JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

Ab Initio Interactive Molecular Dynamics on Graphical Processing Units (GPUs)

Nathan Luehr, Alex G. B. Jin, and Todd J. Martínez*

Department of Chemistry and PULSE Institute, Stanford University, Stanford, California 94305, United States
SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, California 94025, United States

ABSTRACT: A virtual molecular modeling kit is developed based on GPU-enabled interactive *ab initio* molecular dynamics (MD). The code uses the TeraChem and VMD programs with a modified IMD interface. Optimization of the GPU accelerated TeraChem program specifically for small molecular systems is discussed, and a robust multiple time step integrator is employed to accurately integrate strong user-supplied pulling forces. Smooth and responsive visualization techniques are developed to allow interactive manipulation at minimum simulation rates below five MD steps per second. Representative calculations at the Hartree−Fock level of theory are demonstrated for molecular systems containing up to a few dozen atoms.

INTRODUCTION

The steady advance of computers is transforming the basic models used to understand and describe chemistry. In terms of quantitative models, researchers no longer seek human comprehendible, closed-form equations such as the ideal gas law. Instead algorithmic models that are evaluated through intensive computer simulations have become the norm. As a few examples, *ab initio* electronic structure, molecular dynamics (MD), and minimum energy reaction path optimizers are now standard tools for describing chemical systems.

Despite their often-impressive accuracy, quantitative models sometimes provide surprisingly little scientific insight. For example, individual MD trajectories are chaotic and can be just as inscrutable to human understanding as the physical experiment they seek to elucidate. Qualitative cartoon-like models are essential to inspire human imagination and satisfy our quest for understanding. As an illustration, consider the pervasive use of primitive ball-and-stick type molecular models in chemistry. The success of these physical models lies as much in their ability to capture human imagination and support interesting geometric questions as it does in their inherent realism. Useful models should be both accurate and playful.[1]

Fortunately, computers are capable of much more than crunching numbers for quantitative models. With the development of computer graphics and the explosion of immersive gaming technologies, computers also provide a powerful platform for human interaction. Starting with work by Johnson[2] and Levinthal[3] in the 1960s, molecular viewers were developed first to visualize and manipulate X-ray structures and later to visualize the results of MD simulations as molecular movies. The next goal was to allow researchers to interact with realistic physical simulations in real time as they ran, as eloquently foreshadowed by Wilson and co-workers.[4] In pioneering work, Brooks and co-workers built an interactive molecular docking simulator.[1,5] The Sculpt project provided an interactive geometry optimizer that included a user-defined spring force in a modified steepest descent optimizer.[6] By furnishing the molecular potential from a classical force field further simplified with rigid bonds and strict distance cutoffs for nonbonded interactions, Sculpt could achieve real-time calculation rates for protein systems containing up to 80 residues, which was certainly impressive at the time.

Later work replaced Sculpt's geometry optimizer with an MD kernel.[7,8] Rather than being limited to minimum energy structures, the user could then probe the dynamical behavior of protein systems, watching the dynamics trajectory unfold in real time and inserting arbitrary spring forces to steer the dynamics in any direction of interest. Force-feedback devices have also been used to control molecular tugs.[9] These allow users to feel as well as see the molecular interactions and increase the precision of user control. Of course, arbitrary user interaction is a (sometimes large) source of energy that flows into the simulation. Aggressive thermostats are necessary to reduce this heating.[7] As a consequence, the results of interactive dynamics are not immediately applicable, for example, in calculating statistical properties. However, for small forces, the trajectories will explore phase space regions that are still relevant to dynamics in standard ensembles and thus offer many qualitative mechanistic insights.

Interactive classical dynamics remains an active field.[10−12] Modern techniques allow interactive simulations on systems containing more than a million atoms.[13] However, classical force fields suffer from two disadvantages that hamper their application to general-purpose chemical modeling. First, they are empirically tuned and as a result are valid only in a finite region of configuration space. This is particularly problematic for interactive simulations, where the user is free to pull the system into highly distorted configurations. The second, more important disadvantage is that covalent bonds modeled by

Received: May 7, 2015
Published: September 2, 2015

DOI: 10.1021/acs.jctc.5b00419
J. Chem. Theory Comput. 2015, 11, 4536−4544

classical springs cannot rearrange during the simulation. *Ab initio* forces, calculated from first-principles electronic structure theory, do not suffer from these disadvantages and provide ideal potentials for use with interactive dynamics. However, due to prohibitive computational costs, it remains difficult to evaluate *ab initio* forces at the fast rate necessary to support real-time dynamics.

Recently, a divide-and-conquer implementation of the semiempirical atom superposition and electron delocalization molecular orbital (ASED-MO) model has been developed to support real-time energy minimization and MD in the SAMSON program.[14−16] SAMSON offers impressive performance—it is able to carry out real-time calculations on systems containing up to thousands of atoms. User interaction is implemented by alternating between "user action steps" in which the user moves or inserts atoms and standard optimization or MD steps. In order to keep the computational complexity manageable, SAMSON applies several approximations beyond those inherent in ASED-MO. The global system is split into overlapping subproblems that are solved independently in parallel. Also for large systems, distant atoms are frozen so that only forces for atoms in the region of user interaction need to be calculated in each step.

Several previous attempts at full *ab initio* interactive dynamics have been reported. In early work, Marti and Reiher avoided on the fly electronic structure calculations by using an interpolated potential surface.[17] The potential surface is precalculated over some relevant configuration space. Then, during dynamics, the force at each time step is obtained from a simple moving least-squares calculation. However, it is difficult to predict *a priori* what regions of configuration space will be visited. A partial solution is to periodically add additional interpolation points when and where higher accuracy is desired.[18] However, the number of needed interpolation points grows exponentially with the dimensionality of the system. Thus, for nontrivial systems, it is essential to evaluate the *ab initio* gradient on the fly. Recently, the feasibility of such calculations has been tested using standard packages and tools. Combining the Turbomole DFT package, minimal basis sets, effective core potentials, and a quad core processor, Haag and Reiher achieve update rates on the order of a second for systems containing up to eight atoms.[19]

In the following, we present the results of our own implementation of interactive *ab initio* dynamics. By using the GPU accelerated TeraChem[20] code and carefully streamlining the calculation, interactive simulations are possible for systems up to a few dozen atoms. The final result is a virtual molecular modeling kit that combines intuitive human interaction with the accuracy and generality of an *ab initio* quantum mechanical description.

