Open-source multi-GPU-accelerated QM/MM simulations with AMBER and QUICK

Vinícius Wilian D. Cruzeiro [1,2,*], *Madushanka Manathunga* [3,*], *Kenneth M. Merz Jr.* [3,*], *Andreas W. Götz* [1,*]

[1] San Diego Supercomputer Center, University of California San Diego, La Jolla, CA 92093, United States

[2] Department of Chemistry and Biochemistry, University of California San Diego, La Jolla, CA 92093, United States

[3] Department of Chemistry, Department of Biochemistry and Molecular Biology, Institute of Cyber-Enabled Research, Michigan State University, East Lansing, Michigan 48824, United States

ABSTRACT

The quantum mechanics/molecular mechanics (QM/MM) approach is an essential and well-established tool in computational chemistry that has been widely applied in a myriad of biomolecular problems in the literature. In this publication, we report the integration of the QUantum Interaction Computational Kernel (QUICK) program as an engine to perform electronic structure calculations in QM/MM simulations with AMBER. This integration is available through either a file-based interface (FBI) or an application programming interface (API). Since QUICK is an open-source GPU-accelerated code with multi-GPU parallelization, users can take advantage of "free of charge" GPU-acceleration in their QM/MM simulations. In this work, we discuss implementation details and give usage examples. We also investigate energy conservation in typical QM/MM simulations performed at the microcanonical ensemble. Finally, benchmark results for two representative systems, the N-methylacetamide (NMA) molecule and the photoactive yellow protein (PYP) in bulk water, show the performance of QM/MM simulations with QUICK and AMBER using a varying number of CPU cores and GPUs. Our results highlight the acceleration obtained from a single or multiple GPUs; we observed speedups of up to 38x between a single GPU vs. a single CPU core and of up to 2.6x when comparing four GPUs to a single GPU. Results also reveal speedups of up to 3.5x when the API is used instead of FBI.

INTRODUCTION

Quantum mechanics/molecular mechanics (QM/MM) simulations have been extensively employed to address problems encompassing a wide range of fields, such as enzymatic reactions, photochemistry, charge transfer, drug design, and material science.[1–11] The use of quantum mechanical (QM) electronic structure calculations is necessary for the study of problems in which significant rearrangement of electron density occurs, such as chemical reactions or electron transfers processes. However, the use of QM methods becomes increasingly expensive for larger systems, which often makes their practical use prohibitive. QM/MM is a multiscale approach that leverages the outstanding computational efficiency of molecular mechanics (MM) methods. The system is partitioned into a QM region containing the chemically relevant region and a MM region consisting of the surroundings, generally described with a MM force field. Even with this approximation and the reduction of the system treated quantum mechanically, the computational cost in the QM/MM approach is still dominated by the representation of the QM region, specially when *ab initio* methods are employed.

Significant progress has been made in accelerating QM calculations, such as the use of novel methodologies (*e.g.*, the fast multipole method[12, 13] and the density fitting approach[14–18]) and exploiting rapidly evolving hardware.[19, 20] During the last decade graphics processing unit (GPU) acceleration has revolutionized the performance of computational chemistry applications, outperforming central processing unit (CPU) implementations. Examples of that can be found in *ab initio* electronic structure calculations[21–40] but also in other areas such as classical molecular dynamics (MD).[41–49] One software in the first category is the QUantum Interaction Computational Kernel (QUICK) program.[31, 32, 37, 38] QUICK is an open-source GPU-accelerated application capable of obtaining Hartree-Fock (HF) and density functional theory (DFT) energies and gradients and has a recently implemented multi-GPU functionality.[32] The GPU- and multi-GPU-acceleration implementations in QUICK only apply during the computation of the electronic repulsion integrals (ERIs) and exchange-correlation potential in the case of DFT and their derivatives since these are the most computationally intensive computations. It has been shown that the multi-GPU implementation has good load balancing coupled with high parallel efficiency.[32] Another strategy used in QUICK is to perform the GPU computations asynchronously from the remaining operations, such as the one-electron integrals which are computed on the CPUs hosting the GPU cards. This strategy further speeds up the calculations.

QM/MM features are freely available as part of AMBER[50] using various electronic structure software[51] (*e.g.*, Gaussian,[52] Orca,[53] TeraChem,[33] and others). However, there are two major drawbacks with these implementations. First, the integration between AMBER and the external electronic structure software takes place through a file-based interface, which, as will be discussed herein, suffers from performance penalties. Second, because the electronic structure software packages are from third-party developers, they may require a license fee, and are generally not open-source. These drawbacks make it more of a challenge to develop important QM/MM improvements, such as corrections for long-range electrostatics.[54, 55]

In this work, we present the integration of QUICK as the driver responsible for QM calculations in QM/MM simulations performed with AMBER.[50] This integration is available in two options: either through a file-based interface (FBI) that prepares input files, executes the calculation, and parses the output results or through an application programming interface (API) that accesses QUICK directly using a library. The latest version of QUICK as a stand-alone application will be part of the upcoming AmberTools suite version 21 release, also bringing the QM/MM integration discussed in this work. Since AmberTools is open-source and general-purpose GPUs are available at a relatively low cost, users can take advantage of “free of charge” GPU-accelerated QM/MM simulations on hardware ranging from desktop computers to supercomputing clusters.

THE QM/MM APPROACH

In the QM/MM approach, the total energy of the system is expressed as follows:

$$E = E_{QM} + E_{MM} + E_{QM/MM} \tag{1}$$

where E_{QM} and E_{MM} describe, respectively, the isolated QM and MM regions, and $E_{QM/MM}$ is the coupling term describing the interactions between the QM and MM regions.

When performing QM/MM calculations, in addition to specifying the models to be used to represent the QM and MM regions, the form of the $E_{QM/MM}$ term must be specified. The most straightforward representation is called mechanical embedding (ME), where the $E_{QM/MM}$ term is treated at the MM level using van der Waals and electrostatic nonbonded interactions. ME might use point charges that are fixed or derived on-the-fly from the electronic structure calculation in the QM region at each simulation step. In another representation, called electrostatic embedding (EE), the quantum electronic density is explicitly exposed to the MM region’s surrounding point charges during the electronic calculation for the QM region. In EE, van der Waals interactions are computed in the same way as in ME. Further technical details about the QM/MM approach in a beginner-friendly format can be found elsewhere.[51, 56]

QUICK INTEGRATION WORKFLOW

The QUICK integration uses the SANDER MD engine, which supports QM/MM functionality in AMBER[50] and the present integration takes advantage of the existing QM/MM infrastructure. Therefore, the QM/MM setup, including the identification of atoms in the QM region, follows the same scheme already present in SANDER.[51, 57] If the QM/MM crosses covalent bonds, the existing QM/MM module automatically sets up the link atoms.[51, 57] Using our QUICK integration, users can utilize existing SANDER features, including geometry optimization and enhanced sampling schemes (*e.g.*, umbrella sampling or replica exchange MD).

