TeraChem Protocol Buffers (TCPB): Accelerating QM and QM/MM Simulations with a Client-Server Model

Vinicius Wilian D. Cruzeiro $^{1,2}$, Yuanheng Wang $^{1,2}$, Elisa Pieri $^{1,2}$, Edward G. Hohenstein $^{1,2}$, Todd J. Martínez $^{1,2,*}$

$^1$ Department of Chemistry and The PULSE Institute, Stanford University, Stanford, California 94305, USA
$^2$ SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, California 94025, USA

ABSTRACT

The routine use of electronic structure in many chemical simulation applications calls for efficient and easy ways to access electronic structure programs. We describe how the graphics processing unit (GPU) accelerated electronic structure program TeraChem can be set up as an electronic structure server, to be easily accessed by third-party client programs. We exploit Google’s protocol buffer framework for data serialization and communication. The client interface, called TeraChem protocol buffers (TCPB), has been designed for ease of use and compatibility with multiple programming languages, such as C++, Fortran, and Python. To demonstrate the ease of coupling third-party programs with electronic structure using TCPB, we have incorporated the TCPB client into Amber for quantum mechanics/molecular mechanics (QM/MM) simulations. The TCPB interface saves time with GPU initialization and I/O operations, achieving a speedup of more than 2x compared to a prior file-based implementation for a QM region with ~250 basis functions. We demonstrate the practical application of TCPB by computing the free energy profile of p-HBDI$^-$ – a model chromophore in green fluorescent proteins – on the first excited singlet state using Hamiltonian replica exchange (H-REMD) for enhanced sampling. All calculations in this work have been performed with the freely-available version of TeraChem, which is sufficient for many QM region sizes in common use.
I. INTRODUCTION

Solving the electronic Schrodinger equation for a given molecular geometry (the electronic structure problem) is a mainstay of computational chemistry. It is fundamental in problems such as spectroscopy,\textsuperscript{1-2} reaction discovery\textsuperscript{3-4} and excited state exploration.\textsuperscript{5} Unfortunately, its computational cost is often an obstacle in cases where many calculations are needed, such as molecular dynamics. By exploiting graphical processing units (GPUs), TeraChem\textsuperscript{6-10} accelerates electronic structure calculations by 1-2 orders of magnitude in comparison to commonly used CPU-based implementations.\textsuperscript{11-12} This opens the door to routine study of increasingly complex problems and larger molecular systems.

Methodologies for molecular dynamics (MD), geometry optimizations, transition state searches, exploration of minimum energy pathways between reactants and products (e.g., using nudged elastic band\textsuperscript{13-14}), description of free energy profiles (e.g., using enhanced sampling techniques such as umbrella sampling\textsuperscript{15}), among others, are frequently employed in combination with electronic structure calculations. Such methodologies may not exist in the same software that carries out the electronic structure capabilities; however, they are readily available in third-party software such as Amber,\textsuperscript{16} LAMMPS,\textsuperscript{17} i-PI,\textsuperscript{18} or OpenMM.\textsuperscript{19} This motivates the development of an interface – which should be easy to use and computationally efficient – between the electronic structure program and other applications. Such interface allows electronic structure software developers to focus on electronic structure methodologies while users have an opportunity to effortlessly take advantage of electronic structure features for their own needs – easing the creation of new approaches and applications.
In this publication we leverage how TeraChem can be easily accessed as an external library from third-party software. This is achieved with a client-server model based on Google’s protocol buffers for data serialization and communication. Protocol buffers is a mechanism (compatible with multiple programming languages) that allows efficient data exchange between programs, even over the internet. Therefore, in TeraChem’s client-server model, the server can be located either on the local machine (i.e., the same machine that executes the client) or on a remote machine. The client, called TeraChem protocol buffers (TCPB)\textsuperscript{6, 20} and written in C++, is designed to be straightforward to use. When installed, TCPB generates a shared object library whose functions can be accessed by host code written in different programming languages, such as C++, Fortran, and Python. In this way, a programmer who wants to take advantage of TeraChem features only needs to call TCPB functions and does not need any knowledge about protocol buffers.

