhat worse energy conservation. Needless to say, most practical QM/MM molecular dynamics simulations are run in the NVT ensemble, in which energy drift on the order of $\text{kJ mol}^{-1} \text{ps}^{-1}$ for a 895 water molecule system will be easily dissipated with any practical choice of thermostat.

Our final benchmark focuses on the liquid structure of water, as computed from the QM/MM MD simulations. Radial distribution functions (RDF) provide the most straightforward metric of the liquid structure, and exhibit well-known features that have also served as the basis for evaluating forcefield accuracy. For our QM/MM simulations, we specifically

focus on RDFs computed between the primary (QM) and secondary (MM) subsystems, which can be compared to corresponding RDFs resulting from the purely MM/forcefield description. The 100 ps QM/MM, NVE simulation trajectories that were discussed above were used to construct O-H RDFs between water molecule(s) in the primary (QM) and secondary (MM) subsystems, utilizing the VMD software.²⁸² Atom positions were recorded every 50 time steps, providing 2000 frames from which to construct RDFs. Radial bins of width 0.1 Å were used from 0. to 5. Å about the selected atom group. RDFs as computed from both QM/MM/Cutoff and QM/MM/PME simulations, as conducted with either STO-3G, def2-SVP, or 6-31G* basis sets, are presented in Figure 5. In Figure 5, each subpanel a)-f) depicts RDFs computed between: 1) water oxygen and hydrogen atoms in the secondary subsystem “O_{II}-H_{II}”; 2) oxygen atoms in the secondary subsystem and hydrogen atoms in the primary subsystem “O_{II}-H_I”; 3) oxygen atoms in the primary subsystem and hydrogen atoms in the secondary subsystem “O_I-H_{II}”. The RDFs labeled “O_{II}-H_I” and “O_I-H_{II}” are thus a metric of the QM/MM description of water structure/hydrogen bonding, while the “O_{II}-H_{II}” RDF is a metric of the MM description of water structure/hydrogen bonding as given by the SPC/E forcefield.

The “O_{II}-H_{II}” RDF is considered to be the reference, since the SPC/E forcefield has been empirically parameterized to provide a good description of the properties of bulk water. From Figure 5, the first peak in this RDF is located at 1.75 Å with a height of ~ 1.5 , reflecting hydrogen bonding of adjacent water molecules. The second, broader peak is located at 3.25 Å, with a height of approximately 1.5. The other “O_{II}-H_I” and “O_I-H_{II}” RDFs are strongly modulated by the nature of the QM/MM interaction, since they depict water interactions/hydrogen bonding between the primary (QM) and secondary (MM) subsystems. In this regard, it is immediately clear from Figure 5 that these latter RDFs are strongly modulated by the size of the basis set. The “O_{II}-H_I” and “O_I-H_{II}” RDFs computed with the small STO-3G basis set, QM/MM simulations depict weak hydrogen bonding, with substantially reduced height of the first RDF peak. In contrast, the larger 6-31G* basis set, QM/MM simulations predict RDFs with significantly enhanced first peak height, corresponding to over-structuring or exaggerated hydrogen bonding. For a specific basis set, the QM/MM/Cutoff and QM/MM/PME simulations predict very similar RDFs; this is because the RDFs depict short-range liquid structure, and these QM/MM approaches only differ in their long range electrostatic descriptions.

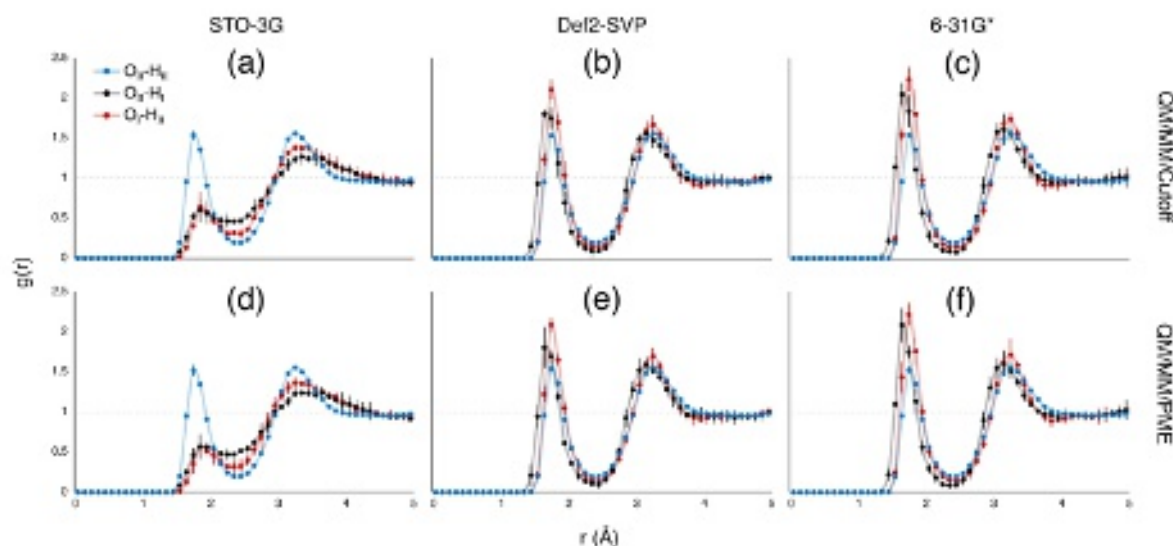


FIG. 5. Radial distribution function $g(r)$ for water oxygen and hydrogen atoms across the primary (I) and secondary (II) subsystems for QM/MM/Cutoff simulations employing a) STO-3G, b) Def2-SVP, and c) 6-31G(d) basis sets and for QM/MM/PME simulations employing d) STO-3G, e) Def2-SVP, and f) 6-31G(d) basis sets as a function of distance (r). Error bars correspond to the 95% confidence interval of the distribution of means.

The fundamental cause of the substantial basis set dependence of the RDFs shown in Figure 5 is different electrostatic and polarization descriptions of the QM water molecule. It is well known that, in addition to the static dipole moment, electronic polarization makes a significant contribution to the hydrogen bond energies of water.²⁸³ In QM/MM, polarization at the primary (QM) subsystem boundary is a subtle, but important issue; because there is no explicit electron operator to account for exchange-repulsion between the primary and secondary subsystems, overpolarization or “electron spill-out” from the QM system onto surrounding MM partial charges, is a potential problem that has been reported and discussed.^{37,284} In lieu of developing a more sophisticated QM/MM interaction Hamiltonian, the issue can only be addressed by either compromising on basis set choice and/or tuning the Lennard-Jones cross interactions.^{285,286} As depicted in Figure 5, larger basis sets will tend to over-polarize in the presence of the local solvent environment, in this case predicting hydrogen bonding that is too strong. Basis sets with diffuse functions will be particularly problematic in this regard, and thus tend to be avoided in QM/MM applications. Smaller basis sets, in addition to their fundamentally limited description of the electronic structure

of the QM system, do not have the flexibility to describe polarization adequately, resulting in under-polarization and hydrogen bonding with the surrounding solvation environment that is too weak. Our benchmarks in Figure 5 indicate that the def2-SVP basis set provides the most “well-balanced” description of electrostatics/polarization within the QM/MM interaction; the “O_{II}-H_I” and “O_I-H_{II}” RDFs computed from these QM/MM simulations are in comparatively good agreement with the “O_{II}-H_{II}” RDF that is predicted purely by the SPC/E forcefield.

Our benchmarks in Figure 5 provide just one example of why basis set choice in QM/MM simulations is subtle and complex. Because a “good” prediction relies on inherent error cancellation (i.e. over-polarization is only prevented because the basis set is limited/incomplete), the best choice of basis set will be system specific, and should be chosen based on careful benchmarking. For a QM/MM simulation of bulk water, the best choice of basis set is even more involved/complex than indicated by the benchmarks in Figure 5. This is because our simulations have considered only one water molecule in the QM region, such that the only intermolecular interactions affected by basis set choice are between primary (QM) and secondary (MM) water molecules; in other words, there are no water-water intermolecular interactions described purely at the QM level. When there are many water molecules in the primary (QM) subsystem, it is essential to also consider the effect of basis set on describing the water structure within the primary subsystem. Previous QM/MM simulations have demonstrated the sensitivity of water structure and solute/water descriptions to choice of basis set;²⁸⁷ indeed this is consistent with *ab initio* molecular dynamics simulations of bulk water, in which the predicted water structure and properties are highly sensitive to the employed level of theory.²⁸⁸ For the QM/MM approaches discussed in this work, choosing the “best” basis set for a given application is thus fundamentally a compromise between adequate description of both the electronic structure and intermolecular interactions within the primary (QM) subsystem, and a balanced electrostatic/polarization description at the primary (QM)/secondary (MM) subsystem boundary.

V. CONCLUSIONS

In this manuscript, we have presented PyDFT-QMMM, a modular, Python-based package that provides a framework for performing DFT-based, QM/MM molecular dynamics

simulations, interfacing with Psi4 and OpenMM as calculators. The primary goal of the package is to provide advanced control over embedding procedures, specifically focusing on periodic boundary conditions. In order to provide an organized description of possible PBC embedding choices, we have presented a novel taxonomy for QM/MM approaches based on a partition of the MM environment into local (secondary) and extended (tertiary) subsystems. This distinction allows for expanded discussion of QM/MM implementations, including our QM/MM/PME⁹² and QM/MM/Cutoff approaches. Of particular interest is the “force matrix” of Equation 10, which describes the levels of theory used to propagate forces arising from interactions between various subsystems. We have derived forces for our QM/MM/PME approach in Appendix A. Benchmarks of the QM/MM/PME forces and general PyDFT-QMMM implementation were demonstrated for a water solute (QM)/water solvent (MM) system. Energy conservation from QM/MM, NVE molecular dynamics simulations and water solvation structure were evaluated for the water solute/water solvent system, employing both QM/MM/Cutoff and QM/MM/PME methods with basis sets of varying size. The solvation structure was shown to sensitively depend on basis-set size, for which small/large basis sets lead respectively to under/over polarization of the QM electron density, and thus intermediate sized (e.g. def2-SVP) basis sets are preferable.