Figure 1 illustrates a QM/MM simulation using SANDER coupled with QUICK. Users must choose one of two options for the entire simulation: the application programming interface (API) or file-based interface (FBI). The FBI follows the QM/MM implementation[51] already in SANDER to perform QM/MM simulations with other electronic structure software packages including Gaussian,[52] Orca,[53] and TeraChem.[33] At each MD step, FBI takes care of preparing an input file, executing the QM calculation by calling the QUICK executable, and reading the output file. In contrast, the API implementation is a new addition to SANDER and has been specially designed for QUICK. In the API, there is a direct communication between SANDER and QUICK through a library, without the need for any I/O operations. The API is expected to save time by executing the QM calculation setup only once during the entire QM/MM simulation, contrary to FBI, where this setup step is done at every MD step when the QUICK executable is called. The API also enables MPI parallel calculations in which both SANDER and QUICK execute in parallel fashion, with the capability of utilizing CPU cores and GPUs across multiple compute nodes. This is not possible through the FBI due to restrictions of systems calls from within an MPI parallel program.

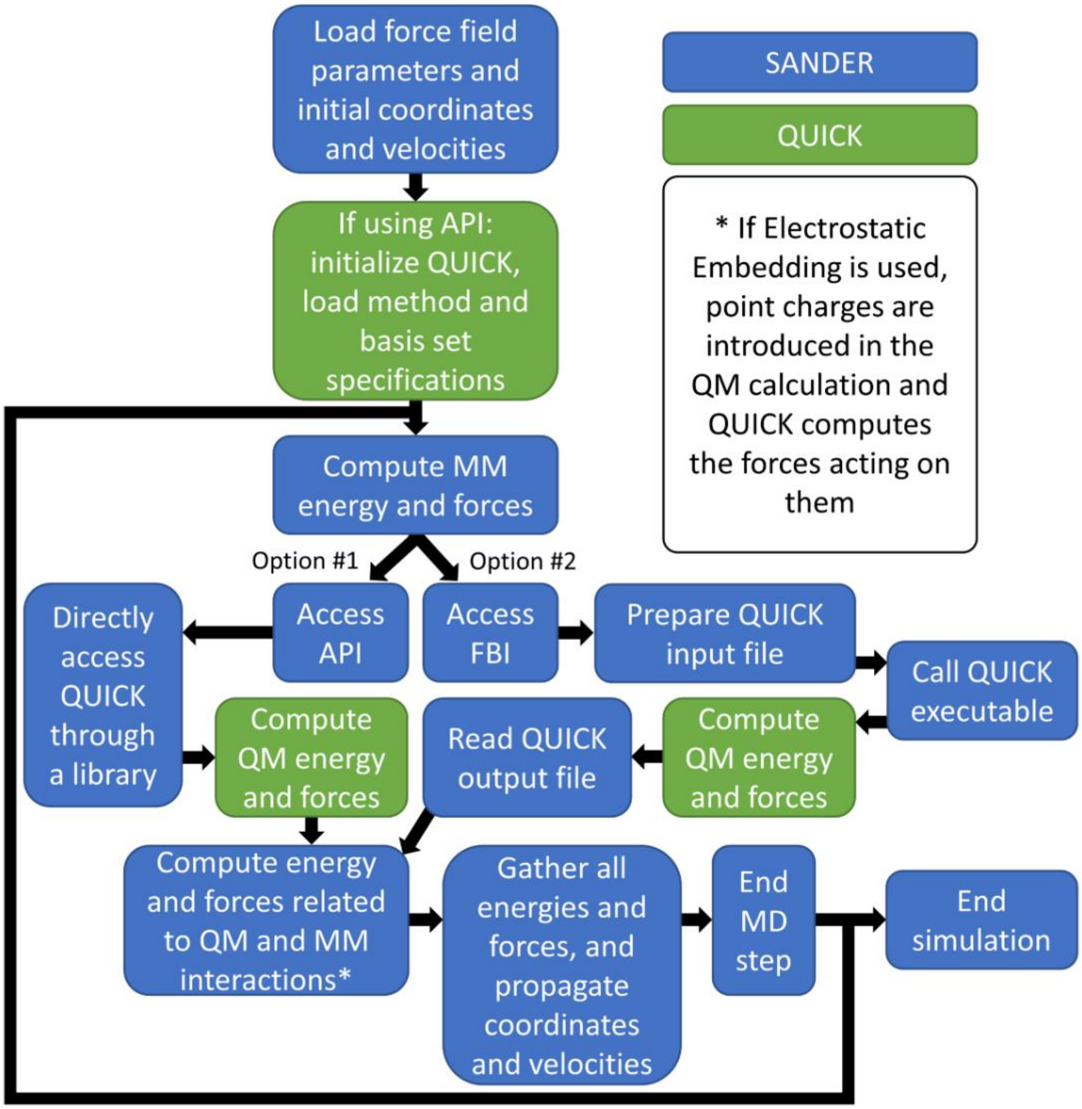
Load force field parameters and initial coordinates and velocities
SANDER
QUICK
If using API: initialize QUICK, load method and basis set specifications
* If Electrostatic Embedding is used, point charges are introduced in the QM calculation and QUICK computes the forces acting on them
Compute MM energy and forces
Option #1
Option #2
Directly access QUICK through a library
Access API
Access FBI
Prepare QUICK input file
Call QUICK executable
Compute QM energy and forces
Read QUICK output file
Compute QM energy and forces
Compute energy and forces related to QM and MM interactions*
Gather all energies and forces, and propagate coordinates and velocities
End MD step
End simulation

Figure 1. Flowchart illustrating a QM/MM simulation with SANDER using QUICK for the electronic structure calculations. Users must choose to use either the application programming interface (API) or file-based interface (FBI) during the entire simulation.

USAGE

To use the features of the QUICK integration reported in this work, users must compile AMBER with QUICK support. This compilation will enable the QUICK API functionalities in SANDER and will also generate QUICK stand-alone executables, likewise for other AmberTools applications. Users must load $AMBERHOME/amber.sh before executing QM/MM simulations since this step ensures that the location of the necessary executables and libraries are set in the environmental variables of the operating system. Please refer to the AmberTools version 21 manual for more information.