In this work, we describe TeraChem’s client-server model and show examples of how programmers can use TCPB in their own custom codes. As a direct application of TCPB, we introduce a new interface between TeraChem and Amber for faster quantum mechanics/molecular mechanics (QM/MM) MD simulations. Furthermore, we present QM/MM benchmarks considering different QM region sizes and, as a practical example of exploiting the power of both TeraChem and Amber within our integration, the free energy profile on the first excited state (S\textsubscript{1}) of the anionic p-hydroxybenzylidene-2,3-dimethylimidazolinone (p-HBDI) in bulk water. Currently, TCPB supports electronic structure calculations with Hartree-Fock (HF),\textsuperscript{7, 9} density functional theory (DFT),\textsuperscript{6, 21} configuration-interaction (CI),\textsuperscript{22} and complete active space (CAS) methods (e.g., CASSCF and CASCI),\textsuperscript{23-26} including excited state calculations.

TCPB is freely available on GitHub and also as part of the recent AmberTools22 release, which also contains the new QM/MM interface presented in this work. The full version of
TeraChem requires a license; however, TeraChem has a freely-available non-commercial version where each calculation can be run for up to 15 minutes using at most 2 GPUs. All the calculations in this publication have been performed with this freely-available version of TeraChem, indicating that this version is sufficient for typical QM/MM applications. For more information on how to obtain TCPB and TeraChem, please refer to the “Data and Software Availability” section.

II. CLIENT-SERVER MODEL

Figure 1 depicts the workflow of TeraChem’s client-server model during a single QM/MM calculation. Both the client and server share the same protocol buffers data structure. Messages are exchanged via standard type-length-value (TLV) packets over a TCP socket, in three different types: input, status, and output. The input message contains the QM atom types and coordinates, electronic structure specifications such as basis set and level of theory, additional TeraChem input options as key-value pairs, and, for QM/MM, the MM coordinates and charges. Status has information about the availability of the server, status of the job (e.g., accepted, rejected, in progress, or completed), and information about the directory where the job is running – which might be useful when the TeraChem server is running on a remote machine. Output contains relevant information such as energy, gradients, atomic charges, dipole moment, and molecular orbital energies/occupations.
Figure 1. Flowchart illustrating how the client-server model works in TeraChem during a single QM/MM calculation. “Input”, “Status”, and “Output” are the types of protocol buffers messages exchanged between the client (in blue) and the server (in green).

The protocol buffers documentation recommends keeping the message transmitted smaller than 1 MB. To comply with this recommendation while allowing users to access larger data elements such as molecular orbital coefficients, the TeraChem server can save all output files generated. These can be retrieved if needed, even when the server is on a remote machine (e.g. using scp) since the output message contains the path to where the server is running. This strategy optimizes data communication compared to passing all possible output data through protocol buffers.

III. USING TCPB FROM YOUR OWN CODE

As mentioned previously, a shared object library – called libtcpb.so – is created when TCPB is compiled. The functions in libtcpb.so can be accessed from multiple programming
languages. For simplicity, in this section we demonstrate with Python, but C++ and Fortran examples are provided in the Supporting Information.

Users can perform full QM or QM/MM calculations with TCPB. When doing QM/MM, two options are available: either consider the MM region as simple point charges without point charge self-interactions or treat the MM region with a classical force field using OpenMM\textsuperscript{19} – which is interfaced with TeraChem. The former option is ideal when interfacing TCPB with software that can handle the description of the MM interactions, while the latter option is useful when one does not want to write an implementation for the MM interactions.

A key advantage of Protocol Buffers is that it standardizes the communication protocol. Because of this, when the TeraChem source code is modified (e.g., implementing a new electronic structure method), there is no need to change TCPB or the host software that calls it. In most cases, users would only need to add or modify keywords in their TeraChem input file.