While prior QM/MM implementations interfacing between QC and MD packages have employed a variety of communication schemes, PyDFT-QMMM interoperates with programs loaded in memory through Python APIs. Such an approach: 1) avoids computational overheads associated with program starting/stopping, file system reading/writing, and data duplication; 2) permits persistence of objects from the underlying QC and MD packages; and 3) provides access to utilities in the QC and MD software made available through Python-bindings. The added flexibility allows for development of more advanced QM/MM procedures. The lightweight core library of PyDFT-QMMM provides sufficient utilities for performing QM/MM molecular dynamics simulations with a variety of embedding schemes, and the architecture allows for facile installation and application of both local and third-party plug-ins. User interaction through Python API allows for scripting of advanced workflows, which are particularly important for studying complex chemical systems. For example, our group has utilized PyDFT-QMMM for computational electrosynthesis applications, which requires custom simulation protocols for modeling electrochemical cells under working conditions.²⁸⁹ We hope that PyDFT-QMMM will serve as an important tool

to the computational community, particularly for QM/MM molecular dynamics/free energy simulations of chemical reactions in complex environments.

The PyDFT-QMMM package is available on github via the link in Reference ²⁹⁰.

Appendix A: Analytic Force Derivations

In order to perform geometry optimizations or statistical sampling through dynamic simulation, forces from the total system Hamiltonian are required for atoms and nuclei in subsystems I, II, and III. There are standard force contributions from the independent QM Hamiltonian in subsystem I and the independent MM Hamiltonian in subsystems II and III. Here, we discuss the contribution from the QM/MM coupling terms. The electrostatic QM/MM coupling terms warrant discussion, while the non-electrostatic non-bonded contributions are treated with mechanical embedding and warrant no further discussion. The QM/MM electrostatic embedding generates forces on both the subsystem I nuclei, $F_{\text{I-II}}$, and the subsystem II atoms, $F_{\text{II-I}}$, as we discuss in the following subsections. We also present the force contribution, $F_{\text{I-III}}$, on the primary subsystem arising from the I-III interaction described by the potential in Equation 8.

1. Electrostatic Forces on Subsystem II

The I-II interaction energy from electrostatic embedding in Equation 4 can be re-written as

$$E_{\text{I-II}}^{\text{QM,elst}} = E_{\text{I-II}}^{\text{elec}} + E_{\text{I-II}}^{\text{nuc}}, \quad (\text{A1})$$

where

$$E_{\text{I-II}}^{\text{nuc}} = \sum_i^{N_{\text{I}}} \sum_n^{N_{\text{II}}} \frac{Z_i q_n}{|\vec{x}_i - \vec{x}_n|}, \quad (\text{A2})$$

and

$$E_{\text{I-II}}^{\text{elec}} = -D_{\mu\nu} \int d\vec{r} \phi_\mu(\vec{r}; \vec{x}) \phi_\nu(\vec{r}; \vec{x}) \sum_n^{N_{\text{II}}} \frac{q_n}{|\vec{r} - \vec{x}_n|}, \quad (\text{A3})$$

where we have now explicitly indicated the parametric dependence of the orbital densities on the set of nuclear coordinates, e.g. $\phi_\mu(\vec{r}; \vec{x})$. Forces from the nuclear term are trivial. Forces from the electronic term on subsystem II come from taking the negative gradient

w.r.t. subsystem II atomic coordinates, e.g.

$$\begin{aligned}\vec{F}_{\{m \in \text{II}\}-\text{I}}^{\text{elec}} &= -\frac{\partial E_{\text{I-II}}^{\text{elec}}}{\partial \vec{x}_m} \\ &= D_{\mu\nu} \int d\vec{r} \phi_\mu(\vec{r}; \vec{x}) \phi_\nu(\vec{r}; \vec{x}) \left(\frac{\partial}{\partial \vec{x}_m} \sum_n^{N_{\text{II}}} \frac{q_n}{|\vec{r} - \vec{x}_n|} \right) \\ &= D_{\mu\nu} \int d\vec{r} \phi_\mu(\vec{r}; \vec{x}) \phi_\nu(\vec{r}; \vec{x}) \left(\frac{\partial}{\partial \vec{x}_m} \frac{q_m}{|\vec{r} - \vec{x}_m|} \right).\end{aligned}\quad (\text{A4})$$

This is a Hellmann-Feynman-like term,^{90,291} or a derivative applied to the integral in the energy expression. Neither derivatives on the atomic orbital basis functions nor a derivative on the density matrix are necessary because the embedded point charges from region II do not have basis sets. Note that these integrals *should* be readily accessible within an electronic structure code/integral library, as they have the same form as a force contribution from the nuclear-electron integrals.

2. Electrostatic Forces on Subsystem I

The contributions from inter-subsystem interactions in Equation 4 can be expanded from Equation A1 to include I-III as

$$E_{\text{I}-\{\text{II} \cup \text{III}\}}^{\text{QM,elst}} = E_{\text{I-II}}^{\text{elec}} + E_{\text{I-II}}^{\text{nuc}} + E_{\text{I-III}}^{\text{elec}} + E_{\text{I-III}}^{\text{nuc}}, \quad (\text{A5})$$

where $E_{\text{I-II}}^{\text{nuc}}$ and $E_{\text{I-II}}^{\text{elec}}$ are given by Equations A2 and A3, respectively, and

$$E_{\text{I-III}}^{\text{nuc}} = \sum_i^{N_{\text{I}}} Z_i v_{\text{III}}(\vec{x}_i), \quad (\text{A6})$$

and

$$E_{\text{I-III}}^{\text{elec}} = -D_{\mu\nu} \int d\vec{r} \phi_\mu(\vec{r}; \vec{x}) \phi_\nu(\vec{r}; \vec{x}) v_{\text{III}}(\vec{r}). \quad (\text{A7})$$

Differentiating the I-II electronic component w.r.t nuclear coordinates gives the force contribution

$$\begin{aligned}\vec{F}_{\{j \in \text{I}\}-\text{II}}^{\text{elec}} &= -\frac{\partial E_{\text{I-II}}^{\text{elec}}}{\partial \vec{x}_j} \\ &= \frac{\partial D_{\mu\nu}}{\partial \vec{x}_j} \int d\vec{r} \phi_\mu(\vec{r}; \vec{x}) \phi_\nu(\vec{r}; \vec{x}) \left(\sum_n^{N_{\text{II}}} \frac{q_n}{|\vec{r} - \vec{x}_n|} \right) \\ &\quad + 2D_{\mu\nu} \int d\vec{r} \frac{\partial \phi_\mu(\vec{r}; \vec{x})}{\partial \vec{x}_j} \phi_\nu(\vec{r}; \vec{x}) \left(\sum_n^{N_{\text{II}}} \frac{q_n}{|\vec{r} - \vec{x}_n|} \right),\end{aligned}\quad (\text{A8})$$

which is provided by standard QC software packages. The I-III electronic contribution is similarly written as

$$\begin{aligned}\vec{F}_{\{j \in \text{I}\}-\text{III}}^{\text{elec}} &= -\frac{\partial E_{\text{I-III}}^{\text{elec}}}{\partial \vec{x}_j} \\ &= \frac{\partial D_{\mu\nu}}{\partial \vec{x}_j} \int d\vec{r} \phi_\mu(\vec{r}; \vec{x}) \phi_\nu(\vec{r}; \vec{x}) v_{\text{III}}(\vec{r}) \\ &\quad + 2D_{\mu\nu} \int d\vec{r} \frac{\partial \phi_\mu(\vec{r}; \vec{x})}{\partial \vec{x}_j} \phi_\nu(\vec{r}; \vec{x}) v_{\text{III}}(\vec{r}).\end{aligned}\tag{A9}$$

It is standard practice to incorporate all derivative terms that involve explicit differentiation of the density matrix, as in line 2 of Equation A9, into the “energy-weighted” density matrix contribution,²⁹¹ e.g.

$$\frac{\partial D_{\mu\nu}}{\partial \vec{x}_j} \int d\vec{r} \phi_\mu(\vec{r}; \vec{x}) \phi_\nu(\vec{r}; \vec{x}) v_{\text{III}}(\vec{r}) + \dots \Rightarrow -\sum_p \epsilon_p c_{\mu p}^* c_{\nu p} \frac{\partial S_{\mu\nu}}{\partial \vec{x}_j},\tag{A10}$$

where ‘...’ denotes the standard derivative contributions of the Kohn-Sham energy involving explicit differentiation of the density matrix, including the density matrix derivative from Equation A8. In this equation, $S_{\mu\nu}$ is the overlap matrix, and ϵ_p are the orbital energies with p denoting molecular orbital indices. Thus, the term that needs to be added to an electronic structure code is the Hellmann-Feynman-like part of the force contribution, or line 3 of Equation A9. In practice, these integrals are computed by numerical quadrature in our QM/MM/PME approach. Because all integrals involving $v_{\text{III}}(\vec{r})$ are computed using numerical quadrature, and the quadrature grid (in principle) depends on the nuclear coordinates x , neglecting this dependence in the differentiation could lead to small numerical error;^{90,292–294} however, in practice, this can be remedied by using a large quadrature grid and foregoing a pruning scheme.