METHOD

The *ab initio* interactive MD (AI-IMD) system described below is based on an interactive MD (IMD) interface that was previously developed to enable interactive steered molecular dynamics in the context of classical force fields.[9] A high level overview of the original scheme is shown in Figure 1. Molecular visualization and management of the haptic (or "touch") interface is handled by VMD.[21] Along with the current molecular geometry, VMD displays a pointer that the user controls through a 3D haptic device (shown schematically in Figure 1). Using the pointer, the user can select and tug an atom feeling the generated force through feedback to the haptic

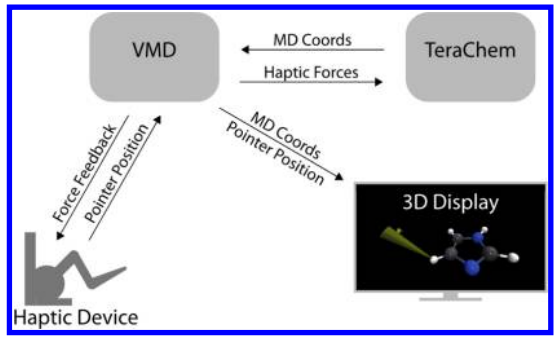

Figure 1. Schematic representation of the IMD interface previously developed for classical MD calculations. VMD is responsible for visualization while TeraChem performs AIMD calculations in real time.

device. VMD also sends the haptic forces to a separate MD program, in our case TeraChem,[20] where they are included with the usual *ab initio* gradient in the following haptic-augmented force:

$$\mathbf{F}(\mathbf{R}, t) = -\nabla E^{qm}(\mathbf{R}) + \mathbf{F}^{hap}(t) \quad (1)$$

where E^{qm} is the *ab initio* potential energy surface and $\mathbf{F}^{hap}$ is the user-generated haptic force. After integrating the system forward in time, TeraChem returns updated coordinates for display by VMD.

A major advantage of the IMD scheme is that it uses spring forces to cushion the user's interaction with the system, rather than raw position updates. In other words, the haptic pointer location represents the equilibrium position of a harmonic spring that is attached to the desired atom. The force constant for this spring is user-tunable to adjust the simulation's sensitivity to haptic perturbations. This scheme allows the user to add weak biases that do not totally disrupt the initial momentum of the system. It also avoids severe discontinuities that would overthrow the numerical stability of standard MD integrators. As a result, the system's dynamics and energy are always well-defined, albeit for a time-dependent Hamiltonian, and the magnitude of haptic perturbations can in principle be accurately measured and controlled.

Communication across the IMD interface is asynchronous. In VMD, the render loop does not wait for updated coordinates between draws. Force updates are sent to TeraChem continuously as they happen rather than waiting for the next MD time step. Similarly, TeraChem uses a separate communication thread to store haptic messages in an internal buffer which is protected by a mutex. If the buffer is empty, no haptic forces are applied. Thus, the MD loop never pauses to wait for haptic force updates between time steps. This scheme was designed to minimize communication latencies.[9] Asynchronous communication also logically decouples the software components and allows each to operate in terms of generic streams of coordinates and simplifies the process of adapting the system from classical to *ab initio* MD as detailed below.

Due to the considerations above, the IMD approach provides a robust starting point for AI-IMD. However, several adjustments to the classical IMD approach are needed to accommodate *ab initio* calculations. These are detailed in the following sections.

Simulation Rate. A primary benefit of interactive modeling is that molecular motions, as well as static structures, can be intuitively represented and manipulated. Thus, it is critical to

DOI: 10.1021/acs.jctc.5b00419
J. Chem. Theory Comput. 2015, 11, 4536−4544

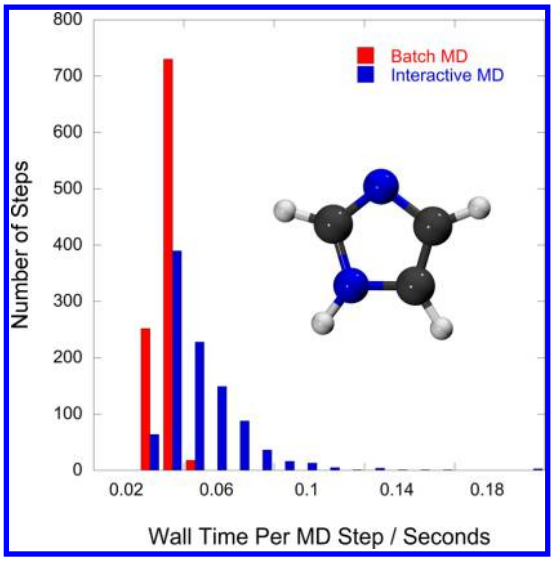

Figure 2. Histogram of wall times for 1000 steps of MD run with user interaction (blue) and without haptic input (red) for the imidazole molecule (inset). The simulation used the unrestricted Hartree−Fock electronic structure method with the 3-21G basis set. Each step of the SCF was converged to $2.0e^{-5}$ atomic units (convergence criteria was the maximum element of the commutator **SDF−FDS**).

maintain a sensation of smooth motion. To achieve this, past research has targeted simulation rates of at least 10 or 20 MD steps per second.[7,8,15] Such update rates are comparable to video frame rates and certainly result in smooth motion. At present, however, quantum chemistry calculations requiring less than 50 ms are possible only for trivially small systems. In order to reach larger molecules and basis sets, it is important to decouple graphical updates from the underlying MD time steps. Ultimately, the necessary simulation rate is dictated not by graphics considerations but by the time scale of the motion being studied. The goal of interactive MD is to shift the movement of atoms to the time scale of seconds on which humans naturally perceive motion. For molecular vibrations, this requires simulating a few femtoseconds of molecular trajectory per second of wall time. Assuming a 1 fs time step, each gradient evaluation is then allowed at least 200 ms to execute. Our experience suggests that up to a full second between *ab initio* gradient evaluations is tolerable, though the resulting dynamics become increasingly sluggish.

In addition to high performance, an interactive interface requires a uniform simulation rate. Each second of displayed trajectory must correspond to a standard interval of simulated time. This is critical both to convey a visual sensation of smooth motion as well as to provide consistent haptic physics. For example, as the simulation rate increases, a haptic input exerted over a fixed interval of wall time will perform an increasing amount of work on the system, since the same duration of pull in wall time translates into longer pulls in simulated time.[9] Problematically, the effort needed to evaluate the *ab initio* gradient varies widely depending on the molecular coordinates. For many geometries, such as those near equilibrium, the SCF equations can be converged in just a few iterations by using guess orbitals from previous MD steps. For strongly distorted geometries, however, hundreds of SCF iterations are sometimes required. Even worse, the SCF calculation may diverge, causing the entire calculation to abort. Since users tend to drive the system away from equilibrium, difficult to converge geometries are more common in interactive MD than in dynamics run on traditional ensembles, as demonstrated in Figure 2. To handle these distorted geometries, we employ the very robust ADIIS +DIIS convergence accelerator.[22,23] However, for well-behaved geometries, ADIIS+DIIS was found on average to require more iterations than DIIS alone. Thus, the best approach is to converge with standard DIIS for up to ∼20 iterations and switch over to ADIIS+DIIS only where DIIS fails.

In order to establish a fixed simulation rate, we consider the variance in timings for individual MD steps illustrated for a particular system in Figure 2. Noting that the large variance in MD step timings is primarily the result of a few outliers, the target wall time for a simulation step, T_{wall}, can be set far below the worst-case gradient evaluation time. For the vast majority of steps, the *ab initio* gradient then completes before T_{wall} has elapsed, and the MD integrator must pause briefly to allow the display to keep pace. For pathologically long MD steps, such as those requiring many ADIIS iterations, the displayed trajectory is frozen and haptic input ignored until the MD step completes.