With the design principle of keeping TCPB as easy-to-use as possible, we have implemented four simple functions that embody the main features necessary to create an interface with TeraChem. These functions are: \texttt{connect}, \texttt{setup}, \texttt{compute\_energy\_gradient}, and \texttt{finalize}. \texttt{Connect} checks if the TeraChem server is available and attempts to access it. \texttt{Setup} reads the TeraChem input file and prepares part of the protocol buffers input message. \texttt{Compute\_energy\_gradient} completes the protocol buffers input message for a given QM region (and MM region, if applicable), sends this message to the TeraChem server, waits until the calculation finishes, receives the protocol buffers output message, and returns the energy and gradients. \texttt{Finalize} clears environment variables. The functions \texttt{connect}, \texttt{setup}, and \texttt{finalize} must be called only once, while \texttt{compute\_energy\_gradient} may be called multiple times (e.g., for
different steps in a MD simulation). **Figure 2** shows an example that performs a single point calculation for a water trimer, where only one water molecule is placed in the QM region.

```python
import sys
import pytcpb as tc

host = "localhost"
port = 12345
tcfile = "terachem.inp"
qmmatypes = ["O", "H", "H"]

status = tc.connect(host, port)
if (status != 0): sys.exit(1)
status = tc.setup(tcfile,qmmatypes)
if (status != 0): sys.exit(1)

qmcoords = 
[[-8.4655945, -5.3668218, 8.0230287],
 [-9.1698954, -7.1146294, 8.3055347],
 [-6.8124622, -5.2095966, 9.3095461]]

mmcoords = 
[[-5.0631428, -4.0810522, 11.1992721],
 [-3.3909243, -4.9021382, 11.3776622],
 [-4.6379545, -2.3143474, 11.1960596],
 [-11.4780067, -1.6653588, 9.8462283],
 [-10.1875128, -2.8372346, 9.0597243],
 [-10.2402362, -0.6124413, 11.2202481]]

mmcharges = 
[[-0.834, 0.417, 0.417, -0.834, 0.417, 0.417],
 [0.834, 0.417, 0.417, -0.834, 0.417, 0.417]]

totenergy, qmgrad, mmgrad, status \n= tc.compute_energy_grad(qmmatypes, qmcoords, mmcoords, mmcharges, globaltreatment = 0)
if (status != 0): sys.exit(1)
tc.finalize()
```

**Figure 2.** Sample Python code for a single point QM/MM calculation using TCPB.

At the top of the code in **Figure 2**, the *PyTCPB* library is imported. *PyTCPB* is a wrapper that uses Python’s CTypes library to convert *libtcpb.so* functions into Python functions. *PyTCPB* is automatically placed in the Python environment as part of the TCPB installation process. Further details of TCPB usage are available in the Supporting Information.

The TeraChem server needs to be active before running TCPB, otherwise the *connect* function will exit with an error indicating that the server was not found. The Supporting Information explains how TeraChem can be executed in server mode.
In the example code shown, the host is set to the same machine running the TCPB client (i.e., localhost); however, it could be pointing to a remote machine by replacing the “host” field with a remote address. The port must match the same one specified when the TeraChem server was activated. The TeraChem input file, set in the tcfile variable, is identical to a regular one but without specifications of QM coordinates or (for QM/MM calculations) MM coordinates and charges.

The TCPB functions have a detailed treatment for error messages since different values returned to the status variable indicate specific problems; for more details, please refer to the Supporting Information. The compute_energy_gradient function expects coordinates in bohr and charges in atomic units.

The globaltreatment flag determines how the initial guess for the wavefunction is constructed. A value of 0 means that the wavefunction from the previous compute_energy_gradient calculation will be used as an initial guess, but only when the number of QM atoms is the same as in the previous compute_energy_gradient call; in case of a change in the number of QM atoms, a completely new wavefunction guess will be constructed from scratch. A globaltreatment value of 1 means that the guess will always be constructed from scratch.

To include MM interactions computed with OpenMM in the energy and forces returned by TCPB, the TeraChem input file (terachem.inp in the Python example above) needs to contain a topology file specification through the prmtop flag and specification about which atoms belong to the QM region through the qmindices flag. When this option is used, the mmcharges variable in the compute_energy_gradient function will be ignored. Usage examples for this option are provided in the Supporting Information.
Electronic structure software commonly produce scratch data. In some cases, scratch data may not be desirable for further analyses but could occupy a considerable amount of storage. This could be particularly concerning for MD simulations, since the storage amount depends linearly on the number of MD steps (assuming that data for each MD step will be independently stored). With this issue in mind, TeraChem has an input flag (genscrdata) that controls whether scratch data will be generated or not.