The force on the primary subsystem arising from the nuclear component of the I-III interaction, Equation A6, requires further discussion of implementation details. In our direct QM/MM/PME method, $v_{\text{III}}(\vec{x}_i)$ at real space positions is evaluated by interpolating from v_{III} values on the PME grid. Real space coordinates are first projected to scaled, fractional coordinates on the PME grid

$$\vec{u}_i = K \mathbf{a}^* \cdot \vec{x}_i,\tag{A11}$$

where $\mathbf{a}^*$ is the matrix of reciprocal lattice vectors defined by $\mathbf{a}_\alpha^* \cdot \mathbf{a}_\beta = \delta_{\alpha\beta}$. $\vec{F}_{\{j \in \text{I}\}-\text{III}}^{\text{nuc}}$ is then

given by

$$\begin{aligned}
 \vec{F}_{\{j \in I\}-III}^{\text{nuc}} &= -\frac{\partial E_{I-III}^{\text{nuc}}}{\partial \vec{x}_j} \\
 &= -Z_j \frac{dv_{III}(\vec{x}_j)}{d\vec{x}_j} \\
 &= -Z_j \frac{dv_{III}(\vec{u}_j)}{d\vec{u}_j} \frac{d\vec{u}_j}{d\vec{x}_j} \\
 &= -K * Z_j \frac{dv_{III}(\vec{u}_j)}{d\vec{u}_j} \cdot \mathbf{a}^*.
 \end{aligned} \tag{A12}$$

The value of the electrostatic potential at the scaled, fractional coordinate $\vec{u}_j$ is given by trilinear interpolation from the 3D PME grid. It's derivative is

$$\frac{dv_{III}(\vec{u}_j)}{du_{j,\alpha}} = v_{III}(\vec{u}_j)|_{u_{j,\alpha}=\alpha_1} - v_{III}(\vec{u}_j)|_{u_{j,\alpha}=\alpha_0}, \tag{A13}$$

where α_1 , α_0 are the bounding interpolation PME gridpoints, with $\alpha_0 < u_{i,\alpha} < \alpha_1$. The required $v_{III}(\vec{u}_j)|_{u_{j,\alpha}=\alpha_1}$ and $v_{III}(\vec{u}_j)|_{u_{j,\alpha}=\alpha_0}$ in Equation A13 are each given by bilinear interpolation. We use the SciPy implementation of trilinear interpolation found in `scipy.interpolate.RegularGridInterpolator._evaluate_linear`. The interpolation is computed by summing over the contribution from the eight edges/vertices of the cube enclosing the point. Let these edges/vertices be labeled (000),(001),(010), etc., and f_{000} is the function value evaluated at the (000) edge/vertex. Also let x_d , y_d , z_d denote the fractional distance of the point along the edge in the cartesian direction, i.e.

$$x_d = \frac{x - x_0}{x_1 - x_0}, \tag{A14}$$

and similar for y_d , z_d . The interpolation is given explicitly by the contributions from the eight edges/vertices, as

$$\begin{aligned}
 f(x, y, z) &= f_{000} * (1 - x_d)(1 - y_d)(1 - z_d) \\
 &+ f_{100} * x_d(1 - y_d)(1 - z_d) \\
 &+ f_{010} * (1 - x_d)y_d(1 - z_d) \\
 &+ f_{110} * x_dy_d(1 - z_d) \\
 &+ f_{001} * (1 - x_d)(1 - y_d)z_d \\
 &+ f_{101} * x_d(1 - y_d)z_d \\
 &+ f_{011} * (1 - x_d)y_dz_d \\
 &+ f_{111} * x_dy_dz_d
 \end{aligned} \tag{A15}$$

derivatives are now easily computed, e.g.

$$\begin{aligned}\frac{df(x, y, z)}{dx} = & -f_{000} * (1 - y_d)(1 - z_d) \\ & + f_{100} * (1 - y_d)(1 - z_d) \\ & - f_{010} * y_d(1 - z_d) \\ & + f_{110} * y_d(1 - z_d) \\ & - f_{001} * (1 - y_d)z_d \\ & + f_{101} * (1 - y_d)z_d \\ & - f_{011} * y_dz_d \\ & + f_{111} * y_dz_d\end{aligned}\tag{A16}$$

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 2237792. This research was partially supported by the Air Force Office of Scientific Research, under award FA9550-22-1-0025. The authors acknowledge research cyberinfrastructure resources and services provided by the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology, Atlanta, Georgia, USA

REFERENCES

- ¹J. Gao, Acc. Chem. Res. **29**, 298 (1996).
- ²G. Norjmaa, G. Ujaque, and A. Lledós, Top Catal **65**, 118 (2022).
- ³I. F. Galván, A. Muñoz-Losa, C. Soriano-Correa, M. Luz Sánchez, M. Elena Martín, and M. A. Aguilar, in *Advances in Quantum Chemistry*, Vol. 59 (Elsevier, 2010) pp. 59–97.
- ⁴C. J. Burrows, J. B. Harper, W. Sander, and D. J. Tantillo, J. Org. Chem. **87**, 1599 (2022).
- ⁵D. J. Tantillo, J of Physical Organic Chem **34**, e4202 (2021).
- ⁶J. M. Boereboom, P. Fleurat-Lessard, and R. E. Bulo, J. Chem. Theory Comput. **14**, 1841 (2018).
- ⁷F. Lipparini and B. Mennucci, Chemical Physics Reviews **2**, 041303 (2021).

- ⁸S. Ahmadi, L. B. Herrera, M. Chehelamirani, J. Hostaš, S. Jalife1, and D. R. Salahub, *Int. J. Quantum Chem.* **118**, e25558 (2017).
- ⁹K. Csizi and M. Reiher, *WIREs Comput Mol Sci* **13**, e1656 (2023).
- ¹⁰C. E. Tzeliou, M. A. Mermigki, and D. Tzeli, *Molecules* **27**, 2660 (2022).
- ¹¹V. Vennelakanti, A. Nazemi, R. Mehmood, A. H. Steeves, and H. J. Kulik, *Curr. Opin. Struct. Biol.* **72**, 9 (2022).
- ¹²J. M. Herbert, *Wiley Interdisciplinary Reviews: Computational Molecular Science* **11**, 1 (2021).
- ¹³S. Miertuš, E. Scrocco, and J. Tomasi, *Chemical Physics* **55**, 117 (1981).
- ¹⁴E. Cancès, B. Mennucci, and J. Tomasi, *The Journal of Chemical Physics* **107**, 3032 (1997).
- ¹⁵T. Vreven, B. Mennucci, C. O. Da Silva, K. Morokuma, and J. Tomasi, *The Journal of Chemical Physics* **115**, 62 (2001).
- ¹⁶H. Senn and W. Thiel, *Angew. Chem., Int. Ed.* **48**, 1198 (2009).
- ¹⁷H. Lin, *QM/MM Methods* (American Chemical Society, Washington, DC, USA, 2024).
- ¹⁸J. R. Pliego and J. M. Riveros, *WIREs Comput Mol Sci* **10**, e1440 (2020).
- ¹⁹P. E. Siegbahn and F. Himo, *WIREs Comput Mol Sci* **1**, 323 (2011).
- ²⁰P. Schaefer, D. Riccardi, and Q. Cui, *The Journal of Chemical Physics* **123**, 014905 (2005).
- ²¹T. Benighaus and W. Thiel, *Journal of Chemical Theory and Computation* **5**, 3114 (2009).
- ²²T. Benighaus and W. Thiel, *J. Chem. Theory Comput.* **7**, 238 (2011).
- ²³J. Nochebuena, S. Naseem-Khan, and G. A. Cisneros, *WIREs Comput Mol Sci* **11**, e1515 (2021).
- ²⁴J. Zienau and Q. Cui, *J. Phys. Chem. B* **116**, 12522 (2012).
- ²⁵S. Chalmet, D. Rinaldi, and M. F. Ruiz-Lopez, *Int. J. Quantum. Chem.* **84**, 559 (2001).
- ²⁶C. Cappelli, *Int J of Quantum Chemistry* **116**, 1532 (2016).
- ²⁷J. Chen, Y. Shao, and J. Ho, *J. Phys. Chem. A* **123**, 5580 (2019).
- ²⁸P. P. Fehér and A. Stirling, *New J. Chem.* **43**, 15706 (2019).
- ²⁹R. Berraud-Pache, C. Garcia-Iriepe, and I. Navizet, *ACS Catal.* **6**, 116 (2018).
- ³⁰O. Acevedo and W. L. Jorgensen, *Acc. Chem. Res.* **43**, 142 (2010).
- ³¹Z. Yang, C. Doubleday, and K. N. Houk, *J. Chem. Theory Comput.* **11**, 5606 (2015).
- ³²T. Saito and Y. Takano, *J. Phys. Chem. B* **126**, 2087 (2022).
- ³³W.-W. Guo, T.-S. Zhang, W.-H. Fang, and G. Cui, *Phys. Chem. Chem. Phys.* **20**, 5067

- (2018).
- ³⁴E. A. Briggs, N. A. Besley, and D. Robinson, *J. Phys. Chem. A* **117**, 2644 (2013).
 - ³⁵T. Kubar, M. Elstner, and Q. Cui, *Annu. Rev. Biophys.* **52**, 525 (2023).
 - ³⁶A. R. Murphy, M. A. Hix, and A. R. Walker, *ChemBioChem* **24**, e202200799 (2023).
 - ³⁷E. Brunk and U. Rothlisberger, *Chem. Rev.* **115**, 6217 (2015).
 - ³⁸M. Salaheen and A. Heyden, *ACS Catal.* **8**, 2188 (2018).
 - ³⁹A. A. Sokol, S. T. Bromley, S. A. French, C. R. A. Catlow, and P. Sherwood, *Int. J. Quantum Chem.* **99**, 695 (2004).
 - ⁴⁰M. G. Quesne, F. Silveri, N. H. de Leeuw, and C. R. A. Catlow, *Front. Chem.* **7**, 182 (2019).
 - ⁴¹N. Fey, *Dalton Trans.* **39**, 296 (2009).
 - ⁴²M. Capone, M. Romanelli, D. Castaldo, G. Parolin, A. Bello, G. Gil, and M. Vanzan, *ACS Phys. Chem Au* , acsphyschemau.3c00080 (2024).
 - ⁴³A. Hoekstra, B. Chopard, and P. Coveney, *Phil. Trans. R. Soc. A.* **372**, 20130377 (2014).
 - ⁴⁴H. Lin and D. G. Truhlar, *Theoretical Chemistry Accounts* **117**, 185 (2007).
 - ⁴⁵Y. Zhang and H. Lin, *J. Chem. Theory Comput.* **4**, 414 (2008).
 - ⁴⁶S. Pezeshki and H. Lin, *Molecular Simulation* **41**, 168 (2015).
 - ⁴⁷C. M. Clemente, L. Capece, and M. A. Martí, *J. Chem. Inf. Model.* **63**, 2609 (2023).
 - ⁴⁸B. Raghavan, M. Paulikat, K. Ahmad, L. Callea, A. Rizzi, E. Ippoliti, D. Mandelli, L. Bonati, M. De Vivo, and P. Carloni, *J. Chem. Inf. Model.* **63**, 3647 (2023).
 - ⁴⁹G. A. Bramley, O. T. Beynon, P. V. Stishenko, and A. J. Logsdail, *Phys. Chem. Chem. Phys.* **25**, 6562 (2023).
 - ⁵⁰D. Demapan, J. Kussmann, C. Ochsenfeld, and Q. Cui, *J. Chem. Theory Comput.* **18**, 2530 (2022).
 - ⁵¹G. Jindal and A. Warshel, *J. Phys. Chem. B* **120**, 9913 (2016).
 - ⁵²W. Thiel, *WIREs Comput Mol Sci* **4**, 145 (2014).
 - ⁵³G. Seifert, *J. Phys. Chem. A* **111**, 5609 (2007).
 - ⁵⁴F. Spiegelman, N. Tarrat, J. Cuny, L. Dontot, E. Posenitskiy, C. Martí, A. Simon, and M. Rapacioli, *Advances in Physics: X* **5**, 1710252 (2020).
 - ⁵⁵S. C. L. Kamerlin and A. Warshel, *WIREs Comput Mol Sci* **1**, 30 (2011).
 - ⁵⁶M. Karelina and H. J. Kulik, *J. Chem. Theory Comput.* **13**, 563 (2017).
 - ⁵⁷T. Kirsch, J. M. H. Olsen, V. Bolnykh, S. Meloni, E. Ippoliti, U. Rothlisberger, M. Cas-