Integrating the Haptic Force. A key strength of AI-IMD is that it allows users to make and break bonds between atoms. This level of control requires haptic forces that are stronger than the bonding interactions between atoms. The situation in classical IMD is very different. Classical force fields in general cannot handle bond reorganization. Thus, haptic inputs in classical IMD need only overcome weak, nonbonded interactions. When *ab initio* IMD employs much stronger haptic forces in eq 1, noticeable energetic artifacts are produced in the MD simulation. For example, an isolated atom held strongly by a fixed haptic pointer will visibly gain amplitude with each oscillation. In complex molecular systems, such nonconservative dynamics result in rapid heating of the system that quickly swamps any user control. The cause is simply that the MD time step appropriate for a closed system is too long to accurately handle haptic forces that are stronger than the interatomic interactions. An obvious solution is to use a shorter time step. However, reducing the time step would adversely slow the simulation rate, severely reducing the size of systems that can be modeled interactively. Also, as the system became more sluggish, we found ourselves compensating by increasing the haptic force, exacerbating the problem.

A more elegant solution is to use a multiple time step integrator, such as reversible RESPA,[24] to separate the haptic forces from the *ab initio* interactions. The weaker *ab initio* forces can then use a longer MD time step and be evaluated less frequently (in wall time) than the stronger haptic forces. Between each *ab initio* update, l substeps are used to accurately integrate the haptic force as follows:

DOI: 10.1021/acs.jctc.5b00419
J. Chem. Theory Comput. 2015, 11, 4536−4544

$$
\begin{aligned}
&n \leftarrow n+1;\ m \leftarrow 0\\
&\mathbf{V}_i^{(n+1/2,0)} \leftarrow \mathbf{V}_i^{(n,l)} - \frac{\Delta t}{2}\frac{\nabla_i E^{qm}(\mathbf{X}^{(n,0)})}{M_i}\\
&l \times \begin{cases}
\mathbf{V}_i^{(n+1/2,m+1/2)} \leftarrow \mathbf{V}_i^{(n+1/2,m)} + \dfrac{\mathbf{F}_i^{\mathrm{hap}}(t_{n,m})}{2M_i}\delta t\\
\mathbf{X}_i^{(n,m+1)} \leftarrow \mathbf{X}_i^{(n,m)} + \mathbf{V}_i^{(n+1/2,m+1/2)}\delta t\\
\mathbf{V}_i^{(n+1/2,m+1)} \leftarrow \mathbf{V}_i^{(n+1/2,m+1/2)} + \dfrac{\mathbf{F}_i^{\mathrm{hap}}(t_{n,m+1})}{2M_i}\delta t\\
m \leftarrow m+1
\end{cases}\\
&\mathbf{X}_i^{(n+1,0)} \leftarrow \mathbf{X}_i^{(n,l)}\\
&\mathbf{V}_i^{(n+1,0)} \leftarrow \mathbf{V}_i^{(n+1/2,l)} - \frac{\Delta t}{2}\frac{\nabla_i E^{qm}(\mathbf{X}^{(n+1,0)})}{M_i}
\end{aligned}
\tag{2}
$$

where $\mathbf{X}_i$, $\mathbf{V}_i$, $\mathbf{A}_i$, and M_i are the position, velocity, acceleration, and mass for the ith degree of freedom, respectively. The outer time step Δt represents the MD time step between *ab initio* force updates while the inner time step, $\delta t = \Delta t/l$, governs the interval between haptic force evaluations. The superscript (n,m) notation is used to denote quantities where the outer/inner time step indices are n/m, respectively. As in eq 1, the *ab initio* acceleration is given in terms of the gradient of the quantum energy, while $\mathbf{F}^{\mathrm{hap}}(t_{n,m})$ is the haptic force vector evaluated at time $t_{n,m} = n\Delta t + m\delta t$. Compared to the computational costs of an electronic structure calculation, it is trivial to integrate the haptic force in each substep. Thus, the MTS scheme runs at the full speed of the simpler velocity Verlet algorithm.

A second difficulty arises when incorporating strong haptic forces in MD. As shown in Figure 1 above, in the classical IMD scheme, VMD is responsible for calculating the haptic force based on the currently displayed positions of the targeted atom and haptic pointer. However, due to communication latencies and the time required for VMD to complete each graphical update, the forces received by TeraChem lag the simulated trajectory by at least 7 ms in our configuration. In part, this lag is due to network communication since the computer which handles the haptic input and display is separate from the one that computes the *ab initio* forces. Although this lag might be decreased by hosting the haptic input, display output, and *ab initio* force evaluations on the same machine, it cannot be completely removed and thus should be considered in a robust implementation. Importantly, this lag can shift the forces to be in resonance with the system's nuclear motion. This results in uncontrolled heating of the haptic vibrational mode. The solution is to modify the IMD scheme so that haptic positions, rather than pulling forces, are sent to TeraChem. TeraChem can then always calculate the haptic forces at the correct instantaneous molecular geometry.

Decoupling Display from Simulation. Having developed a robust MD integrator for interactive simulations, we now consider how to best display molecular motion to the user. In order to maintain the visual sensation of smooth motion, the displayed coordinates must be updated with a minimum frequency of ∼20 Hz. Assuming, as above, that it requires several hundred milliseconds of wall time to evaluate each MD step, multiple visual frames must be generated for each simulated MD step. To accomplish this, we distinguish between the simulated system which consists of the usual coordinates, velocities, and accelerations at each time step, i, and a separate display system, $\tilde{\mathbf{X}}(t)$, which is continuous in time and closely follows the simulated system, $\tilde{\mathbf{X}}(i\Delta t) \approx \mathbf{X}_i$.

By separating the simulation and display problems, the numerical integrity of the overall model can be maintained while maximizing interactivity. Robust MD integrators guarantee the numerical stability of the simulation and provide well-defined physical properties, such as potential and kinetic energies, at each MD time step. At the same time, the displayed system is free to compromise accuracy for additional responsiveness. This is advantageous because the tolerances of human perception are much more forgiving than those required for stable numerical integration.

