PUPIL: a Software Integration System for Multi-scale QM/MM-MD Simulations and its Application to Biomolecular Systems

Juan Torras, [1 *] *B.P. Roberts,* [2] *G.M. Seabra,* [3] *and S.B. Trickey* [4 *]

[1]Department of Chemical Engineering, EEI, Universitat Politècnica de Catalunya, Av. Pla de la Massa 8, Igualada 08700, Spain

[2] Centre for eResearch, The University of Auckland, Private Bag 92019, Auckland 1142 New Zealand

[3]Departamento de Química Fundamental, Universidade Federal de Pernambuco, Cidade Universitária, Recife, Pernambuco CEP 50.740-540, Brazil

[4]Department of Physics, Department of Chemistry, and Quantum Theory Project, Univ. of Florida, Gainesville Florida 32611-8435, USA

* Corresponding Authors: joan.torras@upc.edu and trickey@qtp.ufl.edu

Abstract

PUPIL (Program for User Package Interfacing and Linking) implements a distinctive multi-scale approach to hybrid quantum mechanical/molecular mechanical molecular dynamics (QM/MM-MD) simulations. Originally developed to interface different external programs for multi-scale simulation with applications in the materials sciences, PUPIL is finding increasing use in the study of complex biological systems. Advanced MD techniques from the external packages can be applied readily to a hybrid QM/MM treatment in which the forces and energy for the QM region can be computed by any of the QM methods available in any of the other external packages. Here we give a survey of PUPIL design philosophy, main features, and key implementation decisions, with an orientation to biomolecular simulation. We discuss recently implemented features which enable highly realistic simulations of complex biological systems which have more than one active site that must be treated concurrently. Examples are given.

Keywords: multi-scale simulation; hybrid methods; QM/MM; computational biology; biomolecular modeling; multiple actives zones; QM/MM-MD; PUPIL.

1. Introduction

One of the most remarkable but under-appreciated insights of contemporary molecular and condensed phase research is that the atomic constituents of molecules behave in many ways according to the classical mechanics of Newton. A related but also remarkable recognition is that molecules themselves often behave in an essentially classical way. Thus one has, for a single example, the concept of docking, which operationally is purely classical mechanics. The atomic-level insight had been reached by condensed matter and materials physicists (Alder & Wainwright, 1959; Rahman, 1964) over 50 years ago but in the setting of drastically simpler systems than those found in biology. They converted the insight into a powerful computational tool by recognizing, in essence, that classical statistical mechanics for the microcanonical ensemble amounts operationally to simply moving members of a population of interacting particles via Newton's Second Law for a substantial time interval and averaging the resultant energetics to yield the system thermodynamics. (This assumes ergodicity, one of several subtleties we leave aside for the purpose of broad perspective.)

These ideas were anticipated, of course, by molecular vibration theory and analysis on the chemistry side and the corresponding "lattice dynamics" or phonon treatment in the physics of ordered solids. Both start with a "ball and stick" model of the system, with the "balls" representing atoms and the "sticks" representing bonds (and, in a scale model, bond lengths). Such models trace to August Wilhelm von Hofmann. (Meinel, 1992) Vibrational analyses then replace the sticks with suitably chosen springs, corrected, if need be, by low-order anharmonicities. Common to both vibrational analyses and molecular dynamics (MD) is the assumption of a potential energy surface or force field. Given the existence of bond-length data, it is an obvious step to calibrate the force field so as to recover the bond-lengths at the energy minimum.

In a vernacular way, the summary just given comprises molecular mechanics (MM). MM is the treatment of molecular systems by methods of classical mechanics. Statics, i.e. rigid systems,

may be enough to yield valuable insight for the huge (by the standards of materials physics) molecules of biological importance. Elementary docking again provides an example: can two rigid structures be fit together in a snug way? MD adds the classical dynamics to get at the thermodynamics.

Computational approaches to the description of chemical systems in fact fall into two broad categories. The essential differences lie in assumptions about chemical interactions and bonding. MM is one category. MM schemes attempt to describe chemical systems via simple mathematical models for the forces between classical objects. Common aspects of these models (the force fields) include: (1) description of bonds themselves, as well as the angles between bonds to the same atom, as simple harmonic oscillators (SHOs); (2) inclusion of short-range electrostatic interactions using the Coulomb inverse-square law directly; (3) inclusion of long-range electrostatic (Coulomb) interactions using a Particle-mesh Ewald (PME) summation;(Darden, York, & Pedersen, 1993) and (4) description of van der Waals (dispersion) interactions via one of several Lennard-Jones type potentials.

All MM approaches, whether static or MD, are useful primarily for simulating large chemical systems containing $10^3 - 10^5$ or more particles, because the computational tasks are relatively simple: calculation of the forces and temporally discretized solution to Newton's equation. However, the descriptions of chemical systems are of limited accuracy. All of the chemistry is in the force field, so it determines the quality of MM calculations. For simple liquids such as Ar, an equally simple Lennard-Jones form goes a long way toward providing realistic thermodynamics from MD. But for biomolecular systems, construction of reliable potentials has proven to be a scientific craft of its own. See for example the literature of AMBER (Case et al., 2006) and CHARMM. (Brooks et al., 2009) Moreover, the mathematical models used are radically simplified. At most the chemical system composition is that of indivisible atoms. In a united-atom or coarse-grained simulation, composite particles (e.g. rigid molecules) are used.

A pure MM description of a system cannot capture perturbations in electron density reliably and accurately, much less give a uniformly accurate description of the vitally important chemical processes of bond breaking and formation, i.e., chemical reactivity. Moreover, a system can only be described using an MM model if the definition of the system includes all necessary variables (such as bonds, angles, and so forth) and the chosen force field contains the associated parameters. Parameters in this context are coefficients in the force field equations whose precise values vary depending upon the particular types of atoms involved in the bond, angle, torsion, or non-bonding pair of atoms. If any variable is missing, the MM model is somewhere between intrinsically inaccurate and completely irrelevant. If any variable is defined in the topology but its corresponding parameters are missing from the force field, the MM model simply cannot be solved.

Chemical reactivity is critical in all condensed phase processes, including biological ones. As illustrated by the structure of the periodic table, chemical reactivity intrinsically arises from quantum mechanics (QM), which identifies the second category of approach. A QM approach to computing the properties of a chemical system typically describes the system in terms of molecular orbitals, electron densities, and, sometimes, low-order density matrices. In addition to handling chemical processes such as bond breaking and formation, many (but not all) QM methods are able to provide insight into the potential energy surface of the chemical system without a need for prior parameterization. QM methods are, in fact, approximations, because of the intractability of the many-electron Schrödinger equation. The approximations range in sophistication and computational expense. There are comparatively inexpensive semi-empirical approaches, such as the various NDO models (see e.g. (Pople, Santry, & Segal, 1965; Pople & Segal, 1965)) and the AM1 (Dewar, Zoebisch, Healy, & Stewart, 1985) and PM3 (Stewart, 1989) families of semi-empirical Hamiltonians, all of which involve some degree of parameterization. And there are very accurate but computationally extremely intensive methods such as coupled-cluster (Bartlett & Musiał, 2007) and full configuration interaction. (Shavitt, 1998)

Within any given formal methodology, computation requires additional technical approximation. For example, in any QM scheme formulated in terms of molecular orbitals, the scientist must choose one or more basis sets (note that this is true even of grid-based methods, for which the basis is Dirac delta functions at the grid points), that is, sets of functions linear combinations of which give the molecular orbitals. While the computational difficulty of a QM calculation, and the way in which that difficulty scales with number of particles, both vary from method to method, in all cases QM calculations are far more demanding than a MM calculation on a similarly sized system. The phrase "number of particles" itself is deceptive. For MM it is the number of classical objects, hence, at most the number of atoms. In the QM case, it is at least the number of electrons, which is roughly an order of magnitude larger. Added to that is the fact that the computational costs of QM methods for electrons typically scale as some power of the number of electrons, whereas MM cost scaling is roughly $N_p lnN_p$ with N_p the particle number.

Various approaches have been developed over the years to preserve the advantages of QM calculations (e.g., proper treatment of bond breaking and formation and of electron distribution) while reducing the disadvantages of high computational cost and corresponding intractability of large chemical systems. Among them there are the fragmentation methods that depend upon some scheme for partitioning the system into distinct fragments and obtaining the total properties of the system through aggregation of the fragment properties. (Gordon, Fedorov, Pruitt, & Slipchenko, 2012) A clear example is the divide-and-conquer approach as originally formulated by Yang. (Yang, 1991) It treats a system as a set of subsystems that can be solved largely independently of one another. Thus, the density of the system of interest is divided into the sum of the densities of the subsystems, using an efficient one-electron density matrix approach. Consequently, this approach ignores many unimportant interactions and significantly reduces the computational expense.