- cella, and J. Gauss, J. Chem. Theory Comput. **18**, 13 (2022).
- ⁵⁸G. Bistoni, I. Polyak, M. Sparta, W. Thiel, and F. Neese, J. Chem. Theory Comput. **14**, 3524 (2018).
- ⁵⁹D. A. Yarne, M. E. Tuckerman, and G. J. Martyna, The Journal of Chemical Physics **115**, 3531 (2001).
- ⁶⁰D. Hunt, V. M. Sanchez, and D. A. Scherlis, J. Phys.: Condens. Matter **28**, 335201 (2016).
- ⁶¹M. Eichinger, P. Tavan, J. Hutter, and M. Parrinello, The Journal of Chemical Physics **110**, 10452 (1999).
- ⁶²A. Laio, J. VandeVondele, and U. Rothlisberger, The Journal of Chemical Physics **116**, 6941 (2002).
- ⁶³T. Laino, F. Mohamed, A. Laio, and M. Parrinello, J. Chem. Theory Comput. **1**, 1176 (2005).
- ⁶⁴T. Laino, F. Mohamed, A. Laio, and M. Parrinello, J. Chem. Theory Comput. **2**, 1370 (2006).
- ⁶⁵D. Golze, M. Iannuzzi, M.-T. Nguyen, D. Passerone, and J. Hutter, J. Chem. Theory Comput. **9**, 5086 (2013).
- ⁶⁶A. Crespo, D. A. Scherlis, M. A. Martí, P. Ordejón, A. E. Roitberg, and D. A. Estrin, J. Phys. Chem. B **107**, 13728 (2003).
- ⁶⁷C. F. Sanz-Navarro, R. Grima, A. García, E. A. Bea, A. Soba, J. M. Cela, and P. Ordejón, Theor Chem Acc **128**, 825 (2011).
- ⁶⁸A. O. Dohn, E. O. Jónsson, G. Levi, J. J. Mortensen, O. Lopez-Acevedo, K. S. Thygesen, K. W. Jacobsen, J. Ulstrup, N. E. Henriksen, K. B. Møller, and H. Jónsson, J. Chem. Theory Comput. **13**, 6010 (2017).
- ⁶⁹H. Takahashi, T. Hori, T. Wakabayashi, and T. Nitta, J. Phys. Chem. A **105**, 4351 (2001).
- ⁷⁰H. Takahashi and N. Matubayasi, in *Quantum Modeling of Complex Molecular Systems*, Vol. 21, edited by J.-L. Rivail, M. Ruiz-Lopez, and X. Assfeld (Springer International Publishing, Cham, 2015) pp. 153–196, series Title: Challenges and Advances in Computational Chemistry and Physics.
- ⁷¹F. Maseras and K. Morokuma, J Comput Chem **16**, 1170 (1995).
- ⁷²S. Roßbach and C. Ochsenfeld, J. Chem. Theory Comput. **13**, 1102 (2017).
- ⁷³J. Ho, Y. Shao, and J. Kato, Molecules **23**, 2466 (2018).
- ⁷⁴L. W. Chung, W. M. C. Sameera, R. Ramozzi, A. J. Page, M. Hatanaka, G. P. Petrova,

- T. V. Harris, X. Li, Z. Ke, F. Liu, H.-B. Li, L. Ding, and K. Morokuma, *Chem. Rev.* **115**, 5678 (2015).
- ⁷⁵A. H. De Vries, P. T. Van Duijnen, A. H. Juffer, J. A. C. Rullmann, J. P. Dijkman, H. Merenga, and B. T. Thole, *J Comput Chem* **16**, 37 (1995).
- ⁷⁶O. Acevedo and W. L. Jorgensen, *WIREs Comput Mol Sci* **4**, 422 (2014).
- ⁷⁷O. Acevedo, *J. Phys. Chem. A* **118**, 11653 (2014).
- ⁷⁸D. Fang, R. E. Duke, and G. A. Cisneros, *The Journal of Chemical Physics* **143**, 044103 (2015).
- ⁷⁹E. G. Kratz, R. E. Duke, and G. A. Cisneros, *Theor Chem Acc* **135**, 166 (2016).
- ⁸⁰X. Pan, E. Rosta, and Y. Shao, *Molecules* **23**, 2500 (2018).
- ⁸¹D. Pan, *Chemical Engineering Science* **234**, 116447 (2021).
- ⁸²T. Vasilevskaya and W. Thiel, *J. Chem. Theory Comput.* **12**, 3561 (2016).
- ⁸³R. C. Walker, M. F. Crowley, and D. A. Case, *J Comput Chem* **29**, 1019 (2008).
- ⁸⁴D. Riccardi, P. Schaefer, and Q. Cui, *J. Phys. Chem. B* **109**, 17715 (2005).
- ⁸⁵S. Bonfrate, N. Ferré, and M. Huix-Rotllant, *The Journal of Chemical Physics* **158**, 021101 (2023).
- ⁸⁶T. J. Giese and D. M. York, *Journal of Chemical Theory and Computation* **12**, 2611 (2016).
- ⁸⁷K. Nam, J. Gao, and D. M. York, *J. Chem. Theory Comput.* **1**, 2 (2005).
- ⁸⁸Z. C. Holden, R. M. Richard, and J. M. Herbert, *The Journal of Chemical Physics* **139**, 244108 (2013).
- ⁸⁹Z. C. Holden, R. M. Richard, and J. M. Herbert, *The Journal of Chemical Physics* **142**, 059901 (2015).
- ⁹⁰Z. C. Holden, B. Rana, and J. M. Herbert, *The Journal of Chemical Physics* **150**, 144115 (2019).
- ⁹¹O. Kobayashi and S. Nanbu, *Chemical Physics* **461**, 47 (2015).
- ⁹²J. P. Pederson and J. G. McDaniel, *The Journal of Chemical Physics* **156**, 174105 (2022).
- ⁹³Y. Zhou, S. Wang, Y. Li, and Y. Zhang, in *Methods in Enzymology*, Vol. 577 (Elsevier, 2016) pp. 105–118.
- ⁹⁴P. Ojeda-May and J. Pu, *J. Chem. Theory Comput.* **10**, 134 (2014).
- ⁹⁵P. Ojeda-May and J. Pu, *The Journal of Chemical Physics* **143**, 174111 (2015).
- ⁹⁶F. Dehez, M. T. C. Martins-Costa, D. Rinaldi, and C. Millot, *The Journal of Chemical*

- Physics **122**, 234503 (2005).
- ⁹⁷M. Liu, Y. Chen, M. J. Field, and J. Gao, *Isr. J. Chem.* **54**, 1250 (2014).
- ⁹⁸H. M. Senn and W. Thiel, in *Atomistic Approaches in Modern Biology*, Vol. 268, edited by M. Reiher (Springer Berlin Heidelberg, 2007) pp. 173–290, series Title: Topics in Current Chemistry.
- ⁹⁹A. Warshel and M. Levitt, *J. Mol. Biol.* **103**, 227 (1976).
- ¹⁰⁰U. C. Singh and P. A. Kollman, *J. Comput. Chem.* **7**, 718 (1986).
- ¹⁰¹M. C. R. Melo, R. C. Bernardi, T. Rudack, M. Scheurer, C. Riplinger, J. C. Phillips, J. D. C. Maia, G. B. Rocha, J. V. Ribeiro, J. E. Stone, F. Neese, K. Schulten, and Z. Luthey-Schulten, *Nat. Methods* **15**, 351 (2018).
- ¹⁰²M. J. Field, P. A. Bash, and M. Karplus, *J. Comput. Chem.* **11**, 700 (1990).
- ¹⁰³M. J. Field, *J. Chem. Inf. Model.* **62**, 5849 (2022).
- ¹⁰⁴W. L. Jorgensen and J. Tirado-Rives, *J. Comput. Chem.* **26**, 1689 (2005).
- ¹⁰⁵G. d. M. Seabra, R. C. Walker, M. Elstner, D. A. Case, and A. E. Roitberg, *The Journal of Physical Chemistry A* **111**, 5655 (2007).
- ¹⁰⁶S. Dapprich, I. Komáromi, K. S. Byun, K. Morokuma, and M. J. Frisch, *J. Mol. Struct.: THEOCHEM* **461–462**, 1 (1999).
- ¹⁰⁷B. R. Brooks, C. L. B. III, A. D. M. Jr., L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoseck, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus, *J. Comput. Chem.* **30**, 1545 (2009).
- ¹⁰⁸D. A. Case, T. E. C. III, T. Darden, H. Gohlke, R. Luo, K. M. M. Jr., A. Onufriev, C. Simmerling, B. Wang, and R. J. Woods, *J. Comput. Chem.* **26**, 1668 (2005).
- ¹⁰⁹A. O. Dohn, *Int. J. Quantum Chem.* **120**, e26343 (2020).
- ¹¹⁰R. B. Murphy, D. M. Philipp, and R. A. Friesner, *J. Comput. Chem.* **21**, 1442 (2000).
- ¹¹¹T. K. Woo, P. M. Margl, P. E. Blöchl, and T. Ziegler, *J. Phys. Chem. B* **101**, 7877 (1997).
- ¹¹²T. K. Woo, L. Cavallo, and T. Ziegler, *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)* **100**, 307 (1998).
- ¹¹³Q. Cui, M. Elstner, E. Kaxiras, T. Frauenheim, and M. Karplus, *J. Phys. Chem. B* **105**, 569 (2001).

