Car-Parrinello Monitor for more robust Born-Oppenheimer molecular dynamics

Lee-Ping Wang[*,†] and Chenchen Song[‡]

†*Department of Chemistry, University of California; 1 Shields Ave; Davis, CA 95616.*
‡*Department of Chemistry and the PULSE Institute, Stanford University; Stanford, CA 94305.*
¶*SLAC National Accelerator Laboratory; Menlo Park, CA 94025.*

E-mail: leeping@ucdavis.edu

Abstract

Born-Oppenheimer molecular dynamics (BOMD) is a promising simulation method for exploring the possible reaction pathways of a chemical system, but one significant challenge is the increased difficulty of converging the self-consistent field (SCF) calculation that often accompanies the breaking and forming of chemical bonds. To address this challenge, we developed an enhancement to the BOMD simulation method called the Car-Parrinello monitor (CPMonitor) that uses Car-Parrinello molecular dynamics (CPMD) to recover from SCF convergence failures. CPMonitor works by detecting SCF convergence failures in BOMD and switching to a CPMD Hamiltonian to propagate through the region of configuration space where the SCF calculation is unable to converge, then switching back to BOMD when good convergence behavior is reestablished. We present a series of simulation studies that use CPMonitor, including detailed studies of the thermodynamic and dynamical properties of simple systems, as well as *ab initio* nanoreactor simulations containing transition metal atoms that were previously not possible to simulate using standard BOMD methods. Our studies show

that CPMonitor can make BOMD simulations robust to SCF convergence difficulties and improve simulation performance and stability in reaction discovery applications.

1   Introduction

The applications of molecular dynamics (MD) simulations to evaluate properties and discover phenomena are widespread in theoretical chemistry. Born-Oppenheimer molecular dynamics (BOMD) is a MD simulation carried out on an *ab initio* potential energy surface (PES), characterized by evaluation of the electronic ground state and energy at every time step ("on-the-fly").[1–3] The first BOMD simulation using an *ab initio* PES was first reported in 1978 by Leforestier,[4] and used the Hartree-Fock approximation (HF) and the STO-3G basis to describe the gas phase reaction $H^- + CH_4 \longrightarrow CH_4 + H^-$.[5] This work built on prior research that carried out MD simulations on semiempirical potential surfaces[6] and analytic potential surfaces constructed by interpolation from grids of *ab initio* calculations.[7] Since then, BOMD has expanded into a great number of applications including simulation of thermodynamic properties,[8] spectra,[9,10] and reactivity.[11–13]

In contrast to most empirical potentials, *ab initio* potential surfaces are able to correctly describe the formation and dissociation of many types of chemical bonds. BOMD can provide insight into chemical reaction mechanisms independently of mechanistic hypotheses because individual reaction events can occur naturally by integrating the equations of motion. Recently, dynamics-based methods for discovering reaction pathways have developed alongside automated enumeration methods for exploring hypothetical reaction coordinates, providing promising alternatives to relying on chemical intuition for mechanistic insights.[12–21] However, because BOMD requires evaluation of the *ab initio* potential energy surface at each time step, it is computationally expensive and the time scales of many interesting reaction events remain out of the reach of unbiased simulations. Thus, the most productive applications of BOMD to discover reaction events have often involved simulating extreme conditions

where reactivity occurs more rapidly, or the artificial introduction of extreme conditions to discover the reaction events that may be possible at ambient conditions.

One major challenge facing reactive BOMD simulations is that the self-consistent field (SCF) equations required to solve for the *ab initio* PES can fail to converge. The SCF procedure is required to solve for the molecular orbitals and ground state energy in HF and density functional theory (DFT), both of which are commonly used in BOMD simulations. One principal cause of convergence failures is the closing of the energy gap between occupied and virtual molecular orbital energy levels, which is more likely to occur in regions of configuration space characterized by broken bonds.[22] Thus, SCF convergence failures are more likely in BOMD simulations at extreme conditions compared to simulations at ambient conditions or equilibrium structures. Additionally, there is a question of whether SCF methods are appropriate for describing systems that fail to converge because the correct electronic state may contain significant multireference character. Although this question is important, we think that making BOMD simulations robust against SCF convergence failures is worth pursuing because they are often used to provide initial mechanistic hypotheses without needing to be highly accurate - in other words, a pathway discovered using a SCF-level simulation may be refined at a higher, possibly multireference level of theory.[23] Furthermore, these studies can provide more insights into the size and nature of errors incurred by making single-reference approximations in different systems exhibiting multireference character.[24]