An alternative approach is hybrid quantum-mechanics/molecular-mechanics (QM/MM) or multi-scale simulation, which was initially proposed in a 1976 paper by Warshel and Levitt.

(Warshel & Levitt, 1976) In a QM/MM simulation, the researcher chooses a small region of the system that is of particular chemical importance. That region is treated using QM, while the remainder of the system is treated via MM. The approach, while necessarily less accurate than representing the whole system quantum-mechanically, offers a good balance of physical accuracy and relatively low computational cost. (Lin & Truhlar, 2007; Senn & Thiel, 2009) Implicitly we have introduced another assumption in this discussion, namely that QM forces from the sub-system electrons determine the classical forces from within that region upon the subsystem nuclei. Thus we have invoked the Born-Oppenheimer (BO) approximation for QM/MM and QM/MM-MD.(Barnett & Landman, 1993) We do not consider beyond BO methodology here. Since we assume the BO approximation, in the discussion that follows the terms "QM package" or "QM code" means an electronic structure code.

An aside on terminology and associated notation may be helpful. QM/MM as used here denotes the separation of the system into subsystems (regions) according to the way in which the forces are generated within that region. But the very use of forces itself means that *all* of the nuclei (or all of the more coarse-grained particles) nevertheless are positioned or moved according to classical mechanics. Hence the overall approach is still MM and the notation and terminology has two meanings depending on context. Observe that the materials physics community typically uses "multi-scale" instead of "QM/MM". The early PUPIL papers use that terminology. Identifying the source of the forces introduces a bit of cumbersomeness when discussing MD. Here we use QM/MM-MD to denote MD with forces from a QM/MM decomposition. Note that for QM force driving MD without any QM/MM separation, the literature has several different names: Born-Oppenheimer MD (BOMD) (Barnett & Landman, 1993), ab initio MD (AIMD) (Marx & Hutter, 2009) and even quantum MD (QMD) (Horner, Lambert, Kress, & Collins, 2009) are common.

Over time, members of the scientific community have released many codes to carry out MM or QM calculations. Notable MM programs include AMBER, (Case, et al., 2006) CHARMM,

(Brooks, et al., 2009) NAMD, (Phillips et al., 2005) DL_POLY (Smith & Forester, 1996) and so forth. Prominent molecular QM packages include GAUSSIAN, (Frisch et al., 2009) GAMESS, (Schmidt et al., 1993) Jaguar, (Bochevarov et al., 2013) Q-CHEM, (Shao et al., 2006) NWChem, (Valiev et al., 2010) deMon2K (Köster et al., 2011) and Siesta (Soler et al., 2002) etc.

One computational approach to QM/MM is to implement both functionalities in the same package. This approach is used by AMBER (which has evolved from pure MM to contain some limited native QM functionality) and by GAUSSIAN and deMon2k, (Salahub et al., 2015) both of which started as pure QM codes but now have some MM functionality. However, this approach seems, more often than not, to be of limited utility, as those who develop and maintain the software put most of their effort into the program's "strong suit". A tendency to bias toward a development group's strength is, of course, completely understandable. The equally understandable consequence is that the implementation of the other component is restricted as to available techniques and the size and complexity of systems that can be considered.

An alternative is for an MM program and a QM program to interface directly. This approach allows the MM program to access much of the functionality offered by the QM program, and so forth. Commonly this strategy is implemented by making one code into a library for the other, yielding a monolithic package upon compilation. But doing so results in an intimate linking of multiple codes developed by distinct groups. The result is a distinct challenge for both maintenance and enhancement. Changes in one of the codes very often cause changes deep in the others to be mandatory if generation of the monolithic package is to remain supported.

A third (and very distinct) option, the philosophy used by PUPIL, is for the MM program and the QM program each to communicate with a linker program. All the codes, whether MM or QM, maintain their own architectural and developmental autonomy to the greatest degree possible. In this way, theoretically, only one interface need be maintained by each MM program and each QM program, while the development effort can be focused on the linker program. The philosophy insists

upon minimal if any modification to either a QM or MM package. And it presumes that any MM package supported by PUPIL can be utilized in combination with any QM package supported by PUPIL. Recent changes to the ChemShell code, (Metz, Kästner, Sokol, Keal, & Sherwood, 2014) are somewhat analogous in that they have included a number of interfaces to QM codes.

PUPIL began in the materials physics simulation community, so the notation and some of the motivating text in the original papers (Torras, Deumens, & Trickey, 2006; Torras et al., 2007) may be a bit unfamiliar. Thus it is fitting, before going to PUPIL itself, to set formulations and notation for MM, MD, and QM/MM and QM/MD.

2. QM/MM-MD methodology

Most current QM/MM schemes, including the one used in PUPIL, are based on the approach developed by the 2013 Nobel prize recipients Arieh Warshel, Michael Levitt, and Martin Karplus, (Field, Bash, & Karplus, 1990; Warshel & Levitt, 1976). As noted already, the simplest version involves partitioning the system into two regions ("inner" and "outer" regions) to be treated at different levels of approximation. The inner region contains only a small number of atoms that are the chemically relevant part of the system. Itis treated with a QM method for the forces from that region upon the nuclei. The outer region, the remainder of the system, is treated via MM. At this point, there are two main schemes to consider for the QM/MM energy expressions: the subtractive and the additive QM/MM schemes. The subtractive QM/MM scheme calculates the entire system at the MM level of approximation, whereas the inner system is calculated at the QM level. Subtraction of the energy of the MM calculation for the inner QM region to avoid double counting then yields the final energy expression. This kind of scheme was initially implemented by Morokuma and co-workers (Maseras & Morokuma, 1995) and later extended to include calculations using electrostatic embedding. (Chung, Hirao, Li, & Morokuma, 2012) However, in such an interpolation scheme the QM/MM interactions are handled entirely at the MM level. On the other hand, the additive QM/MM

scheme utilizes a similar system partition but requires an explicit treatment of the QM/MM coupling terms. Given this partitioning, the potential energy for the system can be written as

$$E = E^{QM} + E^{MM} + E^{QM/MM} \quad (1)$$

Here E^{QM} is the sum of electronic energies and inter-nuclear Coulomb repulsion energies for the QM part of the system, perturbed by the presence of the atoms in the MM region, E^{MM} is the potential energy for the MM part of the system, and $E^{QM/MM}$ describes the interaction energy between the two regions. Typically it contains terms for electrostatic, van der Waals and bonded interactions across region boundaries

$$E^{QM/MM} = E_{vdW}^{QM/MM} + E_{electr}^{QM/MM} + E_{bond}^{QM/MM}, \quad (2)$$

In the PUPIL implementation,(Torras, et al., 2006; Torras, et al., 2007) the van der Waals interactions between the QM and MM atoms are calculated as usual by the MM program, utilizing standard parameters for whatever force field is adopted. It has been shown that considering this interaction as purely classical does not introduce significant errors in the calculation.(Riccardi, Li, & Cui, 2004) For instance, in the AMBER-PUPIL interface (which we describe below), the van der Waals interactions are calculated using a 12–6 Lennard-Jones potential

$$E_{vdW}^{QM/MM} = \sum_{\alpha}^{QM}\sum_{i}^{MM}\left[\frac{A_{\alpha i}}{R_{\alpha i}^{12}} - \frac{B_{\alpha i}}{R_{\alpha i}^{6}}\right], \quad (3)$$

where Greek letters label atoms in the QM region and Roman letters label those in the MM region, $R_{\alpha i}$ is the distance between atoms α and i, and A and B are standard Lennard-Jones parameters for the interaction between α and i.

The calculation of the electrostatic interaction between QM and MM atoms depends on the resources and information made available by the QM program of choice, but usually can be divided in two parts: the influence of the atoms in the MM region on the atoms in the QM region and vice-versa,

$$E_{elect}^{QM/MM} = E^{QM\leftarrow(MM)} + E^{MM\leftarrow(QM)} \quad , \tag{4}$$

Whenever possible, PUPIL makes use of electrostatic embedding for the calculation of the $E^{QM\leftarrow(MM)}$ term, i.e., the atoms in the MM region are passed to the QM program as effective point charges fixed at their respective coordinates, with the classical force field values used for the charges. As a result, this term usually is already accounted for in E^{QM} and the forces acting on the QM atoms. The $E^{MM\leftarrow(QM)}$ term usually is not calculated, since the object of interest in a MD calculation is actually the force that arises from that interaction, which can be calculated directly. As some QM packages do not calculate the forces exerted on the atoms of the MM region due to the interaction with the atoms in the QM region, calculation of this contribution depends on the specific QM program used. GAUSSIAN, for example, can provide the electric field $\vec{E}$ at the locations of the point charges, which can be used to calculate the forces: (Roberts et al., 2012)

$$\vec{F}_i^{QM} = q_i \vec{E}, \tag{5}$$

where $\vec{F}_i^{QM}$ is the force acting on atom *i* of classical charge q_i in the MM region due to the interaction with the QM density. Also, recent program versions of NWChem and deMon2k are able to calculate the forces exerted on the atoms of the MM region due to the interaction with the atoms in the QM region as a normal program output. For other programs and older versions of NWChem and deMon2k, this force contribution can be obtained by projecting the electronic density of the QM system on a grid, then calculating the interaction between the classical charges and each point of the grid,

$$\vec{F}_i^{QM} = \sum_j^N \vec{r}_{ij} \frac{q_i}{\left|r_{ij}\right|^3} dq_j, \tag{6}$$

where *N* is the number of points in the grid, and $dq_j = \rho_j \, dx \, dy \, dz$.