Test suites for the API and FBI functionalities, which also serve as usage examples, can be found at the following locations within AMBER’s source: $AMBERHOME/test/qmmm_Quick and $AMBERHOME/test/qmmm_EXTERN/*Quick, respectively.

File-based interface (FBI)

Four different QUICK executables can be compiled: *quick* (serial version), *quick.MPI* (parallel version), *quick.cuda* (serial GPU-accelerated version), and *quick.cuda.MPI* (multi-GPU-accelerated version). The FBI can be accessed using the *sander* (serial version) executable. Below is an example of the modifications necessary in the SANDER input file to perform a mechanical embedding QM/MM simulation. In this example, the first two residues of the system are assigned to the QM region, and the simulation is executed at the B3LYP/def2-SVP level with *quick.cuda.MPI* using 2 GPUs:

```
&cntrl
 ...
 ifqnt = 1,
/
&qmmm
 qmmask = ':1-2',
 qm_theory = 'extern',
 qmmm_int = 5,
/
&quick
 method   = 'B3LYP',
 basis    = 'def2-svp',
 executable = 'quick.cuda.MPI',
 do_parallel = 'mpirun -np 2',
/
```

where ifqnt set to 1 activates the QM/MM functionality, qmmask specifies the QM region, qm_theory set as ‘extern’ indicates that the FBI will be used, and qmmm_init set to 5 specifies the use of mechanical embedding. The executable flag can be set to any of the four QUICK executables, and the do_parallel flag must be specified only if using one of the parallel versions of QUICK. It is important to emphasize that some machines may require a command other than *mpirun*, depending on the MPI library being used.

Application programming interface (API)

When using the API implementation, each version of QUICK can be accessed through a different library. Each library has been linked to its corresponding version in SANDER. Therefore, the functionalities of the serial and parallel versions of QUICK can be accessed from the *sander* and *sander.MPI* executables, respectively. Furthermore, the functionalities of the serial GPU-accelerated and multi-GPU-accelerated versions of QUICK can be accessed from newly released SANDER executables called *sander.quick.cuda* and *sander.quick.cuda.MPI*. These executables are identical to *sander* and *sander.MPI* in all SANDER functionalities, except they perform QM/MM calculations with the QUICK library through the API.

In the example below, we present the modifications necessary in the SANDER input file to perform a QM/MM simulation with electrostatic embedding. Unlike for the FBI case, simulations using serial, parallel, serial GPU-accelerated, and multi-GPU-accelerated QUICK functionalities can all use the same input file.

```
&cntrl
 ...
 ifqnt = 1,
/
&qmmm
 qmmask = ':1-2',
 qm_theory = 'quick',
 qmmm_int = 1,
 qm_ewald = 0,
/
&quick
 method   = 'B3LYP',
 basis    = 'def2-svp',
/
```

where ifqnt set to 1 activates the QM/MM functionality, qmmask specifies the QM region, qm_theory set as ‘quick’ makes use of API, qmmm_int set to 1 specifies the use of electrostatic embedding and qm_ewald set to 0 indicates that the QM/MM interactions should be truncated at a given cutoff (the cut variable is specified in the &cntrl namelist).

BENCHMARKS

This section presents benchmark results of typical QM/MM simulations of two different systems in bulk water. The simulations were executed on the Expanse cluster, part of the San Diego supercomputer center (SDSC), on a node that contains four NVIDIA Volta V100-SXM2 type GPUs and Intel Xeon (R) Gold 6248 (2.50 GHz) CPUs. The two systems chosen were the N-methylacetamide (NMA) molecule and the photoactive yellow protein (PYP), which has been previously studied using QM/MM.[61] For the first system, the full NMA molecule with 12 atoms encompassed the QM region, and the 511 water molecules present in the simulation box as the MM region, described by the SPC/Fw force field.[60] For the second system, two different QM regions were considered.[61] QM region 1 contains the *p*-coumaric acid chromophore and the S-C bond from CYS-69, giving a total of 22 atoms and a charge of -1.[61] QM region 2 contains the same atoms as in QM region 1 with the addition of the side chains of GLH-46 and TYR-42, yielding a total of 49 atoms and a charge of -1.[61] Excluding the selected QM region, the protein is represented by the ff99SB force field,[62] the chromophore by the GAFF force field,[63] and the 10758 water molecules in the simulation box by SPC/Fw.[60] The atoms in the QM regions of the NMA and PYP simulations are shown in **Figure 2**.

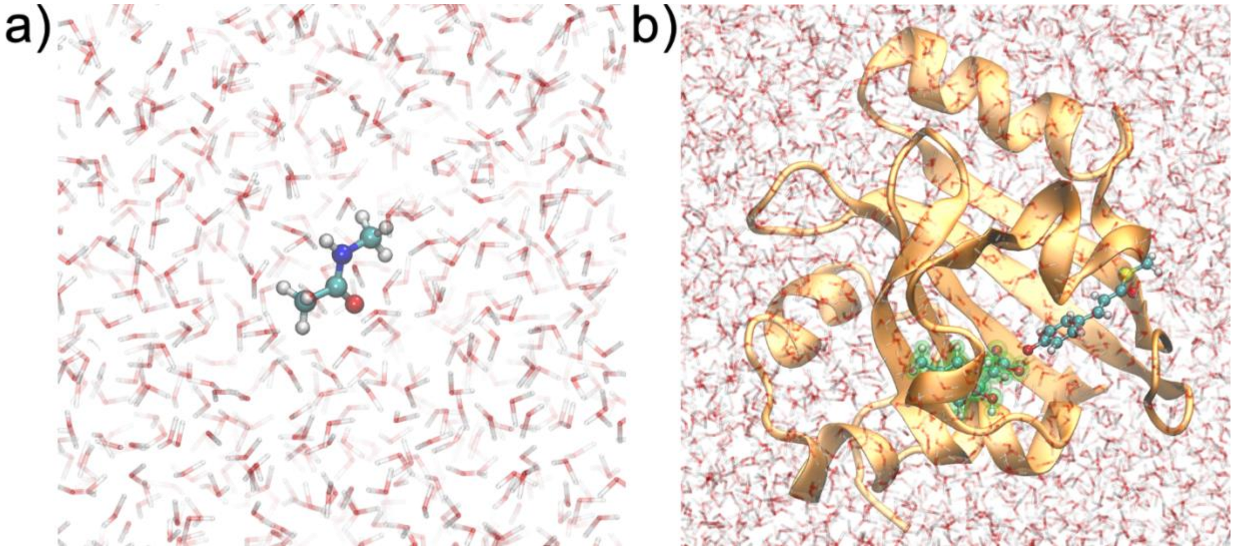

Figure 2. QM regions of a) NMA and b) PYP simulations in bulk water. Atoms highlighted in green for PYP are only present in QM region 2.