Convergence behavior can often be improved using alternative convergence algorithms, as the default algorithm in most codes - the direct inversion in the iterative subspace (DIIS)[25] - has a failure mode in which the iterations can oscillate between multiple partially converged solutions.[26] Alternative algorithms such as geometric direct minimization (GDM),[27] energy DIIS (EDIIS),[28] augmented DIIS (ADIIS),[29] and optimal damping (ODA)[30] aim to converge toward a solution more reliably at the cost of increasing the average number of iterations needed to reach the convergence threshold, and are often used as backup algorithms in case DIIS fails to converge. In addition, the large occupied-virtual rotations caused by a vanish-

ing HOMO-LUMO gap can be mitigated using level shifting,[31] which involves increasing the diagonal entries of the Fock matrix corresponding to virtual orbital indices by a shift parameter prior to diagonalization. Fractional occupation methods, also called electron smearing, may also be used to aid convergence and have proved their effectiveness in density functional calculations of metallic systems, which have no band gap.[32] This approach involves partially populating the orbitals according to a predetermined distribution around the Fermi level corresponding to a temperature parameter, which may be annealed down to 0 K when the iterations approach convergence.

Although there exist many strategies for treating difficult cases of SCF convergence, no single strategy is general enough to fully resolve the convergence problems that occur for reactive BOMD simulations. The performance of the different convergence algorithms, and the proper values of adjustable parameters for level shifting and fractional occupation, all depend on the system and configuration being considered.[32,33] A single simulation trajectory could sample regions of configuration space that differ in the types of chemical species, which may require different choices of algorithms and parameters to ensure convergence. Even if convergence issues could be resolved for a given configuration by making the appropriate choices of methods and parameters, it requires time-consuming manual intervention that limits the usefulness of these reactive simulations in practice. Therefore, there still exists a need for general methods to treat SCF convergence difficulties in reactive BOMD simulations.

In this paper, we introduce such a simulation method, called the *Car-Parrinello monitor* (CPMonitor), which augments the capabilities of standard BOMD simulations by using Car-Parrinello MD (CPMD) to propagate through regions of configuration space where SCF convergence failures appear. Car-Parrinello MD is a well-established simulation method first described in Ref. 34, which differs from BOMD in that the electronic variables are treated as dynamical degrees of freedom in an extended Lagrangian formalism, complete with fictitious electronic masses and velocities. The majority of CPMD applications have been in periodic DFT simulations of extended systems,[35–39] and formulations of CPMD for atom-centered

basis sets have also been developed[40-43] that use either the molecular orbital coefficients or the density matrix in the orthogonalized orbital basis as the dynamical variables. Generally speaking, CPMD simulations do not require converging the SCF equations at each time step, but they require the use of a reduced time step (typically 0.1 fs vs. 1.0 fs) and the simulated properties may differ from BOMD, depending on the choice of the fictitious electron mass parameter.[44,45] CPMD is also closely related to the extended-Lagrangian BOMD approaches by Niklasson and coworkers,[46-48] where an auxiliary density matrix close to the ground-state density is propagated in time and used as an initial guess to the SCF procedure, improving the time-reversibility characteristics of the BOMD simulation.

A BOMD simulation equipped with CPMonitor is capable of switching between a normal mode that uses BOMD, or a recovery mode that uses CPMD. Upon the detection of SCF convergence failure in BOMD, the CPMD recovery mode is initialized from a previously saved point in the trajectory and used to propagate the trajectory until normal SCF convergence behavior is restored. In this paper, we first introduce and discuss the concepts and logic of CPMonitor. We then present a series of three simulation studies where CPMonitor is applied in increasingly complex and difficult systems. The first two systems provide a detailed analysis of what happens when CPMonitor switches between BOMD and CPMD modes, and the third system shows how CPMonitor can enable new applications of ab initio MD in reaction discovery that were not previously possible. Thus, we think that CPMonitor is a valuable addition to *ab initio* MD simulation methods, especially in applications that simulate extreme conditions and in reaction discovery contexts. The appendix describes some details of our CPMD implementation including some new developments that improve CPMD performance.