- ¹¹⁴M. J. Field, J. Chem. Theory Comput. **4**, 1151 (2008).
- ¹¹⁵C. R. Jacob, S. M. Beyhan, R. E. Bulo, A. S. P. Gomes, A. W. Götz, K. Kiewisch, J. Sikkema, and L. Visscher, J Comput Chem **32**, 2328 (2011).
- ¹¹⁶A. W. Götz, M. A. Clark, and R. C. Walker, J Comput Chem **35**, 95 (2014).
- ¹¹⁷Y. Takano, K. Nakata, Y. Yonezawa, and H. Nakamura, J Comput Chem **37**, 1125 (2016).
- ¹¹⁸M. Zheng and M. P. Waller, J Comput Chem **39**, 799 (2018).
- ¹¹⁹J. D. Samaniego-Rojas, L.-I. Hernandez, L. Lopez-Sosa, R. I. Delgado, B. Gomez, J.-C. Lambry, J. Alexandre, B. A. Zuniga-Gutierrez, R. Flores-Moreno, P. Calaminici, G. Geudtner, and A. M. Koster, in *Multiscale Dynamics Simulations: Nano and Nanobio Systems in Complex Environments*, Theoretical and Computational Chemistry No. 22, edited by D. R. Salahub and D. Wei (2022) pp. 1–54.
- ¹²⁰V. W. D. Cruzeiro, Y. Wang, E. Pieri, E. G. Hohenstein, and T. J. Martínez, The Journal of Chemical Physics **158**, 044801 (2023).
- ¹²¹Y. Pang and R. Bjornsson, Inorg. Chem. **62**, 5357 (2023).
- ¹²²B. Raghavan, F. K. Schackert, A. Levy, S. K. Johnson, E. Ippoliti, D. Mandelli, J. M. H. Olsen, U. Rothlisberger, and P. Carloni, J. Chem. Inf. Model. **63**, 1406 (2023).
- ¹²³S. Dohm, E. Spohr, and M. Korth, J Comput Chem **38**, 51 (2017).
- ¹²⁴Y. Shao, Z. Gan, E. Epifanovsky, A. T. Gilbert, M. Wormit, J. Kussmann, A. W. Lange, A. Behn, J. Deng, X. Feng, D. Ghosh, M. Goldey, P. R. Horn, L. D. Jacobson, I. Kaliman, R. Z. Khaliullin, T. Kuś, A. Landau, J. Liu, E. I. Proynov, Y. M. Rhee, R. M. Richard, M. A. Rohrdanz, R. P. Steele, E. J. Sundstrom, H. L. Woodcock, P. M. Zimmerman, D. Zuev, B. Albrecht, E. Alguire, B. Austin, G. J. O. Beran, Y. A. Bernard, E. Berquist, K. Brandhorst, K. B. Bravaya, S. T. Brown, D. Casanova, C.-M. Chang, Y. Chen, S. H. Chien, K. D. Closser, D. L. Crittenden, M. Diedenhofen, R. A. DiStasio, H. Do, A. D. Dutoi, R. G. Edgar, S. Fatehi, L. Fusti-Molnar, A. Ghysels, A. Golubeva-Zadorozhnaya, J. Gomes, M. W. Hanson-Heine, P. H. Harbach, A. W. Hauser, E. G. Hohenstein, Z. C. Holden, T.-C. Jagau, H. Ji, B. Kaduk, K. Khistyayev, J. Kim, J. Kim, R. A. King, P. Klunzinger, D. Kosenkov, T. Kowalczyk, C. M. Krauter, K. U. Lao, A. D. Laurent, K. V. Lawler, S. V. Levchenko, C. Y. Lin, F. Liu, E. Livshits, R. C. Lochan, A. Luenser, P. Manohar, S. F. Manzer, S.-P. Mao, N. Mardirossian, A. V. Marenich, S. A. Maurer, N. J. Mayhall, E. Neuscamman, C. M. Oana, R. Olivares-Amaya, D. P. O'Neill, J. A. Parkhill, T. M. Perrine, R. Peverati, A. Prociuk, D. R. Rehn, E. Rosta, N. J. Russ,

- S. M. Sharada, S. Sharma, D. W. Small, A. Sodt, T. Stein, D. Stück, Y.-C. Su, A. J. Thom, T. Tsuchimochi, V. Vanovschi, L. Vogt, O. Vydrov, T. Wang, M. A. Watson, J. Wenzel, A. White, C. F. Williams, J. Yang, S. Yeganeh, S. R. Yost, Z.-Q. You, I. Y. Zhang, X. Zhang, Y. Zhao, B. R. Brooks, G. K. Chan, D. M. Chipman, C. J. Cramer, W. A. Goddard, M. S. Gordon, W. J. Hehre, A. Klamt, H. F. Schaefer, M. W. Schmidt, C. D. Sherrill, D. G. Truhlar, A. Warshel, X. Xu, A. Aspuru-Guzik, R. Baer, A. T. Bell, N. A. Besley, J.-D. Chai, A. Dreuw, B. D. Dunietz, T. R. Furlani, S. R. Gwaltney, C.-P. Hsu, Y. Jung, J. Kong, D. S. Lambrecht, W. Liang, C. Ochsenfeld, V. A. Rassolov, L. V. Slipchenko, J. E. Subotnik, T. Van Voorhis, J. M. Herbert, A. I. Krylov, P. M. Gill, and M. Head-Gordon, *Molecular Physics* **113**, 184 (2015).
- ¹²⁵J. J. Mortensen, A. H. Larsen, M. Kuisma, A. V. Ivanov, A. Taghizadeh, A. Peterson, A. Haldar, A. O. Dohn, C. Schäfer, E. O. Jónsson, E. D. Hermes, F. A. Nilsson, G. Kastlunger, G. Levi, H. Jónsson, H. Häkkinen, J. Fojt, J. Kangsabanik, J. Sødequist, J. Lehtomäki, J. Heske, J. Enkovaara, K. T. Winther, M. Dulak, M. M. Melander, M. Ovesen, M. Louhivuori, M. Walter, M. Gjerding, O. Lopez-Acevedo, P. Erhart, R. Warmbier, R. Würdemann, S. Kaappa, S. Latini, T. M. Boland, T. Bligaard, T. Skovhus, T. Susi, T. Maxson, T. Rossi, X. Chen, Y. L. A. Schmerwitz, J. Schiøtz, T. Olsen, K. W. Jacobsen, and K. S. Thygesen, *The Journal of Chemical Physics* **160**, 092503 (2024).
- ¹²⁶J. P. Marcolongo, A. Zeida, J. A. Semelak, N. O. Foglia, U. N. Morzan, D. A. Estrin, M. C. González Lebrero, and D. A. Scherlis, *Front. Chem.* **6**, 70 (2018).
- ¹²⁷M. D. Elola, D. A. Estrin, and D. Laria, *J. Phys. Chem. A* **103**, 5105 (1999).
- ¹²⁸J. R. Shoemaker, Larry W. Burggraf*, and M. S. Gordon, *J. Phys. Chem. A* **103**, 3245 (1999).
- ¹²⁹C. Ma, L. Martin-Samos, S. Fabris, A. Laio, and S. Piccinin, *Computer Physics Communications* **195**, 191 (2015).
- ¹³⁰M. J. Field, M. Albe, C. Bret, F. Proust-De Martin, and A. Thomas, *J. Comput. Chem.* **21**, 1088 (2000).
- ¹³¹P. L. Cummins and J. E. Gready, *J Comput Chem* **15**, 704 (1994).
- ¹³²A. De La Lande, A. Alvarez-Ibarra, K. Hasnaoui, F. Cailliez, X. Wu, T. Mineva, J. Cuny, P. Calaminici, L. López-Sosa, G. Geudtner, I. Navizet, C. Garcia Iriepa, D. R. Salahub, and A. M. Köster, *Molecules* **24**, 1653 (2019).