Note well that special treatment must be used whenever there are covalent bonds that cross the QM–MM boundaries, such that cutting the molecule at the boundary would leave both the MM

and QM regions with incomplete valences. This can be the case in biomolecular simulations, for instance, wherein only a limited set of the residues, e.g. the residues in the active site, are to be included in the QM region. To deal with this issue, PUPIL uses the link atom method (Singh & Kollman, 1986). A non-physical quantum atom (the *link atom*) is introduced in the QM region along the covalent bond between the MM and QM region, at the appropriate distance from the QM atom, completing its valence. To avoid improperly high electrostatic interactions, the MM atom is not included among the point charges around the QM region. In the case of the MM region, all force field terms that include at least one MM atom are calculated, and all terms involving QM atoms exclusively are omitted. There are no MM terms including the link atom. Once the QM gradient (i.e. the force on the link atom treated as a QM atom) is available, it is redistributed to the QM and MM atoms that form the linked pair. Redistribution is via a chain-rule recipe from the energy gradient with respect to the link-atom coordinate to gradients with respect to QM and MM atom coordinates. A detailed discussion can be found in (Field, Albe, Bret, Proust-De Martin, & Thomas, 2000) and (Walker, Crowley, & Case, 2008).

3. The PUPIL framework

PUPIL is an acronym for "Program for User Package Interfacing and Linking". Its original design (Torras, Deumens, & Trickey, 2006) was motivated by a materials physics problem, namely hydrolytic weakening: in the presence of water a ceramic under tension fractures much more readily when wet than when dry. The design process involved serious examination of the QM and MM software that was deemed potentially relevant. For the QM side that meant quantum chemistry software, with an initial emphasis on semi-empirical methods calibrated to high-level coupled cluster calculations on model systems. (Mallik, Runge, Dufty, & Cheng, 2007) It was clear even at that stage, however, that the design had to accommodate more sophisticated QM software without fundamental alteration. For the MM side, the design considerations were most strongly influenced

by what seemed to be the dominant MD code for materials at the time, DL_POLY. (Todorov, Smith, Trachenko, & Dove, 2006) Little if any consideration was given to MM in the static sense but that has not turned out to be a limitation. A third major category of functionality was "domain identification" (DI). The DI concept is to provide automated identification of the chemically active region within which the QM forces are necessary. Even today, that identification almost always is done by the software user (identification "by hand") rather than by the software itself, but the design anticipated automation. The fourth major category of functionality was user support via an easy-to-use interface.

As we have summarized already, at the time of the PUPIL initial design, there were two main ways to do a QM/MM simulation which exploited the capabilities of existing codes. One was to merge them into a single code of some sort. Typically this takes considerable rewriting of the component codes, as well as writing of new data-interchange and control-interchange code, with the architectural outcome being that one component code becomes the manager of the other. In addition to the labor involved, there is another serious problem. Much of that work must be redone each time there is a new major release of any of the component codes. Almost universally, the modifications are too deeply entangled in the internals of the component codes to allow easy updates. Those barriers are part of the motivation for an approach that is common now, namely to construct the simulation via scripting: the component codes are invoked and controlled via the script and data are moved, reformatted, combined, and processed via the script. For a skillful user, the benefit is that a novel simulation can be assembled quickly, but the prerequisite is intimate familiarity with the input, output, and control structures of each component code. The disadvantages are a very high barrier to less-experienced users, replication of effort, and error proneness owing to lack of systematic protocols (an almost inescapable side effect of the flexibility provided by scripting).

PUPIL therefore is a software environment mainly designed with three aims. The first was (and is) to bring to the user a general open source tool to perform QM/MM (multi-scale) simulations

within materials physics, chemistry, and biochemistry by exploiting existing QM and MM codes, the "user packages". (Notice that this "packages" terminology implicitly acknowledges that there may be some MM capability in a QM code and conversely. PUPIL design does not care; its focus is on interoperation.) The second was (and is) to provide a means for developers to contribute new and improved capabilities easily, either by adding support for new user packages or by adding new common and interesting functionalities. The third aim was (and is) to achieve the first two as generically as possible, that is to say, with as minimal as practicable intrusion into the user packages as possible. The extent to which PUPIL has achieved those design aims is illustrated at least in part by the fact its range of usage now goes outside materials physics and chemistry to more general QM/MM-MD simulations on structures as demanding as complex enzymes.

The design philosophy of PUPIL is to provide an environment wherein all the common capabilities needed for the multi-scale simulation are collected into PUPIL itself. Thus, developers of user packages can link their applications to PUPIL to perform simulations interacting with other software in which data and simulation control are transferred from one user package to another in a straightforward manner. PUPIL itself acts as the supervisor program. It coordinates execution and communication between the user packages, each of which provides a calculation unit (CU). The supervisor is implemented as a distributed program with one Manager and several Workers, one Worker for each CU. The Manager and Workers communicate through the network using the client-server model design.

3.1 Features

All QM/MM-MD simulations performed within the PUPIL framework involve the execution of at least two user packages, an MD engine (e.g. AMBER, DL_POLY) and a QM engine (e.g., NWChem, deMon2k, etc.). We leave DI aside for the moment. These programs are executed, coordinated, and managed by the PUPIL Manager. A user therefore may choose, in mix and match fashion, among any of the possible combinations of codes currently interfaced with PUPIL. The list

of QM and MM codes currently interfaced to PUPIL is given in Table 1. Each code has a specific interface in PUPIL, written in Java (details below), to allow for communication and exchange of information to support all features in the external codes. The communication between packages is basic enough, i.e., coordinates, atom types, atomic charges, forces, and energy, that new additional features in a user package almost always are incorporated immediately by the general features of the PUPIL Manager. (The exception would be some scientific method or concept not previously supported at all.)

PUPIL functionalities and capabilities are distributed among the QM Engine, MD engine, and the PUPIL Manager. The user must be familiar with the user packages selected for the particular QM/MM-MD simulation at the level of knowledge of input and output file formats and contents. A set of input file templates from each external engine must be supplied by the user with the usual information associated with each engine. For example this would consist of system coordinates, MM atom particle types, and force field for the MD engine, and QM atom types, QM approximation ("level of theory" to use common but unhappy terminology), and convergence directives for the QM engine. However, information related to the QM/MM-MD simulation and the coupling between QM and MM calculations is supplied through the PUPIL Graphical User Interface (GUI). It yields an output file containing all the information necessary to assist the PUPIL Manager to conduct the whole simulation. Neither engine knows explicitly about the other.

As laid out in Section 2, all QM/MM-MD calculations are performed within the additive QM/MM scheme of energy partition between an "inner" (QM) and an "outer" (MM) region plus a coupling term between them. Two different QM/MM coupling schemes are allowed, mechanical and electrostatic embedding. In the former scheme the QM calculations are performed in the inner region in the absence of the outer region, with the interaction between the outer and inner regions treated at the MM level (both bonded and non-bonded interactions) of approximation. In the latter scheme, the QM Hamiltonian includes classical partial charges from the MM description as point charges which

thereby polarize the QM region. Similarly, the forces on the classical partial charges due to the interaction with the electronic density of the QM region are also included. That, in turn, induces polarization of the MM region by the QM region along an MD trajectory.