2 Methods

2.1 Born-Oppenheimer molecular dynamics

Born-Oppenheimer MD (BOMD) is widely used to carry out AIMD simulations of molecular systems employing Gaussian basis sets. In this paper, we use BOMD as a reference point and only deviate from it when poor SCF convergence makes it impossible or impractical to take BOMD time steps. A brief overview of BOMD is provided here for comparing and constrasting the dynamical variables and equations of motion with CPMD, which is important for introducing CPMonitor.

The dynamical variables of BOMD are the $3N$ nuclear degrees of freedom $\mathbf{r}$ with corresponding masses $\mathbf{m}$ and velocities $\mathbf{v}$. In a NVE microcanonical ensemble BOMD simulation, the total energy of the system $\mathcal{E}_{\mathrm{BOMD}}$ is conserved:

$$\mathcal{E}_{\mathrm{BOMD}}(\mathbf{r}, \mathbf{v}) = E_{\mathrm{SCF}}(\mathbf{r}) + \frac{1}{2} \sum_{i=1}^{3N_{\mathrm{atom}}} m_i v_i^2 \tag{1}$$

We assume the Born-Oppenheimer potential energy of the system E_{SCF} is obtained by variational minimization of a single Slater determinant, commonly carried out by the self-consistent field (SCF) procedure as:

$$E(\mathbf{r}, \mathbf{P}') = \mathrm{Tr} \left[\mathbf{P}' \left(\mathbf{H}_0'(\mathbf{r}) + \frac{1}{2}\mathbf{G}'(\mathbf{r}, \mathbf{P}') \right) \right] + V_{NN}(\mathbf{r})$$

$$\mathbf{P}'_{\mathrm{SCF}}(\mathbf{r}) = \underset{\mathbf{P}'}{\mathrm{argmin}}\, E(\mathbf{r}, \mathbf{P}'); \quad E_{\mathrm{SCF}}(\mathbf{r}) = E(\mathbf{r}, \mathbf{P}'_{\mathrm{SCF}}(\mathbf{r})) \tag{2}$$

where $\mathbf{P}'$ is the density matrix, V_{NN} the nuclear repulsion energy, and $\mathbf{H}_0'$ and $\mathbf{G}'$ are the one-electron and two-electron parts of the Fock matrix $\mathbf{F}'$ respectively. The primes indicate we are in the nonorthogonal AO basis. This variational procedure encompasses Hartree-Fock as well as hybrid and range-separated Kohn-Sham DFT functionals. The minimization of Eq. 2 with respect to $\mathbf{P}'$ is further subject to the idempotency constraint on the density matrix in the orthogonalized AO basis, i.e. $\mathbf{PP} = \mathbf{P}$. The orthogonal and nonorthogonal bases are related

6

as $\mathbf{P} = \mathbf{U}\mathbf{P}'\mathbf{U}^T$ for the density matrix and $\mathbf{H}_0 = \mathbf{U}^{-T}\mathbf{H}_0'\mathbf{U}^{-1}$ for one-electron operators, where the orthogonalization matrix $\mathbf{U}$ is related to the overlap matrix $\mathbf{S}'$ as $\mathbf{S}' = \mathbf{U}^T\mathbf{U}$. In practice, applying the calculus of variations to the constrained minimization problem leads to the nonlinear eigenvalue equations for the Hartree-Fock or Kohn-Sham orbitals that are implemented in most codes.

Convergence is reached when an error metric, such as the norm or maximum element of the DIIS error vector $\mathbf{e} = \mathbf{F}'\mathbf{P}'\mathbf{S}' - \mathbf{S}'\mathbf{P}'\mathbf{F}'$ ($\mathbf{S}'$ is the overlap matrix), falls below a threshold. Following SCF convergence, the nuclear gradient of the SCF energy may be computed for use in MD simulation. Owing to the variational nature of E_{SCF}, the analytic gradient does not require solving for the orbital response and may be obtained at relatively low cost equivalent to only one or a few SCF cycles.