IV. QM/MM SIMULATIONS IN AMBER

QM/MM is a technique often employed in material design, drug discovery, and photochemistry.\textsuperscript{28-35} Since quantum mechanics (QM) calculations become progressively more expensive for larger systems, QM/MM splits the system into QM and MM regions. The QM region contains the chemically relevant part of the system and is described with electronic structure calculations, whereas the MM region is described with molecular mechanics (MM) approaches that are orders of magnitude faster than QM calculations. With QM/MM one can study situations where MM approaches cannot be applied, such as electron transfer processes, chemical reactions, or photochemistry involving excited states, while circumventing the high computational cost of performing QM calculations for the entire system.

Figure 3 portrays the systems evaluated with QM/MM in this work and discussed in the subsections below. A detailed description of how to perform QM/MM simulations in Amber using TeraChem is provided in the Supporting Information, along with input files for all the calculations reported here.
Figure 3. Systems explored with QM/MM simulations in bulk water (represented with balls-and-sticks). Panel A: Photoactive yellow protein (PYP, in grey cartoon representation) with different QM regions: the smallest QM region tested (QM region 1) contains only the chromophore (highlighted in blue); the other highlighted residues are added to increasingly larger QM regions; Red residues are added in QM region 2, green in QM region 3, and yellow in QM region 4; therefore, QM region 4 contains all residues highlighted in blue, red, green, and yellow. Panel B: p-HBDI, indicating the I- and P-twisting dihedral angles studied with umbrella sampling simulations.

IV.1 Benchmarks

Numerous approaches have been proposed to describe the electrostatic interactions between QM and MM regions in QM/MM.\textsuperscript{32, 36-38} Here, we mention only mechanical embedding (ME) and electrostatic embedding (EE). In ME the QM region is left in vacuum, and all QM/MM electrostatic interactions are fully described at the MM level. In contrast, in EE the QM region is surrounded by point charges at the positions of the MM atoms. In Amber, a hard QM/MM interaction cutoff radius is employed to control how many surrounding point changes will be considered. Because the QM electronic density is directly exposed to its surroundings in the EE representation, one may expect higher accuracy compared to ME, at the expense of an increased computational cost due to the surrounding point charges. The surrounding point charges will affect only the computational cost of the one-electron nuclei repulsion integrals in the QM calculation.
since point charges have effectively the same role as the nuclei in the Hamiltonian of the system. Since TeraChem has a GPU-accelerated implementation for these one-electron integrals\(^\text{10}\) (which are generally computationally less intensive than two-electron integrals), we expect to observe similar computational performance between EE and ME. The two-electron integrals and their computational cost are the same for EE or ME.

**Table 1** details the computational performance of QM/MM simulations for the photoactive yellow protein (PYP) considering different QM region sizes that follow previous work.\(^\text{34}\) **Figure 3A** highlights the differences between the QM regions. QM regions 1 to 4 contain, respectively, 22, 49, 104 and 159 atoms, which are typical QM region sizes for QM/MM studies. The QM calculations in Table 1 have been performed with B3LYP/def2-SVP, which leads to, respectively, 244, 509, 994, and 1404 basis functions for QM regions 1 to 4. A QM/MM interaction cutoff of 8Å was employed (i.e., MM atoms more than 8Å from the nearest QM atom do not affect the QM Hamiltonian). A time step of 0.5 fs is used in molecular dynamics (MD) simulations and mixed precision is used in the GPU-accelerated electronic structure. In **Table 1** we compare the performance of the TCPB interface with the previously existing file-based (FB) interface.\(^\text{37}\) The FB interface reads and writes files to exchange data and makes a new call to the TeraChem executable on each MD step. **Table 1** also compares timings between QM/MM simulations with either EE or ME and using either 1 or 2 GPUs.
Table 1. Computational performance (in ps/day) of QM/MM simulations for the photoactive yellow protein (PYP) in bulk water executed with TeraChem and Amber. The simulations are performed at the B3LYP/def2-SVP level of theory with a timestep of 0.5 fs, mixed precision GPU computations, and a QM/MM cutoff of 8Å. EE = electrostatic embedding, ME = mechanical embedding, TCPB = TeraChem protocol buffers interface, and FB = file-based interface.