Consider a simplified case in which the simulated trajectory is integrated using the velocity Verlet integrator.[25] In each MD step, the simulated system is propagated forward in time by Δt as follows:

$$
\begin{aligned}
&\text{Step 1: } \mathbf{X}^{(n+1)} = \mathbf{X}^{(n)} + \mathbf{V}^{(n)}\Delta t + \frac{1}{2}\mathbf{A}^{(n)}\Delta t^2\\
&\text{Step 2: } \mathbf{A}_i^{(n+1)} = \frac{-\nabla_i E^{ai}(\mathbf{X}^{(n+1)}) + \mathbf{F}_i^{\mathrm{hap}}((n+1)\Delta t)}{M_i}\\
&\text{Step 3: } \mathbf{V}^{(n+1)} = \mathbf{V}^{(n)} + \frac{\mathbf{A}^{(n)} + \mathbf{A}^{(n+1)}}{2}\Delta t
\end{aligned}
\tag{3}
$$

Here, the evaluation of the *ab initio* gradient dominates the wall time required to compute each MD step, T_{wall}. Latency between haptic inputs and the system's response would be minimized by immediately displaying each coordinate vector, $\mathbf{X}^{(n+1)}$, as it is calculated in step 1. However, in this case displaying further motion during the time-consuming calculation of $\mathbf{A}^{(n+1)}$ in step 2 would require extrapolation forward in time. In practice, such extrapolation leads to noticeable artifacts as the simulated and displayed coordinates diverge. An alternative is to buffer the displayed trajectory by one MD step. Thus, $\mathbf{X}^{(n)}$ is displayed as $\mathbf{X}^{(n+1)}$ is calculated in step 1, and the display can then interpolate toward a known $\mathbf{X}^{(n+1)}$ during the succeeding gradient evaluation. In this way, smooth motion is achieved. The distinction between simulated and visualized systems is illustrated in Figure 3.

A variety of interpolation schemes are possible for $\tilde{\mathbf{X}}$. For example, linear interpolation would give the following trajectory:

$$
\tilde{X}(nT_{\mathrm{wall}} + t) = \mathbf{X}^{(n)} + \frac{\mathbf{X}^{(n+1)} - \mathbf{X}^{(n)}}{T_{\mathrm{wall}}}t \tag{4}
$$

where t refers to wall time since the previous (nth) MD step. While eq 4 provides continuous molecular coordinates, the visualized velocity jumps discontinuously by

$$
\Delta\tilde{V} = \mathbf{A}^{(n+1)}\frac{\Delta t^2}{T_{\mathrm{wall}}} \tag{5}
$$

after each MD step. These velocity jumps can be reduced to

$$
\Delta\tilde{V} = \frac{\mathbf{A}^{(n)} - \mathbf{A}^{(n-1)}}{2}\frac{\Delta t^2}{T_{\mathrm{wall}}} \tag{6}
$$

DOI: 10.1021/acs.jctc.5b00419
J. Chem. Theory Comput. 2015, 11, 4536−4544

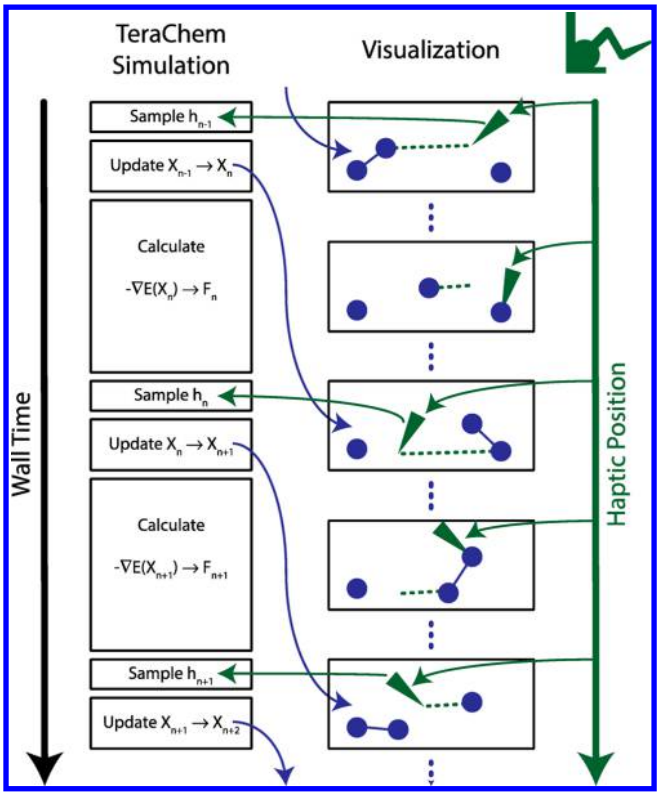

Figure 3. Schematic of simulated and visualized systems. Visualized frames (right boxes) lag the simulated coordinates (left boxes) by one *ab initio* gradient evaluation. The visualized coordinates are interpolated between MD time steps, but the simulated and visualized coordinates match after each step. The visualized haptic position is read from the device at each visual frame but is sampled into h_n by the simulation once per MD step. The l MTS substeps of eq 2 calculate the haptic forces from a common haptic position.

by continuously accelerating the display coordinates over the interpolated path as follows:

$$\tilde{X}(nT_{\text{wall}} + t) = \mathbf{X}^{(n)} + \mathbf{V}^{(n)}\frac{\Delta t}{T_{\text{wall}}}t + \frac{1}{2}\mathbf{A}^{(n)}\frac{\Delta t^2}{T_{\text{wall}}^2}t^2 \tag{7}$$

This trajectory is again continuous in coordinate space, and additionally, in the special case of a constant acceleration, the velocity is also continuous between MD steps. Since, for MD simulations, molecular forces change gradually from step to step, eq 7 is sufficient to reduce velocity discontinuities below the threshold of human perception.

The simple approach of eq 7 can be extended to the more complicated MTS integrator developed above. For example, the displayed trajectory can be defined piecewise for some time t between inner time steps (n,m) and $(n,m+1)$ as follows.

$$\tilde{X}((n + m/l)T_{\text{wall}} + t) = \mathbf{X}^{(n,m)} + \mathbf{V}^{(n,m)}\frac{\Delta t}{T_{\text{wall}}}t + \tilde{A}^{(n,m)}t^2 \tag{8}$$

Here, the value of the apparent acceleration, $\tilde{\mathbf{A}}$, must be considered carefully. For example, it is tempting to define the acceleration as in eq 3:

$$\tilde{A}_i^{(n,m)} = \frac{-\nabla_i E^{ai}(\mathbf{X}^{(n,0)}) + \mathbf{F}_i^{\text{hap}}((n + m/l)T_{\text{wall}})}{\mathbf{M}_i}\frac{\Delta t^2}{T_{\text{wall}}^2} \tag{9}$$

However, this would cause the displayed and simulated coordinates to diverge from one another, since RESPA applies the *ab initio* acceleration linearly rather than quadratically during the inner time steps. The situation is improved by dropping the *ab initio* gradient from $\tilde{\mathbf{A}}$. However, this corresponds to a linear interpolation in the *ab initio* forces similar to eq 4. Our final approach defines an apparent velocity, $\tilde{\mathbf{V}}$, in order to smooth the *ab initio* acceleration over the time step, while applying each haptic force only over its own substep as follows, where $\tilde{\mathbf{A}}$ is defined in terms of the simulated forces according to eq 9:

$$\tilde{V}^{(n,m)} = \mathbf{V}^{(n,0)}\frac{\Delta t}{T_{\text{wall}}} + \frac{\Delta t}{l}\sum_{p<m}\tilde{A}^{(n,p)}$$

$$\tilde{X}((n + m/l)T_{\text{wall}} + t) = \mathbf{X}^{(n,0)} + \frac{\Delta t}{l}\sum_{p<m}\left(\tilde{V}^{(n,p)} + \frac{\Delta t}{2l}\tilde{A}^{(n,p)}\right) + \tilde{V}^{(n,p)}t + \frac{t^2}{2}\tilde{A}^{(n,p)} \tag{10}$$

As a result of this smoothing, the simulated and displayed systems do not match at each substep. However, by applying the same haptic forces calculated for the simulated system to the visual system, the systems will match to within machine precision at the outer time steps. To avoid any buildup of round-off error, the display coordinates are resynced to the simulated system after each step.