- ¹³³R. A. Kendall, E. Aprà, D. E. Bernholdt, E. J. Bylaska, M. Dupuis, G. I. Fann, R. J. Harrison, J. Ju, J. A. Nichols, J. Nieplocha, T. Straatsma, T. L. Windus, and A. T. Wong, *Computer Physics Communications* **128**, 260 (2000).
- ¹³⁴T. Janowski, K. Wolinski, and P. Pulay, *Chemical Physics Letters* **530**, 1 (2012).
- ¹³⁵N. M. Thellamurege, D. Si, F. Cui, H. Zhu, R. Lai, and H. Li, *J. Comput. Chem.* **34**, 2816 (2013).
- ¹³⁶T. Kubař, K. Welke, and G. Groenhof, *J. Comput. Chem.* **36**, 1978 (2015).
- ¹³⁷G. Te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. Van Gisbergen, J. G. Snijders, and T. Ziegler, *J Comput Chem* **22**, 931 (2001).
- ¹³⁸H. L. Woodcock III, M. Hodoscek, A. T. B. Gilbert, P. M. W. Gill, H. F. Schaefer III, and B. R. Brooks, *J. Comput. Chem.* **28**, 1485 (2007).
- ¹³⁹T. Okamoto, K. Yamada, Y. Koyano, T. Asada, N. Koga, and M. Nagaoka, *J Comput Chem* **32**, 932 (2011).
- ¹⁴⁰K. Meier, N. Schmid, and W. F. Van Gunsteren, *J Comput Chem* **33**, 2108 (2012).
- ¹⁴¹M. C. Colombo, L. Guidoni, A. Laio, A. Magistrato, S. Piana, U. Röhrig, K. Spiegel, M. Sulpizi, J. VandeVondele, M. Zumstein, and U. Röhrlisberger, *Chimia* **56**, 13 (2002).
- ¹⁴²M. J. Loferer, H. H. Loeffler, and K. R. Liedl, *J Comput Chem* **24**, 1240 (2003).
- ¹⁴³D. Salahub, S. Noskov, B. Lev, R. Zhang, V. Ngo, A. Goursot, P. Calaminici, A. Köster, A. Alvarez-Ibarra, D. Mejía-Rodríguez, J. Řezáč, F. Cailliez, and A. De La Lande, *Molecules* **20**, 4780 (2015).
- ¹⁴⁴S. K. Sahoo and N. N. Nair, *J Comput Chem* **37**, 1657 (2016).
- ¹⁴⁵B. Hégyel, F. Bogár, G. G. Ferenczy, and M. Kállay, *Theor Chem Acc* **134**, 132 (2015).
- ¹⁴⁶P. K. Biswas and V. Gogonea, *The Journal of Chemical Physics* **123**, 164114 (2005).
- ¹⁴⁷B. L. Grigorenko, A. V. Nemukhin, I. A. Topol, and S. K. Burt, *J. Phys. Chem. A* **106**, 10663 (2002).
- ¹⁴⁸P. D. Lyne, M. Hodoscek, and M. Karplus, *J. Phys. Chem. A* **103**, 3462 (1999).
- ¹⁴⁹U. Ryde, *J Computer-Aided Mol Des* **10**, 153 (1996).
- ¹⁵⁰U. Ryde and M. H. M. Olsson, *Int. J. Quantum Chem.* **81**, 335 (2001).
- ¹⁵¹J. Sauer and M. Sierka, *J. Comput. Chem.* **21**, 1470 (2000).
- ¹⁵²P. Sherwood, A. H. De Vries, M. F. Guest, G. Schreckenbach, C. R. A. Catlow, S. A. French, A. A. Sokol, S. T. Bromley, W. Thiel, A. J. Turner, S. Billeter, F. Terstegen, S. Thiel, J. Kendrick, S. C. Rogers, J. Casci, M. Watson, F. King, E. Karlsen, M. Sjøvoll,

- A. Fahmi, A. Schäfer, and C. Lennartz, *Journal of Molecular Structure* **632**, 1 (2003).
- ¹⁵³J. N. Harvey, *Faraday Disc.* **127**, 165 (2004).
- ¹⁵⁴S. Metz, J. Kästner, A. A. Sokol, T. W. Keal, and P. Sherwood, *WIREs Comput Mol Sci* **4**, 101 (2014).
- ¹⁵⁵Y. Lu, M. R. Farrow, P. Fayon, A. J. Logsdail, A. A. Sokol, C. R. A. Catlow, P. Sherwood, and T. W. Keal, *J. Chem. Theory Comput.* **15**, 1217 (2019).
- ¹⁵⁶Y. Lu, K. Sen, C. Yong, D. S. D. Gunn, J. A. Purton, J. Guan, A. Desmoutier, J. Abdul Nasir, X. Zhang, L. Zhu, Q. Hou, J. Jackson-Masters, S. Watts, R. Hanson, H. N. Thomas, O. Jayawardena, A. J. Logsdail, S. M. Woodley, H. M. Senn, P. Sherwood, C. R. A. Catlow, A. A. Sokol, and T. W. Keal, *Phys. Chem. Chem. Phys.* **25**, 21816 (2023).
- ¹⁵⁷H. Lin and D. G. Truhlar, *J. Phys. Chem. A* **109**, 3991 (2005).
- ¹⁵⁸H. Lin, Y. Zhang, S. Pezeshki, A. W. Duster, B. Wang, X.-P. Wu, S.-W. Zheng, L. Gagliardi, and D. G. Truhlar, *Comput. Phys. Commun.* **295**, 108987 (2024).
- ¹⁵⁹P. Altoè, M. Stenta, A. Bottoni, and M. Garavelli, *Theor Chem Account* **118**, 219 (2007).
- ¹⁶⁰J. Torras, E. Deumens, and S. B. Trickey, *J Computer-Aided Mater Des* **13**, 201 (2006).
- ¹⁶¹J. Torras, Y. He, C. Cao, K. Muralidharan, E. Deumens, H.-P. Cheng, and S. Trickey, *Computer Physics Communications* **177**, 265 (2007).
- ¹⁶²J. Torras, B. P. Roberts, G. M. Seabra, and S. B. Trickey, in *Advances in Protein Chemistry and Structural Biology*, Vol. 100 (Elsevier, 2015) pp. 1–31.
- ¹⁶³E. G. Kratz, A. R. Walker, L. Lagardère, F. Lipparini, J.-P. Piquemal, and A. G. Cisneros, *J. Comput. Chem.* **37**, 1019 (2016).
- ¹⁶⁴J. Řezáč, *J. Comput. Chem.* **37**, 1230 (2016).
- ¹⁶⁵H. Gökcan, E. A. Vázquez-Montelongo, and G. A. Cisneros, *J. Chem. Theory Comput.* **15**, 3056 (2019).
- ¹⁶⁶P. Seeber, S. Seidenath, J. Steinmetzer, and S. Gräfe, *WIREs Comput Mol Sci* **13**, e1644 (2023).
- ¹⁶⁷B. Zhang, D. Altarawy, T. Barnes, J. M. Turney, and H. F. S. III, *J. Chem. Theory Comput.* **15**, 4362 (2019).
- ¹⁶⁸S. Martí, *J. Comput. Chem.* **42**, 393 (2021).
- ¹⁶⁹J. M. H. Olsen, V. Bolnykh, S. Meloni, E. Ippoliti, M. P. Bircher, P. Carloni, and U. Rothlisberger, *J. Chem. Theory Comput.* **15**, 3810 (2019).

- ¹⁷⁰A. Antalík, A. Levy, S. Kvedaravičiūtė, S. K. Johnson, D. Carrasco-Busturia, B. Raghavan, F. Mouvet, A. Acocella, S. Das, V. Gavini, D. Mandelli, E. Ippoliti, S. Meloni, P. Carloni, U. Rothlisberger, and J. M. H. Olsen, MiMiC: A High-Performance Framework for Multiscale Molecular Dynamics Simulations (2024), arXiv:2403.19035 [physics].
- ¹⁷¹T. A. Barnes, E. Marin-Rimoldi, S. Ellis, and T. D. Crawford, *Comput. Phys. Commun.* **261**, 107688 (2021).
- ¹⁷²S. Lehtola, *The Journal of Chemical Physics* **159**, 180901 (2023).
- ¹⁷³G. A. Kaminski and W. L. Jorgensen, *J. Phys. Chem. B* **102**, 1787 (1998).
- ¹⁷⁴T. Vreven and K. Morokuma, *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)* **109**, 125 (2003).
- ¹⁷⁵T. Vreven, K. S. Byun, I. Komáromi, S. Dapprich, J. A. Montgomery, K. Morokuma, and M. J. Frisch, *J. Chem. Theory Comput.* **2**, 815 (2006).
- ¹⁷⁶M. J. T. Oliveira, N. Papior, Y. Pouillon, V. Blum, E. Artacho, D. Caliste, F. Corsetti, S. de Gironcoli, A. M. Elena, A. García, V. M. García-Suárez, L. Genovese, W. P. Huhn, G. Huhs, S. K. E. Küçükbenli, A. H. Larsen, A. Lazzaro, I. V. Lebedeva, Y. Li, D. López-Durán, P. López-Tarifa, M. Lüders, M. A. L. Marques, J. Minar, S. Mohr, A. A. Mostofi, A. O'Cais, M. C. Payne, T. Ruh, D. G. A. Smith, J. M. Soler, D. A. Strubbe, N. Tancogne-Dejean, D. Tildesley, M. Torrent, and V. W. zhe Yu, *J. Chem. Phys.* **153**, 024117 (2020).
- ¹⁷⁷A. H. Larsen, J. J. Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. B. Jensen, J. Kermode, J. R. Kitchin, E. L. Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. B. Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. Schiøtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, , and K. W. Jacobsen, *J. Phys.: Condens. Matter* **29**, 273002 (2017).
- ¹⁷⁸J. M. Turney, A. C. Simmonett, R. M. Parrish, E. G. Hohenstein, F. A. Evangelista, J. T. Fermann, B. J. Mintz, L. A. Burns, J. J. Wilke, M. L. Abrams, N. J. Russ, M. L. Leininger, C. L. Janssen, E. T. Seidl, W. D. Allen, H. F. Schaefer, R. A. King, E. F. Valeev, C. D. Sherrill, and T. D. Crawford, *WIREs Comput Mol Sci* **2**, 556 (2012).
- ¹⁷⁹R. M. Parrish, L. A. Burns, D. G. A. Smith, A. C. Simmonett, A. E. DePrince, E. G. Hohenstein, U. Bozkaya, A. Y. Sokolov, D. R. Remigio, R. M. Richard, J. F. Gonthier, A. M. James, H. R. McAlexander, A. Kumar, M. Saitow, X. Wang, B. P. Pritchard, P. Verma, H. F. Schaefer, K. Patkowski, R. A. King, E. F. Valeev, F. A. Evangelista,