<table>
<thead>
<tr>
<th>GPU type</th>
<th>QM/MM type</th>
<th># of GPUs</th>
<th>PYP (QM region 1)</th>
<th>PYP (QM region 2)</th>
<th>PYP (QM region 3)</th>
<th>PYP (QM region 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>TCPB</td>
<td>FB</td>
<td>TCPB</td>
<td>FB</td>
</tr>
<tr>
<td>GTX 970</td>
<td>EE</td>
<td>1</td>
<td>5.17</td>
<td>3.33</td>
<td>2.10</td>
<td>1.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>8.30</td>
<td>3.67</td>
<td>3.38</td>
<td>2.24</td>
</tr>
<tr>
<td>V100</td>
<td>EE</td>
<td>1</td>
<td>5.50</td>
<td>3.46</td>
<td>2.17</td>
<td>1.74</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>8.91</td>
<td>3.80</td>
<td>3.58</td>
<td>2.29</td>
</tr>
<tr>
<td>A100</td>
<td>EE</td>
<td>1</td>
<td>17.28</td>
<td>7.84</td>
<td>7.85</td>
<td>5.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>17.28</td>
<td>7.01</td>
<td>9.40</td>
<td>4.81</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>1</td>
<td>17.29</td>
<td>7.98</td>
<td>8.07</td>
<td>5.26</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>17.29</td>
<td>7.10</td>
<td>9.70</td>
<td>4.87</td>
</tr>
<tr>
<td></td>
<td>ME</td>
<td>1</td>
<td>16.44</td>
<td>7.54</td>
<td>7.74</td>
<td>5.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>17.47</td>
<td>7.07</td>
<td>9.45</td>
<td>4.71</td>
</tr>
</tbody>
</table>

As anticipated, Table 1 shows that EE and ME simulations have similar performance for all QM region sizes (leaving little justification for the use of ME given its decreased accuracy compared to EE). Furthermore, TCPB is significantly faster than the file-based (FB) interface: simulations using TCPB are up to $2.5 \times$ and $1.4 \times$ faster than the FB interface for, respectively, the smallest and largest QM regions examined. The difference in computational efficiency between TCPB and the FB interface is largely due to the GPU initialization, which can take several seconds. In TCPB this stage happens only once – when the server is initialized – while in the FB interface this stage takes place at every timestep in the simulation, since it is done every time the TeraChem executable is called. The percent of the total computation time corresponding to GPU initialization decreases for larger QM regions. Thus, the timings between TCPB and FB interface become more similar as the QM region size increases. Table 1 also points out that TeraChem’s multi-GPU parallelization may further accelerate the calculations. Simulations with two GTX 970 GPUs are $1.6 \times$ and $1.7 \times$ faster than with a single card, respectively, for the smallest and largest QM region.
sizes evaluated. The newer GPU cards, V100 and A100, show a performance speedup in comparison to GTX 970 of up to $3.3 \times$ and $4.3 \times$ on a single GPU, respectively, for the smallest and largest QM region sizes evaluated. However, the multi-GPU parallelization in the newer cards is not satisfactory for the smaller QM regions. Current efforts are ongoing to improve TeraChem’s multi-GPU performance on the newer cards.

Isborn et. al. have previously attempted to create a more efficient interface between TeraChem and Amber using MPI communication. This interface also addressed issues with GPU startup and I/O operations. However, it was specifically tailored for communication between TeraChem and Amber, and difficult to extend to arbitrary interfaces. In contrast, the TCPB interface presented in this work is more general and can be easily interfaced from multiple programming languages, thus easing the access from any program (i.e. with no special use restriction to Amber).