In the present implementation, TeraChem is responsible for calculating both the simulated and displayed trajectories. At intervals of about 10 ms, a communication thread evaluates $\tilde{\mathbf{X}}(t)$ and sends the updated coordinates to VMD.

Optimizing TeraChem for Small Systems. TeraChem was originally designed to handle large systems such as proteins.[26,27] To enable these calculations, the ERIs contributing to the Coulomb and exchange operators are computed using GPUs. To maximize performance, TeraChem uses custom unrolled procedures for each angular momentum class of ERIs (e.g., one function handles Coulomb (sslss) type contributions, another (sslsp), and so on).[28] For large systems, near perfect load balancing is achieved by distributing each type of ERI across all available GPUs. A schematic depiction of this strategy is shown in the top frame of Figure 4. However, for small systems, there is not enough parallel work in an ERI batch to saturate even one, much less as many as eight GPUs. Thus, a new strategy was implemented as illustrated in the lower frame of Figure 4.

Here, each ERI class runs on a single GPU, and different GPUs run different types of ERI kernels in parallel. The throughput of each GPU was further improved by using CUDA streams to enable simultaneous execution of multiple kernels on each GPU. Dynamic load balancing was enabled through Intel Cilk Plus work-stealing queues. Together, these schemes significantly increase the parallelism exposed by a small calculation.

When handling small systems, it is also essential to minimize execution latencies that may not have been apparent in larger calculations. In GPU code, communication across the PCIe bus is a common cause of latency. This was minimized by using asynchronous CUDA streams which are provided as part of the CUDA runtime. Asynchronous CUDA streams reduce latencies for host-GPU communication and enable memory transfers to

DOI: 10.1021/acs.jctc.5b00419
J. Chem. Theory Comput. 2015, 11, 4536−4544

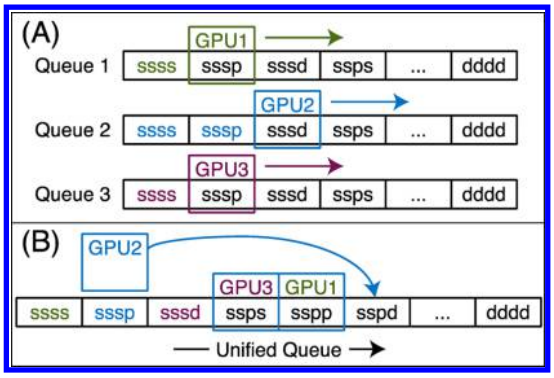

Figure 4. Multi-GPU parallelization strategies. (A) Static load balancing ideal for large systems, where each class of integrals (e.g., sssp) provides enough work to saturate all GPUs. (B) Dynamic scheduling in which each GPU computes independent integral classes, allowing small jobs to saturate multiple GPUs with work.

overlap kernel execution. Another important cause of latency is the allocation and release of GPU memory. Memory allocation is an expensive operation on most architectures, but these allocations are particularly costly on the GPU because they trigger synchronization across devices and even serialization of execution for kernels run on separate GPUs. To eliminate these costs, large blocks of memory are preallocated from each GPU at the start of the calculation. Individual CPU threads request these preallocated blocks as needed. A mutual exclusion lock is used to guarantee thread safety during the assignment of GPU memory blocks.[29] Once assigned, the memory block is the sole property of a single thread and can be used without further synchronization. Extending this system, blocks of page-locked host memory are also preallocated and distributed along with the GPU memory blocks. By assembling GPU input data in page-locked host memory, the driver can avoid staging transfers through an internal buffer, further improving latency.

As shown in Table 1, the new optimizations accelerate single GPU calculations on small molecules significantly (e.g., imidazole with nine atoms), while they have less effect on larger molecules (e.g., taxol and olestra with 110 and 453 atoms, respectively). Furthermore, the parallel performance is significantly improved for both small and medium-sized molecules, but again the performance for large molecules is unaffected. These results are expected, since the improvements we detail above are focused on cases where there is insufficient computational work to saturate the GPUs and/or to hide latencies due to PCIe and/or memory allocation. We found that both of the major improvements (GPU load balancing of small kernels and avoidance of memory allocation) contributed roughly equally to the observed performance enhancement for small and medium (up to ∼100 atoms) molecules.

Assessment. The algorithms detailed above have been implemented in a system using two separate computers. The first computer is responsible for the haptic input and visual display, leveraging the interactive molecular dynamics interface in VMD using the Windows 7 operating system. VRPN server[30] was used to provide the drivers for the Sensable Phantom Omni haptic controller.[31] The second computer is responsible for *ab initio* force evaluations with TeraChem. This computer used the Linux operating system (CentOS 6.5) and eight NVIDIA GTX 970 graphics cards with dual quad-core Xeon E5−2643 CPUs running at 3.3 GHz. Each GPU was associated with a single core of the CPU. Communication between the

Table 1. Result of Optimizing TeraChem (Using RHF/6-31G*) for a Few Representative Molecules, Spanning a Range of Sizes from 9 to 453 Atoms[a]

	imidazole (9 atoms)			
	unoptimized		optimized	
GPUs	seconds	SpdUp	seconds	SpdUp
1	1.98	1.0	0.82	1.0
2	2.09	0.9	0.44	1.9
3	2.32	0.9	0.31	2.6
4	2.63	0.8	0.35	2.4
5	3.05	0.6	0.22	3.7
6	3.36	0.6	0.20	4.1
7	3.79	0.5	0.18	4.6
8	4.57	0.4	0.17	4.8
	taxol (110 atoms)			
	unoptimized		optimized	
GPUs	seconds	SpdUp	seconds	SpdUp
1	87.58	1.0	71.30	1.0
2	52.47	1.7	39.20	1.8
3	41.25	2.1	28.10	2.5
4	36.82	2.4	23.28	3.1
5	35.44	2.5	20.96	3.4
6	34.40	2.5	18.72	3.8
7	34.28	2.6	17.42	4.1
8	34.78	2.5	16.68	4.3
	olestra (453 atoms)			
	unoptimized		optimized	
GPUs	seconds	SpdUp	seconds	SpdUp
1	511.08	1.0	480.17	1.0
2	283.54	1.8	269.04	1.8
3	208.98	2.4	193.87	2.5
4	173.09	3.0	161.41	3.0
5	159.73	3.2	146.15	3.3
6	147.99	3.5	133.01	3.6
7	142.27	3.6	125.35	3.8
8	139.03	3.7	119.24	4.0

[a]Times are total wall times per MD step averaged over five 1 fs time steps. The initial guess was taken from the previous step, and the wave function was converged to $1.0e^{-5}$ Hartree in the maximum element of the commutator, **SPF−FPS**, where **S**, **P**, and **F** are the overlap, one-particle density, and Fock matrices, respectively.

two computers was carried out over a standard Ethernet network. User-applied forces from the haptic device are mapped linearly into a simulation force with a user-adjustable scaling constant (currently determined by trial and error). Other mappings are possible, and nonlinear mappings could be useful in allowing the user to detect both weaker electrostatic forces as well as the stronger covalent forces for which the current system was primarily designed. However, such mappings were not explored in the current work.