QM packages linked to PUPIL can have two different behaviors depending upon how they are invoked by the PUPIL Manager: Cyclic or Start–Stop. In Cyclic mode, a CU is started by the Manager, and all the actions involved in any individual QM/MM-MD simulation step (e.g., data request, data insertion, computation, and return of results to the PUPIL Manager) are performed at that step without restarting the QM package. Start-Stop mode so far has been used for QM CUs. In that mode, the QM CU is started by the PUPIL Manager. Upon completion for that step, the QM CU terminates and its output files are parsed to get the information back to the Manager for use in the MD CU. Thus, a new instance of the QM CU is started and executed at each force evaluation. The advantage of Start-Stop mode lies in the ease with which PUPIL can link packages without requiring any source-code modification or recompilation. It is the only route available to supporting closed-source codes. The disadvantage is loss of speed and flexibility. Some of the CUs that are more tightly coupled to PUPIL require minimal source-code modification and linking with the PUPIL libraries to be used with the PUPIL interface. Specifically, those QM CUs which operate in Cyclic mode as well as all MD user packages so far have this kind of link to the PUPIL Manager.

We have already remarked on treatment of the QM-MM region boundary by the link-atom approach to saturate the dangling bond of any QM atom left over from a broken QM-MM bond. This link atom is usually taken to be a hydrogen atom. However, the user also may use any of the quantum atoms allowed by the external QM engine to saturate the free valence of the QM atom. An example is pseudoatoms with a parameterized effective core potential (ECP) which can be adjusted to mimic the properties of the original chemical bond being cut. (Mallik, Taylor, Runge, Dufty, & Cheng, 2006)

3.1.1. High performing computing

Distinct (so far as we know) from other implementations, PUPIL treats the MD, QM, and DI codes at the same level, so that PUPIL can control their execution on the same footing. The necessary resources, i.e., processors, and the information to communicate and control the external code execution are stored and coordinated by the PUPIL Manager. In this way, PUPIL is capable of dynamically starting and stopping external MPI codes on demand, with communication among external codes conducted within the CORBA protocol by means of the client/server paradigm. (Torras, et al., 2007) See details below. The great advantage of this architecture lies in the ease with which workloads can be distributed across multiple computing resources. The recent addition to the PUPIL Manager of the capability to handle a fixed number of multiple independent active zones (QM regions) during the whole simulation (Torras, 2015) is made possible by this architecture. Another beneficial aspect is the capability to assign different computational resource to the different CUs depending on their computational time scaling (thereby managing load balancing). Indeed, running two or more separate binaries in a high-performance computing (HPC) environment can be optimized efficiently by balancing the resources assigned to each parallelized external code involved.

This approach overall is a generalization of the original "inner" and "outer" region paradigm of Warshel, Levitt, and Karplus. In principle, the DI user package would decide the number, type, and extent of active zones on the fly during the simulation. Although the current PUPIL implementation does not support that advanced feature, it is conceptually possible. It would require capabilities of a much more sophisticated DI code than presently exists. The crucial PUPIL property to be emphasized is that such a DI code again would not need to know, nor would it know, about the QM and MM user packages. PUPIL simply would provide the sophisticated DI with the data (from the QM and MM package results) needed to determine the boundaries and properties of each active region, with those results then relayed back to the appropriate user packages by PUPIL.

The most time-consuming process, hence the major bottleneck, in QM/MM-MD simulations is the force calculation by the external QM engine. Typically 80-95 % of the total time invested in

one QM/MM-MD step is consumed by the QM calculation. The next largest time cost is from building the quantum zone embedding.(Torras, et al., 2006; Torras, et al., 2007) Thus, major effort must be given to the parallelization of the QM code calculation. PUPIL is able to deal with parallelized code for a user package, i.e., the QM code. To facilitate the execution within an MPI environment of any external code, the PUPIL Manager takes advantage of its capability to dynamically start and stop processes to assign specific resources to each CU. Thus, prior to starting any parallel Worker (QM or MD), the parallel environment must be initiated in accordance with local hardware and software cluster characteristics and policies, e.g., OpenMPI, MPICH2, etc. In fact, an automatic startup shell script is generated from the PUPIL core following a user-provided shell script template which incorporates those local cluster characteristics and policies. An example would be the MPI environment commands to get the Worker running in the local hardware environment and the execution syntax for the corresponding CU. Though this is a platform-dependent solution, experience suggests that just a few templates can cope with most MPI environments.

The PUPIL Core is implemented in Java. To provide good performance in building the quantum zone embedding, specific parts of the PUPIL core (for example, application of embedding rules) are executed in parallel using Java threads. Also, the most computationally demanding coupling terms are calculated through a parallel execution using the Java Native Interface (JNI) combined with native C code.

3.2 User Interface

All simulations are done in three steps. Initially, the PUPIL Graphical User Interface (GUI) supports preparation of input data. Second, the PUPIL Manager uses information prepared by the GUI to start the simulation and all the externally linked CUs. Finally, output files from the Manager and each of the CUs are analyzed by the user.

The main functionality of the GUI is collection of general simulation information along with the required input files for the CUs to be used. Thus the user must already be familiar with the external interfaces offered and input data formats required by each CU. Recall the mention above about the user needing to supply input file templates. The GUI helps the user to generate all necessary information to conduct the QM/MM-MD simulation. Thus, the input file templates are preprocessed and parsed to extract information. This information is used during the simulation to coordinate data exchange between CUs. At the end, the GUI saves all collected information in a structured data file (XML) that is then supplied to the Supervisor as input file at simulation time. (As a remark about the limitations of design, in spite of our efforts, the GUI design was subconsciously biased to materials systems, which typically have a small number of different atoms compared to the number found in biomolecules. An unintended consequence is a bit of cumbersomeness.)

3.2.1 QM program and method selection

Obviously the user must specify a CU for each of the three main actors involved in any QM/MM-MD simulation, viz., the force generator (QM engine), molecular dynamics (MD engine), and domain identification (DI engine) method. The GUI helps in the selection. Thus, several QM engines are available from which to choose (see Table 1) to provide the energetics and forces in the QM region. The GUI also enables specification of the common parameters for each QM engine involved in the simulation, such as the use of periodic boundary conditions (PBC), selection of electrostatic embedding, the use of link-atom pairs when there are QM-MM chemical bonds crossing the QM region, and the use of long-range electrostatics in the QM/MM coupling term.

3.2.2 QM region selection. Rules

In addition to selection of the QM engine itself, an obviously important step is selection of the QM region and its environment. The DI Worker is a module for control of the QM/MM partitioning, that is, setting of the inner (QM) region and outer (MM) regions. Currently, two kinds

of Domain Identification are supported: Manual Region Specification and specification through an external program. Manual Region Specification is determination of the QM region by user choice ("by hand"), along with the link-pairs connecting the quantum and classical regions, and the embedding particles used as point charges. The selection is made by a user-friendly interface that enables specification of rules to define the different layers that will comprise the QM region.

In the design of the quantum zone embedding, it is allowable to choose not only the QM region but also some additional embedding regions. One may distinguish among three main regions, namely the quantum, classical, and static-charge regions. All system particles involved in the simulation must be assigned to one of those three regions. There are four different basic categories that allow the user to define all atoms/residues belonging to one of the three regions easily: direct atom/residue type assignment, fixed link pairs during the whole simulation, variable link pairs during simulation (distance-based assignment), and neighboring-residue type assignment. Direct assignment is the normal method of choice for the typical QM/MM-MD simulation in which neither the QM region nor its embedding region changes shape or chemical composition during the whole simulation. In contrast, the neighboring rules are designed mainly for those simulations with a variable quantum region.

We have already mentioned a more sophisticated capability designed into PUPIL, domain identification through an external program. This option allows specification of a complex QM region via a user package as another CU which interacts with the PUPIL simulation Manager analogously with the external MD and external QM programs. This functionality is useful when specification of the quantum region by means of the usual manual region specification rules is complicated, e.g., assignment of multiple QM regions, and the design of variable quantum regions on the fly depending upon some physical or chemical property of the system. Thus the external DI should interact with the Simulation Manager by exchanging information relevant to the QM region, whereas the embedding region can be managed through the usual neighboring rules.

3.3 Technical details

We noted above that the PUPIL Manager, GUI, and substantial portions of the Workers are implemented in Java. The main advantages of using Java are fast implementation, easy maintenance, software reuse, and multi-platform support. Though there is some platform-dependent code, it is mainly localized in the wrapper interface between the PUPIL system and the CUs which are tightly coupled with PUPIL, e.g. MD engines and some QM engines. Most of the CUs are written in FORTRAN, though some are in other languages. Therefore, a wrapper written in C was built through the JNI (Java Native Interface) as a natural bridge between both languages.(Liang, 1999) All the wrappers have been merged in a single C library. However, additional code modifications to the CUs which are tightly coupled to PUPIL become a simple packing and unpacking of data to be exchanged with the PUPIL system on the QM engines, plus some additional routines to hold the QM/MM coupling in MD engines.