- J. M. Turney, T. D. Crawford, and C. D. Sherrill, *J. Chem. Theory Comput.* **13**, 3185 (2017).
- ¹⁸⁰Q. Sun, T. C. Berkelbach, N. S. Blunt, G. H. Booth, S. Guo, Z. Li, J. Liu, J. D. McClain, E. R. Sayfutyarova, S. Sharma, S. Wouters, and G. K.-L. Chan, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **8**, e1340 (2017).
- ¹⁸¹P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp., L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, R. P. Wiewiora, B. R. Brooks, and V. S. Pande, *PLoS Comput. Biol.* **13**, e1005659 (2017).
- ¹⁸²D. G. A. Smith, L. A. Burns, D. A. Sirianni, D. R. Nascimento, A. Kumar, A. M. James, J. B. Schriber, T. Zhang, B. Zhang, A. S. Abbott, E. J. Berquist, M. H. Lechner, L. A. Cunha, A. G. Heide, J. M. Waldrop, T. Y. Takeshita, A. Alenaizan, D. Neuhauser, R. A. King, A. C. Simmonett, J. M. Turney, H. F. Schaefer, F. A. Evangelista, A. E. D. III, T. D. Crawford, K. Patkowski, and C. D. Sherrill, *J. Chem. Theory Comput.* **14**, 3504 (2018).
- ¹⁸³J. V. Alegre-Requena, S. Sowndarya S. V., R. Pérez-Soto, T. M. Alturaifi, and R. S. Paton, *WIREs Comput Mol Sci* **13**, e1663 (2023).
- ¹⁸⁴S. Lehtola and A. J. Karttunen, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **12**, e1610 (2022).
- ¹⁸⁵R. D. Felice, M. L. Mayes, R. M. Richard, D. B. Williams-Young, G. K.-L. Chan, W. A. de Jong, N. Govind, M. Head-Gordon, M. R. Hermes, K. Kowalski, X. Li, H. Lischka, K. T. Mueller, E. Mutlu, A. M. N. Niklasson, M. R. Pederson, B. Peng, R. Shepard, E. F. Valeev, M. van Schilfgaarde, B. Vlaisavljevich, T. L. Windus, S. S. Xantheas, X. Zhang, and P. M. Zimmerman, *J. Chem. Theory Comput.* **19**, 7056 (2023).
- ¹⁸⁶P. Eastman and V. Pande, *Comput. Sci. Eng.* **12**, 34 (2010).
- ¹⁸⁷D. G. A. Smith, L. A. Burns, A. C. Simmonett, R. M. Parrish, M. C. Schieber, R. Galvelis, P. Kraus, H. Kruse, R. Di Remigio, A. Alenaizan, A. M. James, S. Lehtola, J. P. Misiewicz, M. Scheurer, R. A. Shaw, J. B. Schriber, Y. Xie, Z. L. Glick, D. A. Sirianni, J. S. O'Brien, J. M. Waldrop, A. Kumar, E. G. Hohenstein, B. P. Pritchard, B. R. Brooks, H. F. Schaefer, A. Y. Sokolov, K. Patkowski, A. E. DePrince, U. Bozkaya, R. A. King, F. A. Evangelista, J. M. Turney, T. D. Crawford, and C. D. Sherrill, *The Journal of Chemical Physics* **152**, 184108 (2020).
- ¹⁸⁸P. Eastman, M. S. Friedrichs, J. D. Chodera, R. J. Radmer, C. M. Bruns, J. P. Ku, K. A.

- Beauchamp, T. J. Lane, L.-P. Wang, D. Shukla, T. Tye, M. Houston, T. Stich, C. Klein, M. R. Shirts, and V. S. Pande, *J. Chem. Theory Comput.* **9**, 461 (2013).
- ¹⁸⁹U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen, *Journal of Chemical Physics* **103**, 8577 (1995).
- ¹⁹⁰L. Cao and U. Ryde, *Frontiers in Chemistry* **6** (2018).
- ¹⁹¹M. A. Thompson and G. K. Schenter, *J. Phys. Chem.* **99**, 6374 (1995).
- ¹⁹²M. A. Thompson, *J. Phys. Chem.* **100**, 14492 (1996).
- ¹⁹³M. Field, *Mol. Phys.* **91**, 835 (1997).
- ¹⁹⁴J. Gao, *J. Comput. Chem.* **18**, 1061 (1997).
- ¹⁹⁵J. A. Gascon, S. S. F. Leung, E. R. Batista, and V. S. Batista, *J. Chem. Theory Comput.* **2**, 175 (2006).
- ¹⁹⁶C. Curutchet, A. Muñoz-Losa, S. Monti, J. Kongsted, G. D. Scholes, and B. Mennucci, *J. Chem. Theory Comput.* **5**, 1838 (2009).
- ¹⁹⁷A. Ganguly, E. Boulanger, and W. Thiel, *J. Chem. Theory Comput.* **13**, 2954 (2017).
- ¹⁹⁸J. Dziedzic, T. Head-Gordon, M. Head-Gordon, and C.-K. Skylaris, *The Journal of Chemical Physics* **150**, 074103 (2019).
- ¹⁹⁹T. Giovannini, A. Puglisi, M. Ambrosetti, and C. Cappelli, *J. Chem. Theory Comput.* **15**, 2233 (2019).
- ²⁰⁰A. Humeniuk and W. J. Glover, *J. Chem. Theory Comput.* **20**, 2111 (2024).
- ²⁰¹E. Lambros, F. Lipparini, G. A. Cisneros, and F. Paesani, *J. Chem. Theory Comput.* **16**, 7462 (2020).
- ²⁰²N. Bernstein, J. R. Kermode, and G. Csányi, *Rep. Prog. Phys.* **72**, 026501 (2009).
- ²⁰³V. A. Nasluzov, E. A. Ivanova, A. M. Shor, G. N. Vayssilov, U. Birkenheuer, and N. Rösch, *J. Phys. Chem. B* **107**, 2228 (2003).
- ²⁰⁴A. Redondo, W. A. Goddard, C. A. Swarts, and T. C. McGill, *Journal of Vacuum Science and Technology* **19**, 498 (1981).
- ²⁰⁵U. Ryde, in *Methods in Enzymology*, Vol. 577 (Elsevier, 2016) pp. 119–158.
- ²⁰⁶J. Gao, P. Amara, C. Alhambra, and M. J. Field, *J. Phys. Chem. A* **102**, 4714 (1998).
- ²⁰⁷D. M. Philipp and R. A. Friesner, *J. Comput. Chem.* **20**, 1468 (1999).
- ²⁰⁸V. Théry, D. Rinaldi, J. Rivail, B. Maigret, and G. G. Ferenczy, *J Comput Chem* **15**, 269 (1994).
- ²⁰⁹V. Kairys and J. H. Jensen, *J. Phys. Chem. A* **104**, 6656 (2000).

- ²¹⁰J. Pu, J. Gao, and D. G. Truhlar, *J. Phys. Chem. A* **108**, 632 (2004).
- ²¹¹A. Monari, J.-L. Rivail, and X. Assfeld, *Acc. Chem. Res.* **46**, 596 (2013).
- ²¹²J. Jung, C. H. Choi, Y. Sugita, and S. Ten-no, *The Journal of Chemical Physics* **127**, 204102 (2007).
- ²¹³Y. Wang and J. Gao, *J. Phys. Chem. B* **119**, 1213 (2015).
- ²¹⁴F. Alary, R. Poteau, J.-L. Heully, J.-C. Barthelat, and J.-P. Daudey, *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)* **104**, 174 (2000).
- ²¹⁵G. A. DiLabio, M. M. Hurley, and P. A. Christiansen, *The Journal of Chemical Physics* **116**, 9578 (2002).
- ²¹⁶F. Bessac, F. Alary, Y. Carissan, J.-L. Heully, J.-P. Daudey, and R. Poteau, *Journal of Molecular Structure: THEOCHEM* **632**, 43 (2003).
- ²¹⁷A. Mallik, D. E. Taylor, K. Runge, and J. W. Dufty, *Int J of Quantum Chemistry* **100**, 1019 (2004).
- ²¹⁸S. Moon, P. A. Christiansen, and G. A. DiLabio, *The Journal of Chemical Physics* **120**, 9080 (2004).
- ²¹⁹K. Yasuda and D. Yamaki, *The Journal of Chemical Physics* **121**, 3964 (2004).
- ²²⁰O. A. Von Lilienfeld, I. Tavernelli, U. Rothlisberger, and D. Sebastiani, *The Journal of Chemical Physics* **122**, 014113 (2005).
- ²²¹P. Slavíček and T. J. Martínez, *The Journal of Chemical Physics* **124**, 084107 (2006).
- ²²²N. Jardillier and A. Goursot, *Chemical Physics Letters* **454**, 65 (2008).
- ²²³J. L. Lewin and C. J. Cramer, *J. Phys. Chem. A* **112**, 12754 (2008).
- ²²⁴Y.-y. Ohnishi, Y. Nakao, H. Sato, and S. Sakaki, *J. Phys. Chem. A* **112**, 1946 (2008).
- ²²⁵S. Komin and D. Sebastiani, *J. Chem. Theory Comput.* **5**, 1490 (2009).
- ²²⁶D. Bakowies and W. Thiel, *J. Phys. Chem.* **100**, 10580 (1996).
- ²²⁷P. Amara and M. J. Field, *Theoretical Chemistry Accounts: Theory, Computation, and Modeling (Theoretica Chimica Acta)* **109**, 43 (2003).
- ²²⁸M. Hitzenberger and T. S. Hofer, *J. Comput. Chem.* **36**, 1929 (2015).
- ²²⁹N. Koga and K. Morokuma, *Chemical Physics Letters* **172**, 243 (1990).
- ²³⁰I. Antes and W. Thiel, *J. Phys. Chem. A* **103**, 9290 (1999).
- ²³¹E. Dumont and P. Chaquin, *Journal of Molecular Structure: THEOCHEM* **680**, 99 (2004).
- ²³²A. Toniolo, C. Ciminelli, G. Granucci, T. Laino, and M. Persico, *Theor Chem Acc* **111**,