In order to benchmark the numerical stability of the AI-IMD integrator, we first consider the trivial diatomic system, HCl. Because the user does work through the haptic forces, interactive dynamics will not in general conserve the total energy of the system. Energy is conserved, however, in the special case of a fixed pulling position since the spring force can be construed as a function of only the molecular geometry, and the Hamiltonian becomes time independent. Indeed, in this case, the resulting dynamics is closely related to that on a force-modified potential energy surface.[32] Using this special case, the

DOI: 10.1021/acs.jctc.5b00419
J. Chem. Theory Comput. 2015, 11, 4536−4544

AI-IMD integrator is validated as follows. First, the HCl molecule is minimized at the RHF/6-31G** level of theory. The resulting geometry is aligned along the y axis with the chlorine atom located at the origin and the hydrogen at y = 1.265 Å. The test system also includes a haptic spring connecting the hydrogen atom to a fixed pulling point at y = 1.35 Å.

Starting from the above initial geometry, AI-IMD calculations were run using varying haptic spring constants and MTS substeps. All calculations used a fixed time step of 1.0 fs to integrate the *ab initio* forces. For smaller haptic force constants, below 0.7 hartree/Bohr,[2] the haptic-induced motions occur on time scales comparable to *ab initio* forces encountered in near-equilibrium MD. Thus, a simple velocity Verlet integrator conserves energy about as well as the MTS approach developed above. At larger forces, however, velocity Verlet becomes increasingly unstable. Figure 5 demonstrates the stability of our

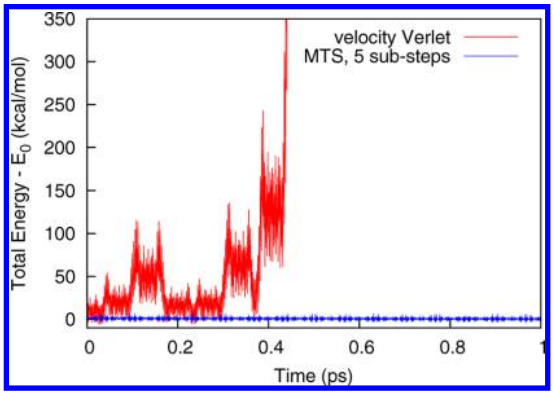

Figure 5. Total energy for IMD simulation of HCl with fixed tugging point. HCl was aligned with y axis, with Cl initially at the origin and H at y = 1.265 Å. The haptic force pulled toward y = 1.35 Å with a force constant of 2.6 hartree/Bohr.[2] A standard velocity Verlet integrator (red) suffers from wild energy oscillations and diverges before reaching 500 fs. The AI-IMD MTS integrator (blue) used 5 substeps per *ab initio* force evaluation and shows no instability throughout the entire 5 ps trajectory (of which 1 ps is shown).

AI-IMD integrator for a spring constant of 2.6 hartree/Bohr.[2] Here, the velocity Verlet integrator rapidly diverges, while an MTS integrator using five substeps remains stable. Although overall energy drifts increase at higher forces, the MTS approach does not exhibit wild divergence even at a force constant of 4.0 hartree/Bohr.[2] This is important because explosive divergence is a much greater obstacle to controlling the system than is a slow energy drift. In general, the motion of the haptic tugging point will itself induce heating so that a thermostat is already required to counter the slow accumulation of energy during long simulations.

We turn next to more interesting systems that better illustrate the potential of AI-IMD. As currently implemented in TeraChem, AI-IMD can be applied to systems containing up to a few dozen atoms using double-ζ basis sets at the SCF level of theory. For smaller systems, polarization functions can also be employed. AI-IMD is thus well suited to treat reactions of small organic molecules. Table 2 shows the average wall time per MD step for simulations on a variety of representative molecules. Here, the average is calculated from short 250 step non-interactive trajectories using the spin-restricted Hartree−Fock method and various basis sets. The initial geometries were optimized at the Hartree−Fock/3-21G level of theory and are

Table 2. Wall Time (seconds) per MD Time Step[a]

molecule (# atoms)	STO-3G	3-21G	6-31G**
HCl (2)	0.00316	0.00384	0.03952
$CH_3F + OH^-$ (7)	0.01160	0.01540	0.09224
imidazole (9)	0.02164	0.03600	0.15728
caffeine (24)	0.10160	0.19556	0.71788
quinoline (30)	0.11272	0.20168	0.85884
spiropyran (40)	0.24688	0.40184	2.49220

[a]Times are averaged over 250 steps of AIMD at the RHF level of theory. Initial geometries are shown in Figure 6. The MD time step was 1.0 fs. Simulations used eight Nvidia GTX970 GPUs in parallel.

shown in Figure 6. At each MD step, the wave function was converged to $1.0e^{-4}$ Hartree in the maximum element of the

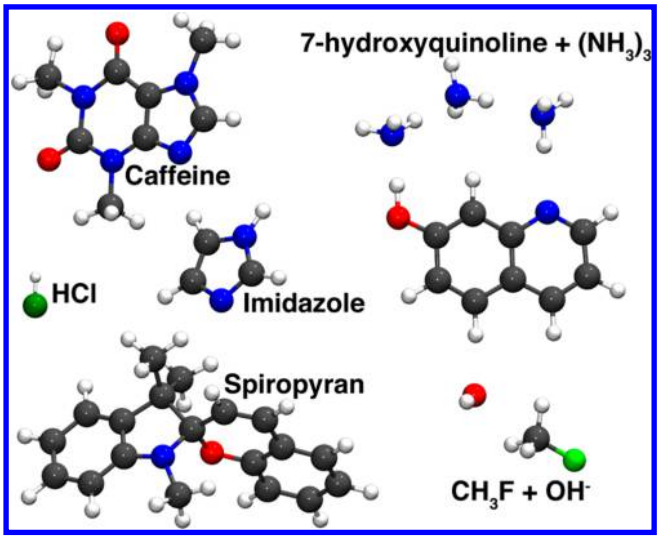

Figure 6. Initial geometries for benchmark calculations listed in Table 2. All structures represent minima at the RHF/3-21G level of theory.

commutator, **SPF−FPS**, where **S**, **P**, and **F** are the overlap, one-particle density, and Fock matrices, respectively. The initial density for each SCF is obtained from the converged density of the previous MD step. The Coulomb and exchange operators are computed predominantly using single precision floating-point operations, with double precision used to accumulate each contribution into the Fock matrix. This scheme has been shown to be accurate for systems up to 100 atoms.[33] Notably, the *ab initio* performance achieved here for systems containing up to 30 atoms with minimal STO-3G and 3-21G basis sets is comparable to or better than what has been recently achieved with semiempirical and tight binding methods.[34]