IV.2 Umbrella sampling on the excited state with Hamiltonian replica exchange

Green fluorescent proteins (GFPs), initially identified in the *Aequorea Victoria* bioluminescent jellyfish, have revolutionized molecular life sciences as biomarkers that allow visualization of individual proteins during intracellular processes. GFPs can be placed in different organisms, including bacteria, and glow green upon ultraviolet irradiation. They have been widely employed to advance the understanding of a variety of problems, including viral infection, protein-protein interaction, and protein expression. The chromophore inside GFP is responsible for light absorption and emission, which can be affected significantly by changes in the local environment (i.e., mutations of nearby residues). Therefore, an initial step towards comprehending the chromophore behavior is to study its photochemistry outside of the protein environment.
In this section, we present the free energy profile on the first excited state \((S_1)\) and the \(S_1-S_0\) energy gap for \(p\)-HBDI\(^-\), a model GFP chromophore, in bulk water. We used the imidazolinone-twisting (I-twisting, \(\phi_I\)) dihedral as the collective variable and considered a fixed value of the phenolate-twisting (P-twisting, \(\phi_P\)) dihedral (see Figure 3). The free energy profile on a grid of \((\phi_I, \phi_P)\) values has been recently reported by our research group.\(^{46}\) The present work reproduces a slice of this grid using Amber machinery (simplifying the preparation of input files and analysis of output files). The use of Amber to drive the process enables calculations with Hamiltonian replica exchange MD (H-REMD),\(^{47-48}\) leading to faster convergence due to enhanced sampling. The use of the TCPB interface leads to faster QM/MM calculations. Since our intention here is only to show a practical example of the new TeraChem/Amber interface, we did not compute the free energy profile on the entire two-dimensional \((\phi_I, \phi_P)\) grid.

The \(p\)-HBDI\(^-\) chromophore under periodic boundary conditions in a water buffer of 15 Å (i.e., the distance between the solute and the edge of the simulation box) was initially equilibrated at ambient pressure and temperature using only a force field representation for all interactions. Water molecules were represented with the SPC/Fw force field\(^{49}\) and the \(p\)-HBDI\(^-\) force field was taken from previous work.\(^{46}\) In preparation for the umbrella sampling simulation, we constructed a one-dimensional grid along \(\phi_I\) from \(-120^\circ\) to \(120^\circ\) in intervals of \(9.6^\circ\), giving a total of 26 grid points; the \(\phi_P\) value was kept at \(0^\circ\) in all calculations. Each grid point was equilibrated for 2 ps applying harmonic biasing potentials on each target \((\phi_I, \phi_P)\) pair-value with a force constant of 50 kcal mol\(^{-1}\) rad\(^{-2}\) to both dihedrals. The initial structure for the simulation of each \((\phi_I, \phi_P)\) pair-value was the last structure from a neighboring grid point. All the simulations were performed with a timestep of 0.5 fs.
For the umbrella sampling simulations, we switched to the QM/MM representation where the p-HBDI chromophore was treated at the α(0.67)-SA3-CASSCF(4,3)/6-31G* level of theory targeting the S₁ state. Each grid point (i.e., umbrella window) was simulated for 4 ps, where the first 2 ps were considered as equilibration. We performed one-dimensional H-REMD, with one exchange attempt every 50 fs, by allowing target \( \phi_I \) values to be exchanged. The free energy profile was generated by post-processing the simulation data using WHAM and enforcing the symmetry \( A(\phi_I, \phi_P) = A(-\phi_I, -\phi_P) \), where \( A \) is the free energy. Figure 4 shows the free energy profile on \( S_1 \) and the average \( S_1-S_0 \) energy gap for each grid point. Error bars were computed using bootstrap for the free energy profile and a 95% confidence interval for the \( S_1-S_0 \) energy gap; the error bars are smaller than the linewidths shown in Figure 4. Figure S1 in the Supporting Information shows the distribution of \( \phi_I \) values for each umbrella sampling window; it confirms good sampling of \( \phi_I \) values since each window overlaps well with its neighboring windows.