Figure 1 shows the general behavior and exchange of information of the whole PUPIL framework. (Torras, et al., 2007) The Manager is the main application to execute the user directives previously assigned using the GUI. The majority of its code has generic behavior, resulting in significant software reuse for support of all the CU Workers. Generally speaking, QM/MM-MD simulations are performed as a distributed execution that runs with a main application and several Java Virtual Machines (JVM), one for each CU Worker. The Manager prepares and starts the simulation environment, logs all the distributed processes, performs error control, and concludes the simulation. Each CU (MD, DI, and QM engines) has its own Worker.

The data flow at a given MD step of the QM/MM-MD simulation is this: the MD Worker receives atom types, coordinates, and velocities from the MD engine. Then, the MD Worker decides about the procedure to identify the quantum region (whether to use an external DI or not), and

freezes the MD engine. Prior to submitting the information to the QM Worker, the MD Worker adjusts the quantum region by adding the required embedding particles (electrostatic embedding or link-pair atoms). Upon receiving the set of forces associated with the QM region from the QM Worker, the MD Worker puts those forces into the MD engine and releases it to proceed with the subsequent MD step.

[Insert Figure 1 here]

Figure 1. Distributed Supervisor processes at each Worker, and their associated CORBA clients and servers.

All the communications among the distributed processes are done through the CORBA protocol (Common Object Request Broker Architecture). Each Worker (process) has at least one CORBA server associated with it, along with several CORBA clients depending upon the other servers with which it is communicating (see Figure 1). Thus, all the client-server communications are performed from Java code and the communication between them and their associated CU, i.e., MD, QM, and DI engines, is through the wrapper interface described above.

4. Biomolecular applications

A particular advantage of the interfacing and linking approach of PUPIL is that, during each computation of the system's Hessian matrix (typically in the MM Worker), the forces on the particles are modified in place according to the results of a QM calculation. This approach means that, in general, any scientific method implemented in the MM calculation unit that involves computation of forces on a per-particle basis can support a QM/MM treatment with PUPIL, even if any native QM/MM implementation in that particular user package does not support that particular method. (Of course, it helps if the user package is designed in such a way that the forces are computed in one

single subroutine, which then is easy to modify to support outgoing connections to the PUPIL Manager.)

[Insert Figure 2 here]

Figure 2. Minimisation of the Heme group (QM region) within the Myoglobin protein (MM region). Highest Occupied Molecular Orbital (HOMO) is also shown.

One such scientific method is energy minimisation (alternatively known as geometry optimization, Figure 2). While this is a fundamental operation in MM, and widely considered necessary before commencing a MD simulation, it often is done only part-way, so as to eliminate egregious close contacts between particles and other grossly unfavorable structural features. A common approach in the MD Worker, therefore, is to use a conjugate-gradients algorithm, which comes with the distinct disadvantage (in a QM context) that it requires many force evaluations. An alternative, if offered by the MD Worker, is to optimize the structure using a quasi-Newtonian method such as the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) method. (D. Liu & Nocedal, 1989) Although each step may be more computationally intensive (once forces are evaluated), L-BFGS requires fewer steps and thus fewer force evaluations overall. This approach to energy minimization is thus very useful in the context of a QM/MM calculation.

Furthermore, PUPIL has the potential to allow for access to a multitude of advanced MD simulation techniques, again depending on the battery of methods implemented in the user package. Here, we describe a few examples. The technique of replica-exchange MD (REMD) (Sugita & Okamoto, 1999) permits a more comprehensive sampling of phase space in shorter timescales than would be possible via conventional MD. This sampling is achieved by running different simulations of the same physical system at different temperatures, and at chosen times offering these different simulations (“replicas”) an opportunity to exchange positions and momenta, the latter being scaled

according to the difference in temperature. Whether an exchange actually occurs is determined by a probability function such as the Metropolis criterion. Zhang *et al.* have shown that at high temperatures ($T > 300$ K), REMD offers an excellent alternative to long-timescale MD simulations.(Zhang, Wu, & Duan, 2005)

An alternative, if one wishes to bias the system towards a particular, known configuration, is to use targeted MD. In it, the potential energy is biased by the addition of a constraint force on each of the particles such that the particle is pulled towards the position it would hold in a final configuration that is specified by the operator. At each step, the constraint force is computed from the vector difference between the final configuration and the current configuration, along with an appropriately chosen Lagrange parameter. (Schlitter, Engels, & Krüger, 1994; Schlitter, Engels, Krüger, Jacoby, & Wollmer, 1993) Targeted MD is a particularly useful technique for simulating gross changes in the structure of a biomolecule that might not be expected to occur on the timescale of a conventional, unbiased MD simulation. To the extent that the constraint forces are expressed mathematically as modest corrections at each step to an otherwise ordinary force matrix, targeted MD is eminently compatible with the PUPIL approach to force evaluation.

A particularly important application of QM/MM techniques is modelling chemical reactions involving bond breaking and formation. Scientists simulating these reactions commonly use the potential of mean force (PMF) technique, which is implemented in various packages. PUPIL allows the investigator to construct PMF profiles of reaction coordinates (that is, the bonds to be broken or formed) using QM methods or basis sets that are not native to the MD program. One particular method for computing a PMF is umbrella sampling, by which a bond length is constrained to be near a specified value (or, more commonly, a series of specified values) by a harmonic potential. The statistical distribution of actual bond lengths arising from the sum of the harmonic potential and the underlying potential energy equation for the system (as specified by the QM and MM approaches used) can be analyzed using the weighted histogram analysis method (WHAM). (Kumar, Rosenberg,

Bouzida, Swendsen, & Kollman, 1992) Thus, the scientist can derive an equilibrium constant and free energy profile for the reaction. An alternative approach to umbrella sampling and WHAM is to use steered molecular dynamics (SMD) and Jarzynski's equality, as described by Park and Schulten. (Park & Schulten, 2004)

Although initially developed with materials simulations in mind, PUPIL is general enough to be applied in any field of molecular simulation. It requires only a comparatively short development of an interface between PUPIL and the desired programs. On account of the software architecture and modular construction of PUPIL, that development typically is limited to some wrappers and file parsers. Most of the QM/MM interface is already implemented and thus can be reused. (Torras, et al., 2006; Torras, et al., 2007) For example, an interface with the biomolecular simulations package AMBER9 (Case, et al., 2006) was developed in 2008, (Torras, Seabra, Deumens, Trickey, & Roitberg, 2008) together with an interface to the GAUSSIAN QM package.(Frisch, et al., 2009) This interface was used to study the decomposition of Angeli's salt in explicit solvent. (Torras, Seabra, & Roitberg, 2009) Angeli's salt, $Na_2N_2O_3$, has unique cardiovascular effects, associated with its ability to yield HNO upon dissociation under physiological conditions. Its dissociation had been studied earlier, using a polarized continuum model to represent the solvent. (Dutton, Fukuto, & Houk, 2004) The use of the AMBER-PUPIL-GAUSSIAN interface allowed the study of the reaction by use of the Multiple Steered MD (MSMD) capabilities built in AMBER with the Jarzynski relationship to calculate the free energy of the process, and by use of GAUSSIAN to calculate the energy of the QM region at UB3LYP and UMP2 levels of density functional theory (DFT) approximation with 6-311+G(d) basis sets. The explicit inclusion of the solvent molecules allowed a more precise determination of the free energy barrier of decomposition, thereby giving evidence of the importance of explicit consideration of the solvent molecules.

Later, a PUPIL interface to the quantum chemistry program NWChem was developed and used to analyze the conformational preferences of proline (Pro) analogues containing a fused

benzene ring, which reduces the molecule flexibility. (Warren et al., 2010) Proline is the only proteinogenic amino acid that is naturally conformationally constrained. This constraint is highly significant in protein structure, and has stimulated the search for Pro analogues with tailored properties. The incorporation of functional groups from different amino acids is particularly interesting. In that study, the authors concentrated on indoline-2-carboxylic acid (Inc) and its methylated derivative, resulting of a fusion of a benzene ring, present in phenylalanine (Phe), to the pyrrolidine bond linking the γ and δ carbons in Pro, and consequently can be considered either a Pro or a Phe analogue, a combination with important applications in drug design. In an attempt to understand the effects of the additional benzene ring in the conformational preferences of Pro, the authors used DFT calculations in vacuum, and estimated the effects of the solvent environment by use of an implicit (SCRF) solvent and explicit solvent by hybrid QM/MM-MD calculations using the AMBER-PUPIL-NWChem interface. The authors noted that the DFT calculations in vacuum overestimated the stability of the structures with *cis* distribution around the ω angle even though, experimentally, only the *trans* arrangement has been detected for the derivative. Inclusion of solvent effects by means of PCM/SCRF calculations did decrease the free energy difference between *cis* and *trans* structures, but the *cis* is still overstabilized. Only after explicit solvent molecules were considered by means of the QM/MM-MD interface was the *trans* disposition predicted to be considerably lower in energy than the *cis*. The authors noted, however, that now the *trans* arrangement was likely overstabilized.