Figure 7 illustrates a typical interactive simulation. Here, the quinoline system introduced above is interactively modeled using RHF and the STO-3G basis set. Each *ab initio* gradient is allowed 200 ms of wall time. The MTS integrator uses 10 substeps, and the force constant of the haptic spring is set to 0.15 hartree/Bohr,[2] which was chosen by trial and error to allow steering of the atomic motions without completely overriding their momenta. Throughout the simulation, the coordinates of the three ammonia nitrogen atoms are frozen to avoid their diffusion into the vacuum. This system has been previously used to experimentally study long distance proton transfer.[35,36] In a similar spirit, we use a haptic tug to remove a proton from the central ammonia and form an ammonium ion on the right as shown in panels B and C of Figure 7. The

DOI: 10.1021/acs.jctc.5b00419
J. Chem. Theory Comput. 2015, 11, 4536−4544

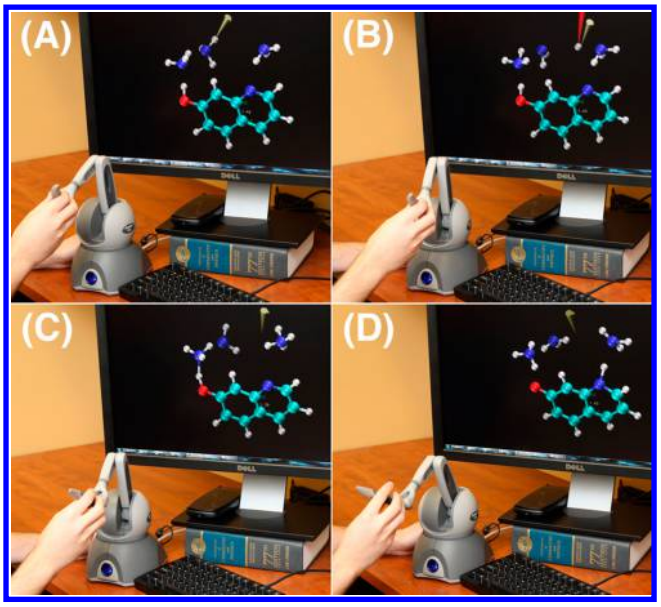

Figure 7. Snapshots of interactive simulation of 7-hydroxyquinoline with three ammonia molecules. (A) The simulation is started from the geometry shown in Figure 4. The Cartesian coordinates of the ammonia nitrogen atoms are fixed to avoid diffusion. (B) Force is applied to transfer a proton from the central ammonia to form an ammonium ion on the right (C). Subsequently, three protons spontaneously transfer, ultimately converting 7-hydroxyquinoline to a tautomeric 7-ketoquinoline (D).

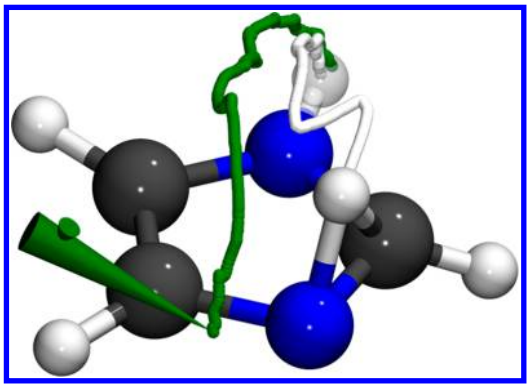

Figure 8. Interactive proton transfer in imidazole molecule. The haptic pointer is attached to a proton originally on the far side of the molecule (translucent atom) and used to pull it across to the nearer nitrogen atom. The path of the haptic pointer is shown in green, while that of the transferring proton is colored in white.

system then spontaneously responds by transferring three additional protons and lengthening the central C−C bond between the quinoline rings to form 7-ketoquinoline as shown in panel D.

We find force-feedback through the haptic device to be an important feature in AI-IMD for several reasons. First, it improves user control by providing a cue to pulling distance. Even with a 3D display, it is sometimes difficult to estimate exact positions within the simulated system. Since the force increases as the spring is stretched, the feedback force helps the user determine where and how hard they are pulling on the system. Second, feedback can provide vector field information that is not easily represented visually. For example, in simulating the imidazole molecule shown in Figure 8, the N−H bonds can in general be bent much more easily than stretched. In attempting to transfer the N−H proton between nitrogen atoms, the user feels that a bending motion is easier and naturally moves the proton along a realistic path. If the user tries to pull the atom into a forbidden region, here the middle of the π-bonding structure of the aromatic ring, the proton resists and instead follows the path shown around the perimeter. It is difficult to represent the field of such repulsions visually. Thus, a force-feedback interface provides a unique and useful perspective.

CONCLUSION

Interactive models have a long history in chemical research. The venerable ball and stick model, invented more than a century ago, still plays an important role in shaping our understanding of chemistry. In that same spirit, interactive *ab initio* calculations represent a new synthesis of intuitive human interfaces with accurate numerical methods. AI-IMD can already be applied to systems containing up to a few dozen atoms at the Hartree−Fock level of theory. As computers and algorithms continue to improve, the scope of on-the-fly calculations will continue to widen. Work is already in progress to extend our work here to higher-level DFT methods and to provide on-the-fly orbital display. These features present no technical challenges beyond what has already been accomplished for Hartree−Fock calculations above.

The integration methods presented here provide robust energy conservation for strong haptic forces as long as the *ab initio* forces are similar to those encountered in equilibrium MD. Of course, the user is free to slam the system into much higher-energy configurations where this assumption simply does not hold. More research is needed to develop graceful ways to continue the integration of the equations of motion in such cases, for example by switching to an empirical description which can be rapidly evaluated at a much shorter time step than is possible for the *ab initio* forces. Such developments would greatly improve the resilience of AI-IMD simulations particularly for nonexpert users. The final test of any model is whether it provides fertile ground in which researchers can formulate and test imaginative scientific questions. This explains the longevity of existing interactive models in chemistry and surely recommends AI-IMD as an area deserving much future research.

AUTHOR INFORMATION

Corresponding Author

*E-mail: toddjmartinez@gmail.com

Notes

The authors declare the following competing financial interest(s): T.J.M. is a founder of PetaChem, LLC.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (OCI-1047577 and ACI-1450179) and by the Department of Defense (Office of the Assistant Secretary of Defense for Research and Engineering) through a National Security Science and Engineering Faculty Fellowship to T.J.M.

REFERENCES

(1) Brooks, F. P.; Ouh-Young, M.; Batter, J. J.; Kilpatrick, P. J. Project GROPE - Haptic Displays for Scientific Visualization. *ACM SIGGRAPH Comp. Graph.* **1990**, *24*, 177.