All simulations used a timestep of 0.5 fs, a QM/MM cutoff of 8 Å, and the B3LYP functional in the QM calculations. **Table 1** presents the computational efficiencies for varying numbers of CPU cores and GPUs. The table shows results for both electrostatic embedding and mechanical embedding, two typical basis sets (6-31G* and def2-SVP), and simulations with QUICK's API and FBI integrations. The number of basis functions for NMA and PYP QM regions 1 and 2 is, respectively, 89, 217, and 440 for 6-31G* and 110, 244, and 509 for def2-SVP. Timings for the 6-31G** basis set, which has the same number of basis functions as def2-SVP, are presented in the Supporting Information.

Table 1. Computational performance (in ps/day) of QM/MM simulations executed with QUICK and AMBER.[a]

			NMA				**PYP (QM region 1)**				**PYP (QM region 2)**			
			6-31G*		**def2-SVP**		**6-31G***		**def2-SVP**		**6-31G***		**def2-SVP**	
QM/MM type	**# of CPUs**	**# of GPUs**	**API**	**FBI**	**API**	**FBI**	**API**	**FBI**	**API**	**FBI**	**API**	**FBI**	**API**	**FBI**
EE	1	0	2.71	2.38	1.76	1.65	0.29	0.28	0.22	0.22	0.07	0.07	0.05	0.05
	1	1	19.43	7.66	17.16	7.71	4.53	2.52	4.52	3.17	1.80	1.34	1.55	1.30
	2	0	4.64	3.61	3.21	2.74	0.55	0.51	0.41	0.40	0.14	0.13	0.10	0.09
	2	2	25.38	7.79	22.83	8.06	6.56	2.93	6.86	3.92	2.98	1.87	2.62	1.96
	4	0	6.30	4.54	5.12	3.99	0.93	0.81	0.67	0.64	0.25	0.24	0.16	0.15
	4	4	27.00	7.59	25.45	7.93	8.81	3.24	9.89	4.62	4.50	2.31	3.98	2.57
	8	0	10.23	6.20	7.45	5.30	1.46	1.18	1.12	1.03	0.41	0.38	0.28	0.27
	16	0	14.39	7.35	11.86	7.03	2.17	1.58	1.90	1.63	0.73	0.64	0.46	0.44
ME	1	0	2.80	2.45	1.82	1.70	0.29	0.28	0.22	0.22	0.07	0.07	0.05	0.05
	1	1	23.80	8.26	21.80	8.55	5.92	2.93	6.74	4.08	2.65	1.77	2.66	2.05
	2	0	4.78	3.71	3.30	2.81	0.55	0.50	0.42	0.40	0.13	0.12	0.10	0.10
	2	2	27.87	8.00	26.24	8.49	7.90	3.18	9.25	4.62	4.17	2.27	4.32	2.79
	4	0	6.43	4.62	5.21	4.07	0.92	0.80	0.67	0.64	0.23	0.23	0.16	0.16
	4	4	28.95	7.69	27.95	8.20	10.09	3.41	12.41	5.12	5.75	2.59	6.15	3.39
	8	0	10.56	6.31	7.68	5.40	1.45	1.16	1.12	1.02	0.38	0.36	0.28	0.27
	16	0	14.76	7.46	12.14	7.13	2.16	1.57	1.90	1.62	0.68	0.60	0.47	0.45

a) Simulations for NMA or PYP in bulk water using a timestep of 0.5 fs and a QM/MM cutoff of 8 Å. QM calculations used the B3LYP functional with either the 6-31G* or def2-SVP basis set. EE = electrostatic embedding, ME = mechanical embedding, API = application programming interface, FBI = file-based interface

As expected, simulations with the API are faster than with the FBI. Speedups of up to 3.5x are observed for the smallest QM size evaluated and up to 2.2x for the largest QM size. The computational efficiency systematically improves with the number of CPU cores or GPUs considered. The power of GPU-acceleration can be inferred from **Table 1**. The performance gain from a single GPU exceeds that of 16 CPU cores. This observation is consistent with the previously documented DFT performance for QUICK serial GPU and MPI parallel versions.[31] Simulations with a single GPU using the API are up to 12x and 38x faster than using only a single CPU core for, respectively, the smallest and largest QM region sizes considered. **Table 1** also highlights further acceleration that can be obtained from multiple GPUs. Simulations with four GPUs are up to 1.5x and 2.6x faster than with a single GPU, respectively, for the smallest and largest QM region sizes considered.

Aiming at further investigating the contributions of different tasks to the total computational time in QUICK, **Table 2** dissects the time spent during one-electron integrals (OEI), electron repulsion integrals (ERI, two-electron integrals), and exchange-correlation (XC) quadrature and their gradients for one MD step of the QM/MM simulations shown in **Table 1** using the API with 1 CPU and 1 GPU.

Table 2. Timings (in seconds) of different tasks on a single QM/MM MD step executed with QUICK and AMBER.[a]

	NMA				PYP (QM region 1)				PYP (QM region 2)			
	6-31G*		def2-SVP		6-31G*		def2-SVP		6-31G*		def2-SVP	
	EE	ME	EE	ME	EE	ME	EE	ME	EE	ME	EE	ME
SCF OEI	0.08	0.01	0.11	0.01	0.44	0.05	0.55	0.06	1.56	0.17	2.18	0.21
SCF ERI	0.12	0.12	0.14	0.14	1.13	1.13	0.93	0.93	3.81	3.80	3.70	3.70
SCF XC	0.30	0.30	0.40	0.40	1.56	1.56	1.77	1.77	2.67	2.66	3.23	3.23
OEI gradients	0.54	0.09	0.73	0.11	3.26	0.43	3.91	0.47	11.09	1.31	14.68	1.58
ERI gradients	0.38	0.38	0.43	0.43	1.70	1.70	1.52	1.51	5.49	5.49	5.54	5.54
XC gradients	0.19	0.19	0.24	0.24	0.48	0.48	0.55	0.55	1.20	1.20	1.44	1.44

a) Simulations were performed with the API and with 1 CPU and 1 GPU. EE = electrostatic embedding, ME = mechanical embedding, OEI = one-electron integrals, ERI = electron repulsion integrals (two-electron integrals), XC = exchange-correlation

Table 2 shows that, as expected, one electron operations are the only ones affected by external point charges in the QM calculation. When electrostatic embedding is employed, the costliest computation is the one-electron gradients. This observation is not true when mechanical embedding is used. As discussed in the next section, the performance in the one-electron operations is an area that can be improved in future versions of QUICK.