**Figure 4.** A) Free energy profile, in kcal/mol, on the first excited state (S₁) along the I-twisting (\( \phi_I \)) dihedral while keeping the P-twisting (\( \phi_P \)) dihedral at 0° for the green fluorescent protein model chromophore p-HBDI in bulk water, and B) \( S_0 \) to \( S_1 \) average energy gap, in kcal/mol. Electronic structure calculations performed with α(0.67)-SA3-CASSCF(4,3)/6-31G*. Error bars for the free energy profile are smaller than the linewidths shown.
The free energy profile and average S₁-S₀ energy gap shown in Figure 4 are in close agreement with results previously reported⁴⁶ obtained with the same electronic structure method. The free energy profile helps us to interpret the p-HBDI⁻ photodynamics and allows us to identify intermediates, transition states, and stable configurations on the excited state. The free energy profile displays minima at \((\phi_I, \phi_P) \cong (±90°, 0°)\), showing that I-twisted structures are energetically favored compared to the Franck–Condon point, \((\phi_I, \phi_P) = (0°, 0°)\). Furthermore, the S₁-S₀ gap reveals that I-twisted structures have smaller gaps than at the Franck–Condon point. Indeed, these structures are close to conical intersections, as discussed previously.⁴⁶ ⁵¹-⁵²

V. CONCLUSIONS AND FUTURE PERSPECTIVES

To facilitate the broader use of electronic structure calculations by molecular dynamics, Monte Carlo, and optimization codes, we presented the TCPB interface that eases interfacing third-party codes with TeraChem. With minimal effort, programmers can use TCPB to access TeraChem directly from different programming languages, such as Fortran, C++ or Python, without any need to access or modify TeraChem source code. Communication between TeraChem and third-party software is done through a client-server model that uses Google’s protocol buffers for data exchange.

We used TCPB to build a new TeraChem/Amber interface for QM/MM simulations and showed it outperforms a prior interface that reads and writes files to exchange data and calls the TeraChem executable on each MD step. The new interface saves time with GPU initialization and I/O operations, and can be easily interfaced with different programs. Furthermore, as a demonstration of the features within our TeraChem/Amber integration, we computed the free energy profile and average S₁-S₀ energy gap at the first excited state of p-HBDI⁻ in bulk water, which is in close agreement with previously reported results.⁴⁶ The free energy profile was
obtained through umbrella sampling simulations using Hamiltonian replica exchange (H-REMD). Since QM/MM simulations are generally prohibitively expensive for long time dynamics, enhanced sampling techniques such as H-REMD are critical for better convergence.

The electrostatic embedding (EE) QM/MM approach within our TeraChem/Amber integration currently lacks a description for long-range electrostatic interactions between the QM region and its surroundings. The implementation of long-range effects within periodic boundary conditions in nuclei repulsion one-electron integrals would address this problem. This can be achieved by using methods such as Ewald or particle mesh Ewald adapted to QM/MM. Such implementation would be facilitated by the software infrastructure presented in this work.

TCPB offers a platform for faster TeraChem calculations, and a client-server model that can be used by other software. We hope it will ease the development of new methodologies and approaches that need QM or QM/MM electronic structure calculations.

AUTHOR INFORMATION

* E-mail: toddjmartinez@gmail.com

Notes

TJM is a co-founder of PetaChem, LLC.

DATA AND SOFTWARE AVAILABILITY

Any data generated or analyzed for this publication that is not included on this article or its Supplementary Information is available from the authors upon request.
AmberTools is publicly available free of charge at: https://ambermd.org. TCPB was released as part of AmberTools version of 2022. The latest development version of TCPB can be accessed for free from GitHub: https://github.com/mtzgroup/tcpb-cpp. TeraChem, including its freely-available version, can be obtained from: https://petachem.com.

SUPPORTING INFORMATION


ZIP file: Examples of how to use TCPB from Fortran, C++, and Python, including when MM interactions are computed by OpenMM interfaced with TeraChem. Input files for all QM/MM simulations reported in this work.

AUTHOR DECLARATIONS

TJM is a cofounder of PetaChem, LLC.

ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support from the Office of Naval Research (N00014-18-1-2659 and N00014-21-1-2151). The authors acknowledge Mingning Zhu for a careful reading of the manuscript.
REFERENCES