More recently, the same interface has been used to study the characteristics of bioactive platforms based on biocomposites of poly(3,4-ethylenedioxythiophene) (PEDOT) and collagen (CLG), named P(EDOT:CLG), where the presence of the collagen protein affects both the morphology and electrochemical activity of PEDOT. (Soto-Delgado, Torras, del Valle, Estrany, & Aleman, 2015) The specific interactions between PEDOT and CLG were studied quantum mechanically with MP2/6-31+G(d,p) methodology both in vacuum and in solution The solvent

presence was represented implicitly using PCM/SCRF and explicitly via the AMBER-PUPIL-NWChem interface at the UB3LYP/6-31+G(d,p) level of approximation, using chloroform or water as solvent. In the calculations, the PEDOT was modeled by the monomer, EDOT, while the CLG was represented by proline or L-hydroxyproline, each terminated by an acetyl and N-methylamide to yield Ac-L-Pro-NMe and Ac-L-Hyp-NMe. The structures derived from the QM/MM-MD calculations were in good agreement with the ones obtained with the implicit solvent models. The same specific interactions for EDOT/Ac-L-Pro-NMe complexes in chloroform and water solutions were found using implicit solvent with PCM/SCRF or explicit solvent with QM/MM-MD. On the other hand, the QM/MM-MD method reveals three different types of specific interactions between the components in EDOT/Ac-L-Hyp-NMe, which turned out to be the combination of the two modes predicted by the implicit solvent model, which were found to be practically isoenergetic.

5. Recent developments

One of the major challenges of *in silico* simulations on complex biological systems is to treat several chemically active zones concurrently because their distinct evolution is linked critically to the global system behavior. Very recently the capability for handling such multiple, disjoint QM zones in QM/MM-MD simulations has been developed within the PUPIL framework. (Torras, 2015) This new capability will allow simulational treatment of complex proteins such as those that contain multiple metallic centers, e.g., the ferritin cage (see Figure 3), ubiquinone oxidoreductase and Laccase, among others. In the first case, the ferritin cage holds several metallic ions within its structure which have been shown to be important in protein-protein interactions via formation of metal-induced self-assembly cages.(X. Liu & Theil, 2005) The complex in NADH: ubiquinone oxidoreductase plays a major role in the respiratory electron transport chain from the NADH to ubiquinone across the membrane, which is necessary for ATP synthesis.(Hayashi & Stuchebrukhov, 2010) A dynamical treatment of independent active zones to deal with distinct electron tunneling

pathways between neighboring Fe/S clusters is indicated. And in the study of metalloenzymes such as Laccase, which has several active metal sites, the new methodology should be especially useful to characterize synergies among those sites.(Piontek, Antorini, & Choinowski, 2002).

[Insert Figure 3 here]

Figure 3. Cu-Ferritin cage (a) with a selected building block monomer (magenta). (b) Detailed location of the active zones in a monomer of the Cu-Ferritin cage.

5.1. Working with Multiple active zones

To describe the multiple active zone scheme for QM/MM-MD calculations (hereafter *maz*-QM/MM-MD approach) we return briefly to the general QM/MM approach. The entire system (S) is partitioned into an inner region (I) that is treated by QM and the outer region (O) described by a force field. The energy partition of the two main regions is modeled via the additive QM/MM scheme,

$$E(S) = E_{QM}(I) + E_{MM}(O) + E_{QM-MM}(I,O) \qquad (7)$$

In the *maz*-QM/MM MD extension, the QM region is defined as the sum of several disjoint QM sub-regions (or active zones, AZs). The energy and its gradients (forces) follow from the general QM/MM approach. Thus any simulation particle (nucleus or more coarse-grained) within an AZ is subject to QM forces from the electrons in that sub-region. But the interactions with the other AZs are treated the same as with the MM region, namely, as forces from point charges in those remote sub-regions. At present, those other AZ point charges are calculated as Mulliken charges, but that is a choice, not an essential design property. This procedure is similar to the one previously proposed by Kiyota et al. (Kiyota, Hasegawa, Fujimoto, Swerts, & Nakatsuji, 2009) but is more general. The energy partition of the QM region then is formulated as follows,

$$E_{QM}(I) = \sum_{A} E_{QM}(I_A) + \frac{1}{2}\sum_{A}\sum_{B\neq A} E_{QM-QM}(I_A, I_B) \qquad (8)$$

The coupling term between two disjoint AZs (the only type allowed; see below) has contributions only from the van der Waals and electrostatic interactions between the QM atoms of those two sub-regions.

$$E_{QM-QM}(I_A, I_B) = E^{vdw}(I_A, I_B) + E^{el}(I_A, I_B) \qquad (9)$$

All AZs are treated as in the ordinary QM/MM method except for incorporating the electrostatic interactions between different AZs by means of the electrostatic-embedding scheme. Different QM/MM calculations therefore are performed concurrently, one for each AZ. As a result the conventional MM region polarizes each AZ in addition to the polarization from the different sets of point charges, each of which sets represents one of the other AZs. Thus, the electrostatic interaction between different sub-regions, $E^{el}(I_A,I_B)$, is approximated by the interaction between the electron density of one AZ with a set of charges of the other (Q_B) to simulate the charge polarization of the remote quantum sub-region instead of using its electronic density and associated multipoles:

$$E^{el}(I_A:\rho_A, I_B:\rho_B) \cong \frac{1}{2}\left(E^{el}(I_A:\rho_A, I_B:Q_B) + E^{el}(I_A:Q_A, I_B:\rho_B)\right) . \qquad (10)$$

Observe that this expression is symmetrized between sub-regions so as not to introduce a violation of Newton's Third Principle.

The potential energy of the whole system can be written as the energy of N independent QM sub-regions plus their corresponding QM/QM and QM/MM coupling terms

$$\begin{aligned} E(S) = E_{MM}(O) &+ \sum_{A}^{N}\left[E_{QM}(I_A) + E_{QM-MM}(I_A, O)\right] \\ &+ \sum_{A}^{N}\sum_{B>A}^{N}\frac{1}{2}\left[E_{QM-QM}(I_A:\rho_A, I_B:Q_B) + E_{QM-QM}(I_B:\rho_B, I_A:Q_A)\right] \end{aligned} \qquad (11)$$

The whole approach is under the hypothesis that all the AZs are sufficiently separated that their charge distributions are essentially non-overlapping. Otherwise, this approach is not valid. Thus only disjoint active zones are allowed. However, *maz*-QM/MM MD opens the opportunity

either to merge different AZs as they approach or to split an AZ if some part of it drifts away the rest. These opportunities are not yet implemented however.

This new methodology, implemented within PUPIL, has been demonstrated with treatment of small molecules in solution and of all five QM regions of the Cu-Ferritin monomer in a unique *maz*-QM/MM MD simulation, successfully (Figure 3). (Torras, 2015) Thus, it opens the possibility to perform further studies analyzing the stability of the Cu-4His-ΔC* cage, which holds about 120 AZs with about 50 of them involved on the self-assembly of protein. Indeed, modelling of large biomolecular systems that present an interrelationship between different active sites becomes much more readily accessible than heretofore.

5.2 Treatment of long-range electrostatics interactions

Long-range electrostatic interactions in conjunction with Periodic Boundary Conditions (PBC) are extensively used for prediction of condensed system properties. Treatment for long-range electrostatic interactions via PBC based on the QM/MM-Ewald summation methodology was added to the PUPIL framework recently. (Torras, 2015) This addition allows the user to choose between a simple electrostatic embedding using all the point charges of the MM particles within the simulation box (real space) and incorporation into real-space interactions those electrostatic interactions with infinitely many images of the simulation box (reciprocal space). Such long-range electrostatics via QM/MM-Ewald summation was described initially by Nam et al. (Nam, Gao, & York, 2004) The technique involves addition of a periodic correction term to both QM and QM/MM interactions for the usual real-space electrostatic interaction between QM and MM partitioning.

6. Conclusions

The QM/MM-MD methodology has proven to be a powerful approach to handle large-scale simulations in biology from the dynamics point of view. Its great potential complements classical methods used so far to obtain either a static image by means of a high-level calculation or a structural

evolution of biological systems using a low-level calculation. To this context, the PUPIL framework adds a general, flexible, modular, and readily scalable environment for performing QM/MM-MD simulations. PUPIL users can choose and match their preferred QM and MD external packages through a well-established interface. They can add packages systematically and with very substantial software reuse. Besides the basic QM/MM coupling terms among the QM and MM regions, all accessible capabilities in any PUPIL-based QM/MM-MD simulation are limited only by those available in the external packages used. The internal structure of PUPIL is designed to facilitate the management of computing resources, allowing different external packages to be executed concurrently in a parallel environment. Several applications using PUPIL to apply this methodology have been described. Generally, PUPIL applications have ranged from the solid-state to complex biomolecular systems. The continuous evolution and refinement of the QM/MM-MD model allows a more accurately system environment treatment, thus obtaining better observables and opening the possibility to study biochemical reactions from a dynamic point of view. Using the new *maz*-QM/MM MD approach, the users of PUPIL have access to large and complex biological systems to explore synergies among different active sites.