As documented previously,[31] the single GPU version of QUICK is capable of performing HF and DFT gradient calculations several times faster than the single GPU version of GAMESS[34, 35, 64] when external point charges are not present. Since the cost of QM/MM is dominated by the computation of energies and forces of the QM region, the QUICK/AMBER API and FBI implementations are expected to perform faster mechanical embedding QM/MM simulations than the GAMESS/AMBER FBI implementation. Since the GAMESS/AMBER integration is currently unable to perform electrostatic embedding QM/MM simulations,[51] we have not compared QUICK and GAMESS in the current study.

CONCLUSIONS AND FUTURE DIRECTIONS

We have presented and discussed the integration of QUICK as the driver responsible for *ab initio* electronic structure calculations in QM/MM simulations performed with AMBER. Users can access this integration in two ways. First, through a file-based interface (FBI) that follows the same approach currently used in AMBER to perform QM/MM simulations with other quantum chemistry software. Second, through a novel implementation that uses an application programming interface (API) for direct communication with QUICK. The API has advantages over FBI, such as the absence of I/O operations and a reduced computation cost to setup the QM calculations. With the API, the QM setup is done only once at the beginning of the QM/MM simulation, in contrast with the FBI that needs to perform this operation at every MD step. Even though our API implementation has been designed specifically for QUICK, other electronic structure software packages with compatible functionalities (*i.e.*, whose features can be accessed through a library) can now make use of this novel architecture for faster QM/MM simulations in AMBER. Hence, the API can be readily exploited to seamlessly integrate QUICK with other MM software packages. Both the FBI and API QM/MM integrations can take advantage of useful features available in AMBER, such as geometry optimization and enhanced sampling approaches.

The API and FBI QM/MM integrations can perform calculations with electrostatic embedding (EE) or mechanical embedding (ME). As previously discussed, EE QM/MM simulations with *ab* *initio* calculations in AMBER currently suffer from the lack of a treatment for long-range electrostatic interactions between the QM and MM region and of the QM region with its periodic images when periodic boundary conditions are employed. One possible solution for this would be implementing particle mesh Ewald (PME) type methods[58, 59] to properly capture the physics inherent to the long-range QM/MM electrostatics. Such an approach has been previously presented.[54] In a recent publication, alternative approaches for describing long-range electrostatic interactions in *ab initio* QM/MM calculations were presented.[55] Since AMBER already has a PME based implementation for QM/MM long-range electrostatics compatible with semiempirical and density functional tight binding (DFTB) Hamiltonians, the implementation of long-range corrections for *ab initio* methods following one of the approaches highlighted above could benefit from this existing infrastructure. Such an implementation is planned and would be expedited in the API QM/MM integration since it allows direct data transfer between AMBER and QUICK.

In this work we have performed QM/MM simulations for the N-methylacetamide (NMA) molecule and the photoactive yellow protein (PYP) inside a water droplet or in bulk water. For the simulations in a water droplet, presented at the Supporting Information, the QM/MM cutoff was extended beyond the limits of the system to ensure that all interactions are taken into consideration, and it has been shown that the default SCF convergence criterion used in QUICK is satisfactory to ensure energy conservation in microcanonical ensemble simulations. Energy conservation was also observed in longer QM/MM simulations performed for NMA (see Supporting Information). Benchmark simulations using a varying number of CPUs and GPUs have been conducted in bulk water. The benchmarks highlight the GPU-acceleration speedup and show that an improved performance is obtained when multiple GPUs are employed. As expected, the benchmarks indicate that the API integration has a better computational efficiency than the FBI integration.

QUICK's GPU-acceleration acts on the computation of the most expensive parts of the electronic structure calculation: electron repulsion integrals (ERI), numerical quadrature of the exchange-correlation (XC) energy in the case DFT is used, and their derivatives. The one-electron integrals and gradients are asynchronously computed on the CPU during the ERI and XC computations to maximize the overall efficiency. In general, one electron computations are significantly faster than the ERI and XC computations. However, since the ERI and XC operations are GPU-accelerated, the cost of one electron tasks may significantly increase or even surpass that of the ERI and XC in extreme cases, such as when many point charges are present around the QM region, as has been observed in this work. In the current QUICK implementation, the Coulomb attraction integral/gradient computation, which dominates the one-electron integral/gradient times, is performed considering all pairwise interactions between electrons and individual point charges. The cost of such tasks can be drastically reduced by using, for example, the fast multipole method (FMM).[12, 13] Therefore, there is room for improvement in QUICK's one-electron operations, currently computed on the CPU. These tasks will benefit from implementation of the FMM method combined with GPU-acceleration in future versions of QUICK.

AUTHOR INFORMATION

* E-mails: vcruzeiro@ucsd.edu, manathun@msu.edu, merz@chemistry.msu.edu, agoetz@sdsc.edu

Corresponding Authors

Kenneth M. Merz, Jr. and Andreas W. Götz

Notes

The authors declare no competing financial interest.

DATA AND SOFTWARE AVAILABILITY

Any data generated and analyzed for this study that are not included in this article and its Supplementary Information are available from the authors upon request.

AmberTools is publicly available free of charge at: https://ambermd.org . QUICK will be released as part of AmberTools starting from the upcoming version of 2021. The latest development version of QUICK can be accessed for free from GitHub: https://github.com/merzlab/QUICK .

SUPPORTING INFORMATION

PDF file: Analysis of the effect of the SCF convergence threshold on energy conservation in QM/MM simulations with QUICK. Analysis of energy conservation in longer QM/MM simulations. Benchmark timings (equivalent to Tables 1 and 2) for B3LYP/6-31G**.

ZIP file: Input files for the QM/MM simulations performed in this work.

ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support from the National Science Foundation (NSF) through grant OAC-1835144. V.C. and A.G. would like to thank the San Diego Supercomputer Center for computer time. M.M. and K.M. are grateful to the Department of Chemistry and Biochemistry and high-performance computer center (iCER HPCC) at the Michigan State University. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by the NSF under grant number ACI-1053575 (resources at the San Diego Supercomputer Center through award TG-CHE130010).