The classical equations of motion for the dynamical variables are given by Newton's laws relating acceleration a to the classical force f, given by the negative total derivative (nuclear gradient) of the potential energy:

$$m_i\ddot{r}_i = m_i a_i = f_i = -\frac{dE_{\mathrm{SCF}}}{dr_i} \tag{3}$$

The velocity Verlet algorithm is a standard method for integrating the equations of motion. Starting with known $\mathbf{r}, \mathbf{v}, \mathbf{a}$ at time t, the positions are propagated forward in time by one time step δt as

$$\mathbf{r}_+ = \mathbf{r} + \mathbf{v} \cdot \delta t + \frac{\mathbf{a}}{2} \cdot \delta t^2 \tag{4}$$

The SCF calculation performed at the updated positions $\mathbf{r}_+$ provides the updated accelerations $\mathbf{a}_+$, then the velocities are updated as

$$\mathbf{v}_+ = \mathbf{v} + \frac{1}{2}(\mathbf{a} + \mathbf{a}_+) \cdot \delta t = \frac{\mathbf{r}_+ - \mathbf{r}}{\delta t} + \frac{\mathbf{a}_+ \cdot \delta t}{2} \tag{5}$$

The coordinates may be written to a trajectory file for every new step, or at regularly

spaced intervals. A restart or checkpoint file containing highly precise positions, velocities and accelerations allows the simulation to run to completion if interrupted.

Because a BOMD simulation trajectory can be terminated prematurely by SCF convergence failure, a transferable method to automatically deal with convergence failures is needed to reliably propagate simulation trajectories to their intended length. In previous work, we implemented a simple composite SCF algorithm that invokes ADIIS automatically when the standard DIIS fails to converge; this algorithm enabled us to carry out *ab initio* nanoreactor simulations using restricted and unrestricted HF that sampled hundresds of chemical species and reaction mechanisms.[13] However, attempts to incorporate transition metal atoms into the simulation, or the use of semilocal or hybrid density functionals such as BP86 and B3LYP, were still stymied by SCF convergence failures. We attributed the increased occurrence of convergence failures to the smaller HOMO-LUMO gaps in these systems.[49] These convergence challenges inspired the idea of using CPMD to propagate through the regions of configuration space that contain convergence failures.

2.2   Car-Parrinello molecular dynamics

In Car-Parrinello molecular dynamics (CPMD),[34] both the nuclear and electronic degrees of freedom are treated as dynamical variables. Our implementation of CPMD largely follows the work of Schlegel and coworkers[41,42,50] with some modifications that are summarized in the Appendix. This implementation chooses the density matrix elements in the orthogonal AO basis as the dynamical electronic variables. For brevity we will present the basic equations assuming a restricted HF/KS wavefunction and uniform electron mass. Generalizations to unrestricted wavefunctions and non-uniform masses are mostly straightforward and are implemented into our code.

The equations of motion are derived from the Lagrangian given by:

$$\mathcal{L} = \sum_{i=1}^{3N_{\text{atom}}} \frac{1}{2}m_i v_i^2 + \frac{1}{2}\mu\text{Tr}[\mathbf{W}\mathbf{W}] - E(\mathbf{r}, \mathbf{P}) - \text{Tr}[\mathbf{\Lambda}(\mathbf{P}\mathbf{P} - \mathbf{P})] \tag{6}$$

where the four terms represent the nuclear kinetic energy, the fictitious electronic kinetic energy, the potential energy of the nuclei and electrons, and the Lagrange multiplier term that ensures the density matrix is idempotent, i.e. $\mathbf{P}\mathbf{P} = \mathbf{P}$. Here μ is the fictitious electron mass, usually between 0.01 and 0.5 amu, $\mathbf{W}$ the velocities of density matrix coefficients, and $\mathbf{\Lambda}$ the matrix of Lagrange multipliers. The conserved quantity in CPMD simulations is given by:

$$\mathcal{E}_{\text{CPMD}}(\mathbf{r}, \mathbf{v}, \mathbf{P}, \mathbf{W}) = \sum_{i=1}^{3N_{\text{atom}}} \frac{1}{2} m_i v_i^2 + \frac{1}{2} \mu \text{Tr}[\mathbf{W}\mathbf{W}] + E(\mathbf{r}, \mathbf{P}) \tag{7}$$