For further information about the open source PUPIL project, code download and new developments, visit the website http://pupil.sourceforge.net.

Acknowledgments

This work has been supported by MINECO and FEDER funds (MAT2012-34498), and by the DIUE of the Generalitat de Catalunya (Research group 2009 SGR 925). SBT was supported under U.S. Dept. of Energy grant DE-SC0002139.

Bibliography

Alder, B. J., & Wainwright, T. E. (1959). Studies in Molecular Dynamics. I. General Method. *The Journal of Chemical Physics, 31*(2), 459-466. doi: 10.1063/1.1730376

Barnett, R. N., & Landman, U. (1993). Born-Oppenheimer molecular-dynamics simulations of finite systems: Structure and dynamics of (H2O)2. *Physical Review B, 48*(4), 2081-2097.

Bartlett, R. J., & Musiał, M. (2007). Coupled-cluster theory in quantum chemistry. *Reviews of Modern Physics, 79*(1), 291-352.

Bochevarov, A. D., Harder, E., Hughes, T. F., Greenwood, J. R., Braden, D. A., Philipp, D. M., . . . Friesner, R. A. (2013). Jaguar: A high-performance quantum chemistry software program with strengths in life and materials sciences. *International Journal of Quantum Chemistry, 113*(18), 2110-2142. doi: 10.1002/qua.24481

Brooks, B. R., Brooks, C. L., Mackerell, A. D., Nilsson, L., Petrella, R. J., Roux, B., . . . Karplus, M. (2009). CHARMM: The biomolecular simulation program. *Journal of Computational Chemistry, 30*(10), 1545-1614. doi: 10.1002/jcc.21287

Case, D. A., Darden, T. A., III, T. E. C., Simmerling, C. L., Wang, J., Duke, R. E., . . . Kollman, P. A. (2006). AMBER 9. San Francisco: University of California.

Chung, L. W., Hirao, H., Li, X., & Morokuma, K. (2012). The ONIOM method: its foundation and applications to metalloenzymes and photobiology. *Wiley Interdisciplinary Reviews: Computational Molecular Science, 2*(2), 327-350. doi: 10.1002/wcms.85

Darden, T., York, D., & Pedersen, L. (1993). Particle mesh Ewald: An N·log(N) method for Ewald sums in large systems. *The Journal of Chemical Physics, 98*(12), 10089-10092. doi: 10.1063/1.464397

Dewar, M. J. S., Zoebisch, E. G., Healy, E. F., & Stewart, J. J. P. (1985). Development and use of quantum mechanical molecular models. 76. AM1: a new general purpose quantum mechanical molecular model. *Journal of the American Chemical Society, 107*(13), 3902-3909. doi: 10.1021/ja00299a024

Dutton, A. S., Fukuto, J. M., & Houk, K. N. (2004). Mechanisms of HNO and NO Production from Angeli's Salt: Density Functional and CBS-QB3 Theory Predictions. *Journal of the American Chemical Society, 126*(12), 3795-3800. doi: 10.1021/ja0391614

Field, M. J., Albe, M., Bret, C., Proust-De Martin, F., & Thomas, A. (2000). The dynamo library for molecular simulations using hybrid quantum mechanical and molecular mechanical potentials. *Journal of Computational Chemistry, 21*(12), 1088-1100. doi: 10.1002/1096-987x(200009)21:12<1088::aid-jcc5>3.0.co;2-8

Field, M. J., Bash, P. A., & Karplus, M. (1990). A combined quantum mechanical and molecular mechanical potential for molecular dynamics simulations. *Journal of Computational Chemistry, 11*(6), 700-733. doi: 10.1002/jcc.540110605

Frisch, M. J., Trucks, G. W., Schlegel, H. B., Scuseria, G. E., Robb, M. A., Cheeseman, J. R., . . . Fox, D. J. (2009). Gaussian 09, Revision D.01. Wallingford CT

Gaussian, Inc.

Gordon, M. S., Fedorov, D. G., Pruitt, S. R., & Slipchenko, L. V. (2012). Fragmentation Methods: A Route to Accurate Calculations on Large Systems. *Chemical Reviews, 112*(1), 632-672. doi: 10.1021/cr200093j

Hayashi, T., & Stuchebrukhov, A. A. (2010). Electron tunneling in respiratory complex I. *Proceedings of the National Academy of Sciences, 107*(45), 19157-19162. doi: 10.1073/pnas.1009181107

Horner, D. A., Lambert, F., Kress, J. D., & Collins, L. A. (2009). Transport properties of lithium hydride from quantum molecular dynamics and orbital-free molecular dynamics. *Physical Review B, 80*(2), 024305.

Kiyota, Y., Hasegawa, J.-Y., Fujimoto, K., Swerts, B., & Nakatsuji, H. (2009). A multicore QM/MM approach for the geometry optimization of chromophore aggregate in protein. *Journal of Computational Chemistry, 30*(8), 1351-1359. doi: 10.1002/jcc.21156

Köster, A. M., Geudtner, G., Calaminici, P., Casida, M. E., Dominguez, V. D., Flores-Moreno, R., . . . Salahub, D. R. (2011). deMon2k, version 3. Mexico City: Cinvestav.

Kumar, S., Rosenberg, J. M., Bouzida, D., Swendsen, R. H., & Kollman, P. A. (1992). THE weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. *Journal of Computational Chemistry, 13*(8), 1011-1021. doi: 10.1002/jcc.540130812

Liang, S. (1999). *Java Native Interface: Programmer's Guide and Reference* (1st ed.): Addison-Wesley Longman Publishing Co., Inc.

Lin, H., & Truhlar, D. (2007). QM/MM: what have we learned, where are we, and where do we go from here? *Theoretical Chemistry Accounts, 117*(2), 185-199. doi: 10.1007/s00214-006-0143-z

Liu, D., & Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming, 45*(1-3), 503-528. doi: 10.1007/bf01589116

Liu, X., & Theil, E. C. (2005). Ferritins: Dynamic Management of Biological Iron and Oxygen Chemistry. *Accounts of Chemical Research, 38*(3), 167-175. doi: 10.1021/ar0302336

Mallik, A., Runge, K., Dufty, J. W., & Cheng, H.-P. (2007). Multiscale modeling of materials based on force and charge density fidelity. *The Journal of Chemical Physics, 127*(22), 224707. doi: 10.1063/1.2802545

Mallik, A., Taylor, D., Runge, K., Dufty, J., & Cheng, H. P. (2006). Procedure for building a consistent embedding at the QM–CM interface. *Journal of Computer-Aided Materials Design, 13*(1-3), 45-60. doi: 10.1007/s10820-006-9014-0

Marx, D., & Hutter, J. (2009). *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods*. Cambridge: Cambridge University Press.

Maseras, F., & Morokuma, K. (1995). IMOMM: A new integrated ab initio + molecular mechanics geometry optimization scheme of equilibrium structures and transition states. *Journal of Computational Chemistry, 16*(9), 1170-1179. doi: 10.1002/jcc.540160911

Meinel, C. (1992). August Wilhelm Hofmann—"Reigning Chemist-in-Chief". *Angewandte Chemie International Edition in English, 31*(10), 1265-1282. doi: 10.1002/anie.199212653

Metz, S., Kästner, J., Sokol, A. A., Keal, T. W., & Sherwood, P. (2014). ChemShell—a modular software package for QM/MM simulations. *Wiley Interdisciplinary Reviews: Computational Molecular Science, 4*(2), 101-110. doi: 10.1002/wcms.1163

Nam, K., Gao, J., & York, D. M. (2004). An Efficient Linear-Scaling Ewald Method for Long-Range Electrostatic Interactions in Combined QM/MM Calculations. *Journal of Chemical Theory and Computation, 1*(1), 2-13. doi: 10.1021/ct049941i

Park, S., & Schulten, K. (2004). Calculating potentials of mean force from steered molecular dynamics simulations. *The Journal of Chemical Physics, 120*(13), 5946-5961. doi: 10.1063/1.1651473

Phillips, J. C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., . . . Schulten, K. (2005). Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry, 26*(16), 1781-1802. doi: 10.1002/jcc.20289

Piontek, K., Antorini, M., & Choinowski, T. (2002). Crystal Structure of a Laccase from the FungusTrametes versicolor at 1.90-Å Resolution Containing a Full Complement of Coppers. *Journal of Biological Chemistry, 277*(40), 37663-37669.