REFERENCES

[1] Ahmadi, S.; Barrios Herrera, L.; Chehelamirani, M.; Hostaš, J.; Jalife, S.; Salahub, D. R. Multiscale Modeling of Enzymes: QM-Cluster, QM/MM, and QM/MM/MD: A Tutorial Review. *Int. J. Quantum Chem.*, **2018**, *118* (9), e25558.

[2] Morzan, U. N.; Alonso de Armiño, D. J.; Foglia, N. O.; Ramírez, F.; González Lebrero, M. C.; Scherlis, D. A.; Estrin, D. A. Spectroscopy in Complex Environments from QM-MM Simulations. *Chem. Rev.*, **2018**, *118* (7), 4071–4113.

[3] Quesne, M. G.; Borowski, T.; de Visser, S. P. Quantum Mechanics/Molecular Mechanics Modeling of Enzymatic Processes: Caveats and Breakthroughs. *Chem. - A Eur. J.*, **2016**, *22* (8), 2562–2581.

[4] Lin, H.; Truhlar, D. G. QM/MM: What Have We Learned, Where Are We, and Where Do We Go from Here? *Theor. Chem. Acc.*, **2007**, *117* (2), 185–199.

[5] Senn, H. M.; Thiel, W. QM/MM Methods for Biomolecular Systems. *Angew. Chemie Int. Ed.*, **2009**, *48* (7), 1198–1229.

[6] van der Kamp, M. W.; Mulholland, A. J. Combined Quantum Mechanics/Molecular

Mechanics (QM/MM) Methods in Computational Enzymology. *Biochemistry*, **2013**, *52* (16), 2708–2728.

[7] Zheng, M.; Waller, M. P. Adaptive Quantum Mechanics/Molecular Mechanics Methods. *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, **2016**, *6* (4), 369–385.

[8] Nogueira, J. J.; González, L. Computational Photophysics in the Presence of an Environment. *Annu. Rev. Phys. Chem.*, **2018**, *69* (1), 473–497.

[9] *QM/MM Studies of Light-Responsive Biological Systems*; Andruniów, T., Olivucci, M., Eds.; Challenges and Advances in Computational Chemistry and Physics; Springer International Publishing: Cham, 2021; Vol. 31.

[10] Groenhof, G. *Introduction to QM/MM Simulations*. In: Monticelli L., Salonen E. (eds) Biomolecular Simulations. Methods in Molecular Biology (Methods and Protocols), vol 924. Humana Press, Totowa, NJ. 2013.

[11] Hofer, T. S.; de Visser, S. P. Editorial: Quantum Mechanical/Molecular Mechanical Approaches for the Investigation of Chemical Systems – Recent Developments and Advanced Applications. *Front. Chem.*, **2018**, *6*.

[12] White, C. A.; Head-Gordon, M. Derivation and Efficient Implementation of the Fast Multipole Method. *J. Chem. Phys.*, **1994**, *101* (8), 6593–6605.

[13] White, C. A.; Johnson, B. G.; Gill, P. M. W.; Head-Gordon, M. The Continuous Fast Multipole Method. *Chem. Phys. Lett.*, **1994**, *230* (1–2), 8–16.

[14] Parrish, R. M.; Burns, L. A.; Smith, D. G. A.; Simmonett, A. C.; DePrince, A. E.; Hohenstein, E. G.; Bozkaya, U.; Sokolov, A. Y.; Di Remigio, R.; Richard, R. M.; Gonthier, J. F.; James, A. M.; McAlexander, H. R.; Kumar, A.; Saitow, M.; Wang, X.; Pritchard, B. P.; Verma, P.; Schaefer, H. F.; Patkowski, K.; King, R. A.; Valeev, E. F.; Evangelista, F.

A.; Turney, J. M.; Crawford, T. D.; Sherrill, C. D. Psi4 1.1: An Open-Source Electronic Structure Program Emphasizing Automation, Advanced Libraries, and Interoperability. *J. Chem. Theory Comput.*, **2017**, *13* (7), 3185–3197.

[15] DePrince, A. E.; Sherrill, C. D. Accuracy and Efficiency of Coupled-Cluster Theory Using Density Fitting/Cholesky Decomposition, Frozen Natural Orbitals, and a t_1-Transformed Hamiltonian. *J. Chem. Theory Comput.*, **2013**, *9* (6), 2687–2696.

[16] Eichkorn, K.; Treutler, O.; Öhm, H.; Häser, M.; Ahlrichs, R. Auxiliary Basis Sets to Approximate Coulomb Potentials. *Chem. Phys. Lett.*, **1995**, *240* (4), 283–290.

[17] Neese, F. An Improvement of the Resolution of the Identity Approximation for the Formation of the Coulomb Matrix. *J. Comput. Chem.*, **2003**, *24* (14), 1740–1747.

[18] Weigend, F.; Häser, M.; Patzelt, H.; Ahlrichs, R. RI-MP2: Optimized Auxiliary Basis Sets and Demonstration of Efficiency. *Chem. Phys. Lett.*, **1998**, *294* (1–3), 143–152.

[19] Gordon, M. S.; Windus, T. L. Editorial: Modern Architectures and Their Impact on Electronic Structure Theory. *Chem. Rev.*, **2020**, *120* (17), 9015–9020.

[20] Gordon, M. S.; Barca, G.; Leang, S. S.; Poole, D.; Rendell, A. P.; Galvez Vallejo, J. L.; Westheimer, B. Novel Computer Architectures and Quantum Chemistry. *J. Phys. Chem. A*, **2020**, *124* (23), 4557–4582.

[21] Ufimtsev, I. S.; Martínez, T. J. Quantum Chemistry on Graphical Processing Units. 1. Strategies for Two-Electron Integral Evaluation. *J. Chem. Theory Comput.*, **2008**, *4* (2), 222–231.

[22] Ufimtsev, I. S.; Martinez, T. J. Quantum Chemistry on Graphical Processing Units. 2. Direct Self-Consistent-Field Implementation. *J. Chem. Theory Comput.*, **2009**, *5* (4), 1004–1015.

[23] Fales, B. S.; Levine, B. G. Nanoscale Multireference Quantum Chemistry: Full

Configuration Interaction on Graphical Processing Units. *J. Chem. Theory Comput.*, **2015**, *11* (10), 4708–4716.

[24] Asadchev, A.; Gordon, M. S. Fast and Flexible Coupled Cluster Implementation. *J. Chem. Theory Comput.*, **2013**, *9* (8), 3385–3392.

[25] DePrince, A. E.; Hammond, J. R. Coupled Cluster Theory on Graphics Processing Units I. The Coupled Cluster Doubles Method. *J. Chem. Theory Comput.*, **2011**, *7* (5), 1287–1295.