DOI: 10.1021/acs.jctc.5b00419
J. Chem. Theory Comput. 2015, 11, 4536−4544

(2) Johnson, C. K. *ORTEP: A FORTRAN Thermal-Ellipsoid Plot Program for Crystal Structure Illustrations*; Oak Ridge National Laboratory: Oak Ridge, TN, 1965.
(3) Levinthal, C. Molecular Model-Building by Computer. *Sci. Am.* **1966**, *214*, 42.
(4) Atkinson, W. D.; Bond, K. E.; Tribble, G. L.; Wilson, K. R. Computing with Feeling. *Comp. and Graph.* **1977**, *2*, 97.
(5) Ouh-young, M.; Beard, D. V.; Brooks, F. P. Force Display Performs Better than Visual Display in Simple 6D Docking Task. *IEEE Int. Conf. Robotics and Auto.* **1989**, 1462.
(6) Surles, M. C.; Richardson, J. S.; Richardson, D. C.; Brooks, F. P. Sculpting Proteins Interactively - Continual Energy Minimization Embedded in a Graphical Modeling System. *Protein Sci.* **1994**, *3*, 198.
(7) Leech, J.; Prins, J. F.; Hermans, J. SMD: Visual steering of molecular dynamics for protein design. *IEEE Comput. Sci. Eng.* **1996**, *3*, 38.
(8) Prins, J. F.; Hermans, J.; Mann, G.; Nyland, L. S.; Simons, M. A virtual environment for steered molecular dynamics. *Future Gen. Comp. Sys.* **1999**, *15*, 485.
(9) Stone, J. E.; Gullingsrud, J.; Schulten, K. A System for Interactive Molecular Dynamics Simulation. *ACM Symp. 3D Graph.* **2001**, 191.
(10) Cooper, S.; Khatib, F.; Treuille, A.; Barbero, J.; Lee, J.; Beenen, M.; Leaver-Fay, A.; Baker, D.; Popovic, Z.; Players, F. Predicting protein structures with a multiplayer online game. *Nature* **2010**, *466*, 756.
(11) Grayson, P.; Tajkhorshid, E.; Schulten, K. Mechanisms of selectivity in channels and enzymes studied with interactive molecular dynamics. *Biophys. J.* **2003**, *85*, 36.
(12) Glowacki, D. R.; O'Connor, M.; Calabro, G.; Price, J.; Tew, P.; Mitchell, T.; Hyde, J.; Tew, D. P.; Coughtrie, D. J.; McIntosh-Smith, S. A GPU-accelerated immersive audio-visual framework for interaction with molecular dynamics using consumer depth sensors. *Faraday Discuss.* **2014**, *169*, 63.
(13) Dreher, M.; Piuzzi, M.; Turki, A.; Chavent, M.; Baaden, M.; Ferey, N.; Limet, S.; Raffin, B.; Robert, S. Interactive Molecular Dynamics: Scaling up to Large Systems. *Procedia Comput. Sci.* **2013**, *18*, 20.
(14) Bosson, M.; Grudinin, S.; Redon, S. Block-adaptive quantum mechanics: An adaptive divide-and-conquer approach to interactive quantum chemistry. *J. Comput. Chem.* **2013**, *34*, 492.
(15) Bosson, M.; Richard, C.; Plet, A.; Grudinin, S.; Redon, S. Interactive quantum chemistry: A divide-and-conquer ASED-MO method. *J. Comput. Chem.* **2012**, *33*, 779.
(16) Haag, M. P.; Vaucher, A. C.; Bosson, M.; Redon, S.; Reiher, M. Interactive Chemical Reactivity Exploration. *ChemPhysChem* **2014**, *15*, 3301.
(17) Marti, K. H.; Reiher, M. Haptic Quantum Chemistry. *J. Comput. Chem.* **2009**, *30*, 2010.
(18) Haag, M. P.; Marti, K. H.; Reiher, M. Generation of Potential Energy Surfaces in High Dimensions and Their Haptic Exploration. *ChemPhysChem* **2011**, *12*, 3204.
(19) Haag, M. P.; Reiher, M. Real-time quantum chemistry. *Int. J. Quantum Chem.* **2013**, *113*, 8.
(20) 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*, 2619.
(21) Humphrey, W.; Dalke, A.; Schulten, K. VMD: Visual molecular dynamics. *J. Mol. Graphics* **1996**, *14*, 33.
(22) Hu, X. Q.; Yang, W. T. Accelerating self-consistent field convergence with the augmented Roothaan-Hall energy function. *J. Chem. Phys.* **2010**, *132*, 054109.
(23) Pulay, P. Improved Scf Convergence Acceleration. *J. Comput. Chem.* **1982**, *3*, 556.
(24) Tuckerman, M.; Berne, B. J.; Martyna, G. J. Reversible Multiple Time Scale Molecular-Dynamics. *J. Chem. Phys.* **1992**, *97*, 1990.
(25) Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R. A Computer-Simulation Method for the Calculation of Equilibrium-Constants for the Formation of Physical Clusters of Molecules - Application to Small Water Clusters. *J. Chem. Phys.* **1982**, *76*, 637.
(26) Kulik, H. J.; Luehr, N.; Ufimtsev, I. S.; Martinez, T. J. Ab Initio Quantum Chemistry for Protein Structures. *J. Phys. Chem. B* **2012**, *116*, 12501.
(27) Ufimtsev, I. S.; Luehr, N.; Martinez, T. J. Charge Transfer and Polarization in Solvated Proteins from Ab Initio Molecular Dynamics. *J. Phys. Chem. Lett.* **2011**, *2*, 1789.
(28) Ufimtsev, I. S.; Martinez, T. J. Quantum chemistry on graphical processing units. 1. Strategies for two-electron integral evaluation. *J. Chem. Theory Comput.* **2008**, *4*, 222.
(29) Overall performance does not seem overly sensitive to the implementation details of the mutex.
(30) https://github.com/vrpn/vrpn/wiki (accessed July 30, 2015).
(31) http://www.geomagic.com/en/products/phantom-omni/overview (accessed July 31, 2015).
(32) Ong, M. T.; Leiding, J.; Tao, H. L.; Virshup, A. M.; Martinez, T. J. First Principles Dynamics and Minimum Energy Pathways for Mechanochemical Ring Opening of Cyclobutene. *J. Am. Chem. Soc.* **2009**, *131*, 6377.
(33) Luehr, N.; Ufimtsev, I. S.; Martinez, T. J. Dynamic Precision for Electron Repulsion Integral Evaluation on Graphical Processing Units (GPUs). *J. Chem. Theory Comput.* **2011**, *7*, 949.
(34) Haag, M. P.; Reiher, M. Studying chemical reactivity in a virtual environment. *Faraday Discuss.* **2014**, *169*, 89.
(35) Tanner, C.; Manca, C.; Leutwyler, S. 7-hydroxyquinoline center dot(NH3)(3): A model for excited state H-atom transfer along an ammonia wire. *Chimia* **2004**, *58*, 234.
(36) Tanner, C.; Manca, C.; Leutwyler, S. Probing the Threshold to H Atom Transfer Along a Hydrogen-Bonded Ammonia Wire. *Science* **2003**, *302*, 1736.

DOI: 10.1021/acs.jctc.5b00419
J. Chem. Theory Comput. 2015, 11, 4536−4544