- 270 (2004).
- ²³³E. Dumont and P. Chaquin, Journal of Molecular Structure: THEOCHEM **758**, 161 (2006).
- ²³⁴E. Dumont and P. Chaquin, Chemical Physics Letters **435**, 354 (2007).
- ²³⁵Y. Zhang, T.-S. Lee, and W. Yang, The Journal of Chemical Physics **110**, 46 (1999).
- ²³⁶Y. Zhang, The Journal of Chemical Physics **122**, 024114 (2005).
- ²³⁷J. M. Parks, H. Hu, A. J. Cohen, and W. Yang, The Journal of Chemical Physics **129**, 154106 (2008).
- ²³⁸B. Wang and D. G. Truhlar, J. Chem. Theory Comput. **6**, 359 (2010).
- ²³⁹X.-P. Wu, L. Gagliardi, and D. Truhlar, Molecules **23**, 1309 (2018).
- ²⁴⁰X.-P. Wu, L. Gagliardi, and D. G. Truhlar, J. Chem. Theory Comput. **15**, 4208 (2019).
- ²⁴¹S. Pezeshki and H. Lin, J. Chem. Theory Comput. **11**, 2398 (2015).
- ²⁴²C. N. Rowley and B. Roux, J. Chem. Theory Comput. **8**, 3526 (2012).
- ²⁴³M. Shiga and M. Masia, The Journal of Chemical Physics **139**, 044120 (2013).
- ²⁴⁴M. Shiga and M. Masia, Molecular Simulation **41**, 827 (2015).
- ²⁴⁵H. Takahashi, H. Kambe, and A. Morita, The Journal of Chemical Physics **148**, 134119 (2018).
- ²⁴⁶H. Takahashi, H. Kambe, and A. Morita, The Journal of Chemical Physics **150**, 114109 (2019).
- ²⁴⁷Z. Shen and W. J. Glover, The Journal of Chemical Physics **155**, 224112 (2021).
- ²⁴⁸B. Kirchhoff, E. O. Jónsson, A. O. Dohn, T. Jacob, and H. Jónsson, J. Chem. Theory Comput. **17**, 5863 (2021).
- ²⁴⁹T. Kerdcharoen, K. R. Liedl, and B. M. Rode, Chemical Physics **211**, 313 (1996).
- ²⁵⁰T. Kerdcharoen and K. Morokuma, Chemical Physics Letters **355**, 257 (2002).
- ²⁵¹A. Heyden, H. Lin, and D. G. Truhlar, J. Phys. Chem. B **111**, 2231 (2007).
- ²⁵²R. E. Bulo, B. Ensing, J. Sikkema, and L. Visscher, J. Chem. Theory Comput. **5**, 2212 (2009).
- ²⁵³N. Bernstein, C. Várnai, I. Solt, S. A. Winfield, M. C. Payne, I. Simon, M. Fuxreiter, and G. Csányi, Phys. Chem. Chem. Phys. **14**, 646 (2012).
- ²⁵⁴N. Takenaka, Y. Kitamura, Y. Koyano, and M. Nagaoka, Chemical Physics Letters **524**, 56 (2012).
- ²⁵⁵M. P. Waller, S. Kumbhar, and J. Yang, ChemPhysChem **15**, 3218 (2014).

- ²⁵⁶H. C. Watanabe, T. Kubař, and M. Elstner, J. Chem. Theory Comput. **10**, 4242 (2014).
- ²⁵⁷M. Zheng and M. P. Waller, WIREs Comput Mol Sci **6**, 369 (2016).
- ²⁵⁸A. W. Duster, C. Wang, C. M. Garza, D. E. Miller, and H. Lin, WIREs Comput Mol Sci **7**, e1310 (2017).
- ²⁵⁹M. J. Field, J. Chem. Theory Comput. **13**, 2342 (2017).
- ²⁶⁰A. Duster, C.-H. Wang, and H. Lin, Molecules **23**, 2170 (2018).
- ²⁶¹H. Watanabe, Molecules **23**, 1882 (2018).
- ²⁶²H. C. Watanabe and Q. Cui, J. Chem. Theory Comput. **15**, 3917 (2019).
- ²⁶³J.-N. Wang, W. Liu, P. Li, Y. Mo, W. Hu, J. Zheng, X. Pan, Y. Shao, and Y. Mei, J. Chem. Theory Comput. **17**, 1318 (2021).
- ²⁶⁴S. Pezeshki and H. Lin, J. Chem. Theory Comput. **7**, 3625 (2011).
- ²⁶⁵S. Pezeshki, C. Davis, A. Heyden, and H. Lin, J. Chem. Theory Comput. **10**, 4765 (2014).
- ²⁶⁶M. Zheng, J. A. Kuriappan, and M. P. Waller, Int J of Quantum Chemistry **117**, e25336 (2017).
- ²⁶⁷J. M. Boereboom, R. Potestio, D. Donadio, and R. E. Bulo, J. Chem. Theory Comput. **12**, 3441 (2016).
- ²⁶⁸M. Böckmann, N. L. Doltsinis, and D. Marx, J. Chem. Theory Comput. **11**, 2429 (2015).
- ²⁶⁹H. Chen, M. Liao, H. Wang, Y. Wang, and L. Zhang, Computer Methods in Applied Mechanics and Engineering **354**, 351 (2019).
- ²⁷⁰Z.-h. Yang, Phys. Chem. Chem. Phys. **22**, 19307 (2020).
- ²⁷¹Z.-h. Yang, Phys. Chem. Chem. Phys. **22**, 17987 (2020).
- ²⁷²J. Mato, A. W. Duster, E. B. Guidez, and H. Lin, J. Chem. Theory Comput. **17**, 5456 (2021).
- ²⁷³Z.-h. Yang, Phys. Chem. Chem. Phys. **23**, 3417 (2021).
- ²⁷⁴O. Glukhova, G. Savostyanov, and M. Slepchenkov, Procedia Materials Science **6**, 256 (2014).
- ²⁷⁵R. E. Bulo, C. Michel, P. Fleurat-Lessard, and P. Sautet, J. Chem. Theory Comput. **9**, 5567 (2013).
- ²⁷⁶E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software* (Addison-Wesley, Reading, MA, 1995).
- ²⁷⁷C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design* (Prentice Hall, Upper Saddle River, NJ, 1998).

- ²⁷⁸R. C. Martin, *Object Mentor* **1**, 597 (2000).
- ²⁷⁹M. Bonomi, G. Bussi, C. Camilloni, G. A. Tribello, P. Banáš, A. Barducci, M. Bernetti, P. G. Bolhuis, S. Bottaro, D. Branduardi, R. Capelli, P. Carloni, M. Ceriotti, A. Cesari, H. Chen, W. Chen, F. Colizzi, S. De, M. De La Pierre, D. Donadio, V. Drobot, B. Ensing, A. L. Ferguson, M. Filizola, J. S. Fraser, H. Fu, P. Gasparotto, F. L. Gervasio, F. Giberti, A. Gil-Ley, T. Giorgino, G. T. Heller, G. M. Hocky, M. Iannuzzi, M. Invernizzi, K. E. Jelfs, A. Jussupow, E. Kirilin, A. Laio, V. Limongelli, K. Lindorff-Larsen, T. Löhr, F. Marinelli, L. Martin-Samos, M. Masetti, R. Meyer, A. Michaelides, C. Molteni, T. Morishita, M. Nava, C. Paissoni, E. Papaleo, M. Parrinello, J. Pfaendtner, P. Piaggi, G. Piccini, A. Pietropaolo, F. Pietrucci, S. Pipolo, D. Provasi, D. Quigley, P. Raiteri, S. Raniolo, J. Rydzewski, M. Salvalaglio, G. C. Sosso, V. Spiwok, J. Šponer, D. W. H. Swenson, P. Tiwary, O. Valsson, M. Vendruscolo, G. A. Voth, A. White, and P. c. The, *Nature Methods* **16**, 670 (2019).
- ²⁸⁰S. Miyamoto and P. A. Kollman, *J Comput Chem* **13**, 952 (1992).
- ²⁸¹R. N. Tazhigulov and K. B. Bravaya, *The Journal of Physical Chemistry Letters* **7**, 2490 (2016).
- ²⁸²W. Humphrey, A. Dalke, and K. Schulten, *Journal of Molecular Graphics* **14**, 33 (1996).
- ²⁸³A. J. Misquitta, *Journal of Chemical Theory and Computation* **9**, 5313 (2013).
- ²⁸⁴K. Senthilkumar, J. I. Mujika, K. E. Ranaghan, F. R. Manby, A. J. Mulholland, and J. N. Harvey, *Journal of The Royal Society Interface* **5**, 207 (2008).
- ²⁸⁵D. Riccardi, G. Li, and Q. Cui, *J. Phys. Chem. B* **108**, 6467 (2004).
- ²⁸⁶M. Freindorf and J. Gao, *J. Comput. Chem.* **17**, 386 (1996).
- ²⁸⁷K. Park, A. W. Götz, R. C. Walker, and F. Paesani, *Journal of Chemical Theory and Computation* **8**, 2868 (2012).
- ²⁸⁸I. C. Lin, A. P. Seitsonen, I. Tavernelli, and U. Rothlisberger, *Journal of Chemical Theory and Computation* **8**, 3902 (2012).
- ²⁸⁹J. H. Hymel, S. N. Khan, J. P. Pederson, and J. G. McDaniel, *J. Phys. Chem. C* **127**, 19489 (2023).
- ²⁹⁰PyDFT-QMMM.
- ²⁹¹J. A. Pople, R. Krishnan, H. B. Schlegel, and J. S. Binkley, *Int. J. Quantum Chem.* **16**, 225 (1979).
- ²⁹²B. G. Johnson, P. M. W. Gill, and J. A. Pople, *The Journal of Chemical Physics* **98**, 5612

This is the author's peer reviewed, accepted manuscript. However, the online version of record will be different from this version once it has been copyedited and typeset.

PLEASE CITE THIS ARTICLE AS DOI: 10.1063/5.0219851

(1993).

²⁹³B. G. Johnson, P. M. Gill, and J. A. Pople, Chemical Physics Letters **220**, 377 (1994).

²⁹⁴J. Baker, J. Andzelm, A. Scheiner, and B. Delley, The Journal of Chemical Physics **101**, 8894 (1994).