[26] Luehr, N.; Ufimtsev, I. S.; Martínez, T. J. Dynamic Precision for Electron Repulsion Integral Evaluation on Graphical Processing Units (GPUs). *J. Chem. Theory Comput.*, **2011**, *7* (4), 949–954.

[27] Liu, F.; Sanchez, D. M.; Kulik, H. J.; Martínez, T. J. Exploiting Graphical Processing Units to Enable Quantum Chemistry Calculation of Large Solvated Molecules with Conductor-like Polarizable Continuum Models. *Int. J. Quantum Chem.*, **2019**, *119* (1), e25760.

[28] Götz, A. W.; Wölfle, T.; Walker, R. C. Quantum Chemistry on Graphics Processing Units. *Annu. Rep. Comput. Chem.*, **2010**, *6* (C), 21–35.

[29] Walker, R. C.; Götz, A. W. *Electronic Structure Calculations on Graphics Processing Units: From Quantum Chemistry to Condensed Matter Physics*; Walker, R. C., Götz, A. W., Eds.; wiley: Chichester, UK, 2016.

[30] Andrade, X.; Aspuru-Guzik, A. Real-Space Density Functional Theory on Graphical Processing Units: Computational Approach and Comparison to Gaussian Basis Set Methods. *J. Chem. Theory Comput.*, **2013**, *9* (10), 4360–4373.

[31] Manathunga, M.; Miao, Y.; Mu, D.; Götz, A. W.; Merz, K. M. Parallel Implementation of Density Functional Theory Methods in the Quantum Interaction Computational Kernel Program. *J. Chem. Theory Comput.*, **2020**, acs.jctc.0c00290.

[32] Manathunga, M.; Chi, J.; Cruzeiro, V. W. D.; Miao, Y.; Mu, D.; Arumugam, K.; Keipert, K.; Aktulga, H. M.; Merz, K. M.; Götz, A. W. Harnessing the Power of Multi-GPU-Acceleration into the Quantum Interaction Computational Kernel Program. *ChemRxiv* Preprint, **2021**. https://doi.org/10.26434/chemrxiv.13769209.v1 (accessed Feb 12, 2021).

[33] Ufimtsev, I. S.; Martinez, T. J. Quantum Chemistry on Graphical Processing Units. 3. Analytical Energy Gradients, Geometry Optimization, and First Principles Molecular Dynamics. *J. Chem. Theory Comput.*, **2009**, *5* (10), 2619–2628.

[34] Asadchev, A.; Allada, V.; Felder, J.; Bode, B. M.; Gordon, M. S.; Windus, T. L. Uncontracted Rys Quadrature Implementation of up to G Functions on Graphical Processing Units. *J. Chem. Theory Comput.*, **2010**, *6* (3), 696–704.

[35] Asadchev, A.; Gordon, M. S. New Multithreaded Hybrid CPU/GPU Approach to Hartree-Fock. *J. Chem. Theory Comput.*, **2012**, *8* (11), 4166–4176.

[36] Yasuda, K. Two-Electron Integral Evaluation on the Graphics Processor Unit. *J. Comput. Chem.*, **2008**, *29* (3), 334–342.

[37] Miao, Y.; Merz, K. M. Acceleration of Electron Repulsion Integral Evaluation on Graphics Processing Units via Use of Recurrence Relations. *J. Chem. Theory Comput.*, **2013**, *9* (2), 965–976.

[38] Miao, Y.; Merz, K. M. Acceleration of High Angular Momentum Electron Repulsion Integrals and Integral Derivatives on Graphics Processing Units. *J. Chem. Theory Comput.*, **2015**, *11* (4), 1449–1462.

[39] Hohenstein, E. G.; Luehr, N.; Ufimtsev, I. S.; Martínez, T. J. An Atomic Orbital-Based Formulation of the Complete Active Space Self-Consistent Field Method on Graphical Processing Units. *J. Chem. Phys.*, **2015**, *142* (22), 224103.

[40] Fales, B. S.; Martínez, T. J. Efficient Treatment of Large Active Spaces through Multi-GPU Parallel Implementation of Direct Configuration Interaction. *J. Chem. Theory Comput.*, **2020**.

[41] Friedrichs, M. S.; Eastman, P.; Vaidyanathan, V.; Houston, M.; Legrand, S.; Beberg, A. L.; Ensign, D. L.; Bruns, C. M.; Pande, V. S. Accelerating Molecular Dynamic Simulation on Graphics Processing Units. *J. Comput. Chem.*, **2009**, *30* (6), 864–872.

[42] Götz, A. W.; Williamson, M. J.; Xu, D.; Poole, D.; Le Grand, S.; Walker, R. C. Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 1. Generalized Born. *J. Chem. Theory Comput.*, **2012**, *8* (5), 1542–1555.

[43] Salomon-Ferrer, R.; Götz, A. W.; Poole, D.; Le Grand, S.; Walker, R. C. Routine Microsecond Molecular Dynamics Simulations with AMBER on GPUs. 2. Explicit Solvent Particle Mesh Ewald. *J. Chem. Theory Comput.*, **2013**, *9* (9), 3878–3888.

[44] Biagini, T.; Petrizzelli, F.; Truglio, M.; Cespa, R.; Barbieri, A.; Capocefalo, D.; Castellana, S.; Tevy, M. F.; Carella, M.; Mazza, T. Are Gaming-Enabled Graphic Processing Unit Cards Convenient for Molecular Dynamics Simulation? *Evol. Bioinforma.*, **2019**, *15*, 1–3.

[45] Lee, T. S.; Cerutti, D. S.; Mermelstein, D.; Lin, C.; Legrand, S.; Giese, T. J.; Roitberg, A.; Case, D. A.; Walker, R. C.; York, D. M. GPU-Accelerated Molecular Dynamics and Free Energy Methods in Amber18: Performance Enhancements and New Features. *J. Chem. Inf. Model.*, **2018**, *58* (10), 2043–2050.

[46] Harvey, M. J.; Giupponi, G.; De Fabritiis, G. ACEMD: Accelerating Biomolecular Dynamics in the Microsecond Time Scale. *J. Chem. Theory Comput.*, **2009**, *5* (6), 1632–1639.

[47] Le Grand, S.; Götz, A. W.; Walker, R. C. SPFP: Speed without Compromise—A Mixed

Precision Model for GPU Accelerated Molecular Dynamics Simulations. *Comput. Phys. Commun.*, **2013**, *184* (2), 374–380.