Pople, J. A., Santry, D. P., & Segal, G. A. (1965). Approximate Self-Consistent Molecular Orbital Theory. I. Invariant Procedures. *The Journal of Chemical Physics, 43*(10), S129-S135. doi: 10.1063/1.1701475

Pople, J. A., & Segal, G. A. (1965). Approximate Self-Consistent Molecular Orbital Theory. II. Calculations with Complete Neglect of Differential Overlap. *The Journal of Chemical Physics, 43*(10), S136-S151. doi: 10.1063/1.1701476

Rahman, A. (1964). Correlations in the Motion of Atoms in Liquid Argon. *Physical Review, 136*, A405. doi: 10.1103/PhysRev.136.A405

Riccardi, D., Li, G., & Cui, Q. (2004). Importance of van der Waals Interactions in QM/MM Simulations. *The Journal of Physical Chemistry B, 108*(20), 6467-6478. doi: 10.1021/jp037992q

Roberts, B. P., Seabra, G. M., Roitberg, A. E., Merz, K. M., Deumens, E., Torras, J., & Trickey, S. B. (2012). Comment on "a minimal implementation of the AMBER–GAUSSIAN interface for Ab Initio QM/MM-MD simulation". *Journal of Computational Chemistry, 33*(19), 1643-1644. doi: 10.1002/jcc.23003

Salahub, D., Noskov, S., Lev, B., Zhang, R., Ngo, V., Goursot, A., . . . de la Lande, A. (2015). QM/MM Calculations with deMon2k. *Molecules, 20*(3), 4780-4812.

Schlitter, J., Engels, M., & Krüger, P. (1994). Targeted molecular dynamics: A new approach for searching pathways of conformational transitions. *Journal of Molecular Graphics, 12*(2), 84-89. doi: 10.1016/0263-7855(94)80072-3

Schlitter, J., Engels, M., Krüger, P., Jacoby, E., & Wollmer, A. (1993). Targeted Molecular Dynamics Simulation of Conformational Change-Application to the T ↔ R Transition in Insulin. *Molecular Simulation, 10*(2-6), 291-308. doi: 10.1080/08927029308022170

Schmidt, M. W., Baldridge, K. K., Boatz, J. A., Elbert, S. T., Gordon, M. S., Jensen, J. H., . . . Montgomery, J. A. (1993). General atomic and molecular electronic structure system. *Journal of Computational Chemistry, 14*(11), 1347-1363. doi: 10.1002/jcc.540141112

Senn, H. M., & Thiel, W. (2009). QM/MM Methods for Biomolecular Systems. *Angewandte Chemie International Edition, 48*(7), 1198-1229. doi: 10.1002/anie.200802019

Shao, Y., Molnar, L. F., Jung, Y., Kussmann, J., Ochsenfeld, C., Brown, S. T., . . . Head-Gordon, M. (2006). Advances in methods and algorithms in a modern quantum chemistry program package. [10.1039/B517914A]. *Physical Chemistry Chemical Physics, 8*(27), 3172-3191. doi: 10.1039/b517914a

Shavitt, I. (1998). The history and evolution of configuration interaction. *Molecular Physics, 94*(1), 3-17. doi: 10.1080/002689798168303

Singh, U. C., & Kollman, P. A. (1986). A combined ab initio quantum mechanical and molecular mechanical method for carrying out simulations on complex molecular systems: Applications to the CH3Cl + Cl− exchange reaction and gas phase protonation of polyethers. *Journal of Computational Chemistry, 7*(6), 718-730. doi: 10.1002/jcc.540070604

Smith, W., & Forester, T. R. (1996). DL_POLY_2.0: A general-purpose parallel molecular dynamics simulation package. *Journal of Molecular Graphics, 14*(3), 136-141. doi: 10.1016/S0263-7855(96)00043-4

Soler, J. M., Artacho, E., Gale, J. D., García, A., Junquera, J., Ordejón, P., & Sánchez-Portal, D. (2002). The SIESTA method for ab initio order- N materials simulation. *Journal of Physics: Condensed Matter, 14*(11), 2745.

Soto-Delgado, J., Torras, J., del Valle, L. J., Estrany, F., & Aleman, C. (2015). Examining the compatibility of collagen and a polythiophene derivative for the preparation of bioactive platforms. *RSC Advances, 5*(12), 9189-9203. doi: 10.1039/c4ra13812k

Stewart, J. J. P. (1989). Optimization of parameters for semiempirical methods I. Method. *Journal of Computational Chemistry, 10*(2), 209-220. doi: 10.1002/jcc.540100208

Sugita, Y., & Okamoto, Y. (1999). Replica-exchange molecular dynamics method for protein folding. *Chemical Physics Letters, 314*(1–2), 141-151. doi: 10.1016/S0009-2614(99)01123-9

Todorov, I. T., Smith, W., Trachenko, K., & Dove, M. T. (2006). DL_POLY_3: new dimensions in molecular dynamics simulations via massive parallelism. [10.1039/B517931A]. *Journal of Materials Chemistry, 16*(20), 1911-1918. doi: 10.1039/b517931a

Torras, J. (2015). Multiple active zones in hybrid QM/MM molecular dynamics simulations for large biomolecular systems. *Physical Chemistry Chemical Physics, 17*(15), 9959-9972. doi: 10.1039/c5cp00905g

Torras, J., Deumens, E., & Trickey, S. B. (2006). Software Integration in Multi-scale Simulations: the PUPIL System. *Journal of Computer-Aided Materials Design, 13*(1-3), 201-212.

Torras, J., He, Y., Cao, C., Muralidharan, K., Deumens, E., Cheng, H.-P., & Trickey, S. B. (2007). PUPIL: A systematic approach to software integration in multi-scale simulations. *Computer Physics Communications, 177*(3), 265-279. doi: 10.1016/j.cpc.2007.01.009

Torras, J., Seabra, G. d. M., Deumens, E., Trickey, S. B., & Roitberg, A. E. (2008). A versatile AMBER-Gaussian QM/MM interface through PUPIL. *Journal of Computational Chemistry, 29*(10), 1564-1573. doi: 10.1002/jcc.20915

Torras, J., Seabra, G. d. M., & Roitberg, A. E. (2009). A Multiscale Treatment of Angeli's Salt Decomposition. *Journal of Chemical Theory and Computation, 5*(1), 37-46. doi: 10.1021/ct800236d

Valiev, M., Bylaska, E. J., Govind, N., Kowalski, K., Straatsma, T. P., Van Dam, H. J. J., . . . de Jong, W. A. (2010). NWChem: A comprehensive and scalable open-source solution for large scale molecular simulations. *Computer Physics Communications, 181*(9), 1477-1489. doi: 10.1016/j.cpc.2010.04.018

Walker, R. C., Crowley, M. F., & Case, D. A. (2008). The implementation of a fast and accurate QM/MM potential method in Amber. *Journal of Computational Chemistry, 29*(7), 1019-1031. doi: 10.1002/jcc.20857

Warren, J. G., Revilla-López, G., Alemán, C., Jiménez, A. I., Cativiela, C., & Torras, J. (2010). Conformational Preferences of Proline Analogues with a Fused Benzene Ring. *The Journal of Physical Chemistry B, 114*(36), 11761-11770. doi: 10.1021/jp105456r

Warshel, A., & Levitt, M. (1976). Theoretical studies of enzymic reactions: Dielectric, electrostatic and steric stabilization of the carbonium ion in the reaction of lysozyme. *Journal of Molecular Biology, 103*(2), 227-249. doi: 10.1016/0022-2836(76)90311-9

Yang, W. (1991). Direct calculation of electron density in density-functional theory. *Physical Review Letters, 66*(11), 1438-1441.

Zhang, W., Wu, C., & Duan, Y. (2005). Convergence of replica exchange molecular dynamics. *The Journal of Chemical Physics, 123*(15), 154105. doi: 10.1063/1.2056540

Table 1. External QM and MM Codes that currently interfaced to PUPIL.

	Electrostatic Embedding	Start-Stop Behavior	Cyclic Behavior	Tightly coupled Interface	MPI execution
QM codes					
deMon2k	X	X			X
GAUSSIAN09	X	X			-[a]
NWChem	X	X			X
Siesta		X	X	X	X
MNDO		X			-
MM codes					
AMBER14	X			X	
DL_POLY 2	X			X	

[a] Conventional parallel execution using threads and LINDA software.

TOC