[48] Kutzner, C.; Páll, S.; Fechner, M.; Esztermann, A.; De Groot, B. L.; Grubmüller, H. Best Bang for Your Buck: GPU Nodes for GROMACS Biomolecular Simulations. *J. Comput. Chem.*, **2015**, *36* (26), 1990–2008.

[49] Eastman, P.; Swails, J.; Chodera, J. D.; McGibbon, R. T.; Zhao, Y.; Beauchamp, K. A.; Wang, L. P.; Simmonett, A. C.; Harrigan, M. P.; Stern, C. D.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid Development of High Performance Algorithms for Molecular Dynamics. *PLoS Comput. Biol.*, **2017**, *13* (7), e1005659.

[50] Case, D. A.; Belfon, K.; Ben-Shalom, I. Y.; Brozell, S. R.; Cerutti, D. S.; Cheatham, T. E., I.; Cruzeiro, V. W. D.; Darden, T. A.; Duke, R. E.; Giambasu, G.; Gilson, M. K.; Gohlke, H.; Goetz, A. W.; Harris, R.; Izadi, S.; Izmailov, S. A.; Kasavajhala, K.; Kovalenko, A.; Krasny, R.; Kurtzman, T.; Lee, T. S.; LeGrand, S.; Li, P.; Lin, C.; Liu, J.; Luchko, T.; Luo, R.; Man, V.; Merz, K. M.; Miao, Y.; Mikhailovskii, O.; Monard, G.; Nguyen, H.; Onufriev, A.; Pan, F.; Pantano, S.; Qi, R.; Roe, D. R.; Roitberg, A.; Sagui, C.; Schott-Verdugo, S.; Shen, J.; Simmerling, C.; Skrynnikov, N. R.; Smith, J.; Swails, J.; Walker, R. C.; Wang, J.; Wilson, L.; Wolf, R. M.; Wu, X.; Xiong, Y.; Xue, Y.; York, D. M.; Kollman, P. A. AMBER 2020. University of California, San Francisco 2020.

[51] Götz, A. W.; Clark, M. A.; Walker, R. C. An Extensible Interface for QM/MM Molecular Dynamics Simulations with AMBER. *J. Comput. Chem.*, **2014**, *35* (2), 95–108.

[52] Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Petersson, G. A.; Nakatsuji, H.; Li, X.; Caricato, M.; Marenich, A. V; Bloino, J.; Janesko, B. G.; Gomperts, R.; Mennucci, B.; Hratchian, H. P.;

Ortiz, J. V; Izmaylov, A. F.; Sonnenberg, J. L.; Williams-Young, D.; Ding, F.; Lipparini, F.; Egidi, F.; Goings, J.; Peng, B.; Petrone, A.; Henderson, T.; Ranasinghe, D.; Zakrzewski, V. G.; Gao, J.; Rega, N.; Zheng, G.; Liang, W.; Hada, M.; Ehara, M.; Toyota, K.; Fukuda, R.; Hasegawa, J.; Ishida, M.; Nakajima, T.; Honda, Y.; Kitao, O.; Nakai, H.; Vreven, T.; Throssell, K.; Montgomery Jr., J. A.; Peralta, J. E.; Ogliaro, F.; Bearpark, M. J.; Heyd, J. J.; Brothers, E. N.; Kudin, K. N.; Staroverov, V. N.; Keith, T. A.; Kobayashi, R.; Normand, J.; Raghavachari, K.; Rendell, A. P.; Burant, J. C.; Iyengar, S. S.; Tomasi, J.; Cossi, M.; Millam, J. M.; Klene, M.; Adamo, C.; Cammi, R.; Ochterski, J. W.; Martin, R. L.; Morokuma, K.; Farkas, O.; Foresman, J. B.; Fox, D. J. Gaussian 16 Revision C.01. 2016.

[53] Neese, F. The ORCA Program System. *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, **2012**, *2* (1), 73–78.

[54] Giese, T. J.; York, D. M. Ambient-Potential Composite Ewald Method for Ab Initio Quantum Mechanical/Molecular Mechanical Molecular Dynamics Simulation. *J. Chem. Theory Comput.*, **2016**, *12* (6), 2611–2632.

[55] Pan, X.; Nam, K.; Epifanovsky, E.; Simmonett, A. C.; Rosta, E.; Shao, Y. A Simplified Charge Projection Scheme for Long-Range Electrostatics in Ab Initio QM/MM Calculations. *J. Chem. Phys.*, **2021**, *154* (2), 024115.

[56] Dohn, A. O. Multiscale Electrostatic Embedding Simulations for Modeling Structure and Dynamics of Molecules in Solution: A Tutorial Review. *Int. J. Quantum Chem.*, **2020**, *120* (21).

[57] Walker, R. C.; Crowley, M. F.; Case, D. A. The Implementation of a Fast and Accurate QM/MM Potential Method in Amber. *J. Comput. Chem.*, **2008**, *29* (7), 1019–1031.

[58] Darden, T.; York, D.; Pedersen, L. Particle Mesh Ewald: An N Log(N) Method for Ewald

Sums in Large Systems. *J. Chem. Phys.*, **1993**, *98* (12), 10089–10092.

[59] Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. A Smooth Particle Mesh Ewald Method. *J. Chem. Phys.*, **1995**, *103* (19), 8577–8593.

[60] Wu, Y.; Tepper, H. L.; Voth, G. A. Flexible Simple Point-Charge Water Model with Improved Liquid-State Properties. *J. Chem. Phys.*, **2006**, *124* (2), 024503.

[61] Isborn, C. M.; Götz, A. W.; Clark, M. A.; Walker, R. C.; Martínez, T. J. Electronic Absorption Spectra from MM and *Ab Initio* QM/MM Molecular Dynamics: Environmental Effects on the Absorption Spectrum of Photoactive Yellow Protein. *J. Chem. Theory Comput.*, **2012**, *8* (12), 5092–5106.

[62] Hornak, V.; Abel, R.; Okur, A.; Strockbine, B.; Roitberg, A.; Simmerling, C. Comparison of Multiple Amber Force Fields and Development of Improved Protein Backbone Parameters. *Proteins Struct. Funct. Bioinforma.*, **2006**, *65* (3), 712–725.

[63] Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and Testing of a General Amber Force Field. *J. Comput. Chem.*, **2004**, *25* (9), 1157–1174.

[64] Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S.; Windus, T. L.; Dupuis, M.; Montgomery, J. A. General Atomic and Molecular Electronic Structure System. *J. Comput. Chem.*, **1993**, *14* (11), 1347–1363.