The equations of motion for dynamical variables are obtained via the Euler-Lagrange equations. For the nuclear positions and density matrix elements, we have:

$$m_i \ddot{r}_i = m_i a_i = f_i = -\left. \frac{\partial E(\mathbf{r}, \mathbf{P})}{\partial \mathbf{r}_i} \right|_{\mathbf{P}} \tag{8}$$

$$\mu \ddot{\mathbf{P}} = \mu \mathbf{B} = -\left(\left. \frac{\partial E(\mathbf{r}, \mathbf{P})}{\partial \mathbf{P}} \right|_{\mathbf{r}} + \mathbf{\Lambda}\mathbf{P} + \mathbf{P}\mathbf{\Lambda} - \mathbf{\Lambda} \right) \tag{9}$$

where the density matrix acceleration is denoted as $\mathbf{B}$.

Following the velocity Verlet integration scheme, the nuclear dynamical variables at the next time step $\mathbf{r}_+$, $\mathbf{v}_+$ are given by Eqs. 4-5, using the classical force from Eq. 8. The electronic dynamical variables at the next time step are given by:

$$\mathbf{P}_+ = \mathbf{P} + \mathbf{W} \cdot \delta t + \frac{\mathbf{B}}{2} \cdot \delta t^2 \tag{10}$$

$$\mathbf{W}_+ = \frac{\mathbf{P}_+ - \mathbf{P}}{\delta t} + \frac{\mathbf{B}_+ \cdot \delta t}{2} \tag{11}$$

where $\mathbf{B}_+$ is obtained by passing $\mathbf{P}_+$, $\mathbf{r}_+$ into Eq. 9. Further discussions of how P_+ and W_+ are computed are given in the Appendix, which includes a conjugate gradient algorithm for more rapid convergence of P_+.

2.3 CPMonitor

In this section, we will discuss the workflow of CPMonitor, as illustrated in Figure 1. All parameters that control the behavior of the method are described in Table 1. When the SCF calculations continue to converge successfully in normal BOMD steps, CPMonitor is said to be in *normal mode* (colored as green in Figure 1). The BOMD steps will also be called "macro-iterations", in contrast with the "micro-iterations" in CPMD as discussed later. Compared with a standard BOMD simulation, the difference is only when the trajectory frame is written to disk. Instead of writing the current coordinates immediately to disk as a new trajectory frame, the CPMonitor writes the coordinates of a previous frame corresponding to time $t - n_S \cdot \delta t$, while storing all newer frames in memory in case of SCF convergence failure in the future. A frame stored in memory consists of the coordinates $\mathbf{r}(t)$, velocities $\mathbf{v}(t)$, accelerations $\mathbf{a}(t)$ and density matrices $\mathbf{P}(t)$, which contains all the information needed to reset the BOMD simulation to the stored frame and initiate CPMD if necessary.

If the SCF calculation fails to converge during a BOMD step, an error message will be sent to CPMonitor. Upon receiving the error message, CPMonitor will write the coordinates of the stored frame to disk as usual. However, since a new frame is not available due to convergence failure, CPMonitor will use the stored frame to reset the simulation system (e.g. coordinate and velocities) and thus enters the *recovery mode* (colored as red). At the start of the recovery mode, the simulation time is set back to the stored frame, i.e. $t - n_S \cdot \delta t$, and the time step is set to $\delta\tau = \delta t / n_C$, where n_C is the number of "micro-iterations". To initialize CPMD, an initial value of the density matrix velocity $\mathbf{W}$ is generated by carrying out a small even number of micro-iterations (n_G) using BOMD and estimating the time derivative using finite difference on the stored density matrices. The CPMD simulation is then initialized to the time point corresponding to where the finite difference derivative is taken. We use a central difference stencil, so CPMD is initiated at time $t = n_S \cdot \delta t + n_G / 2 \cdot \delta\tau$.
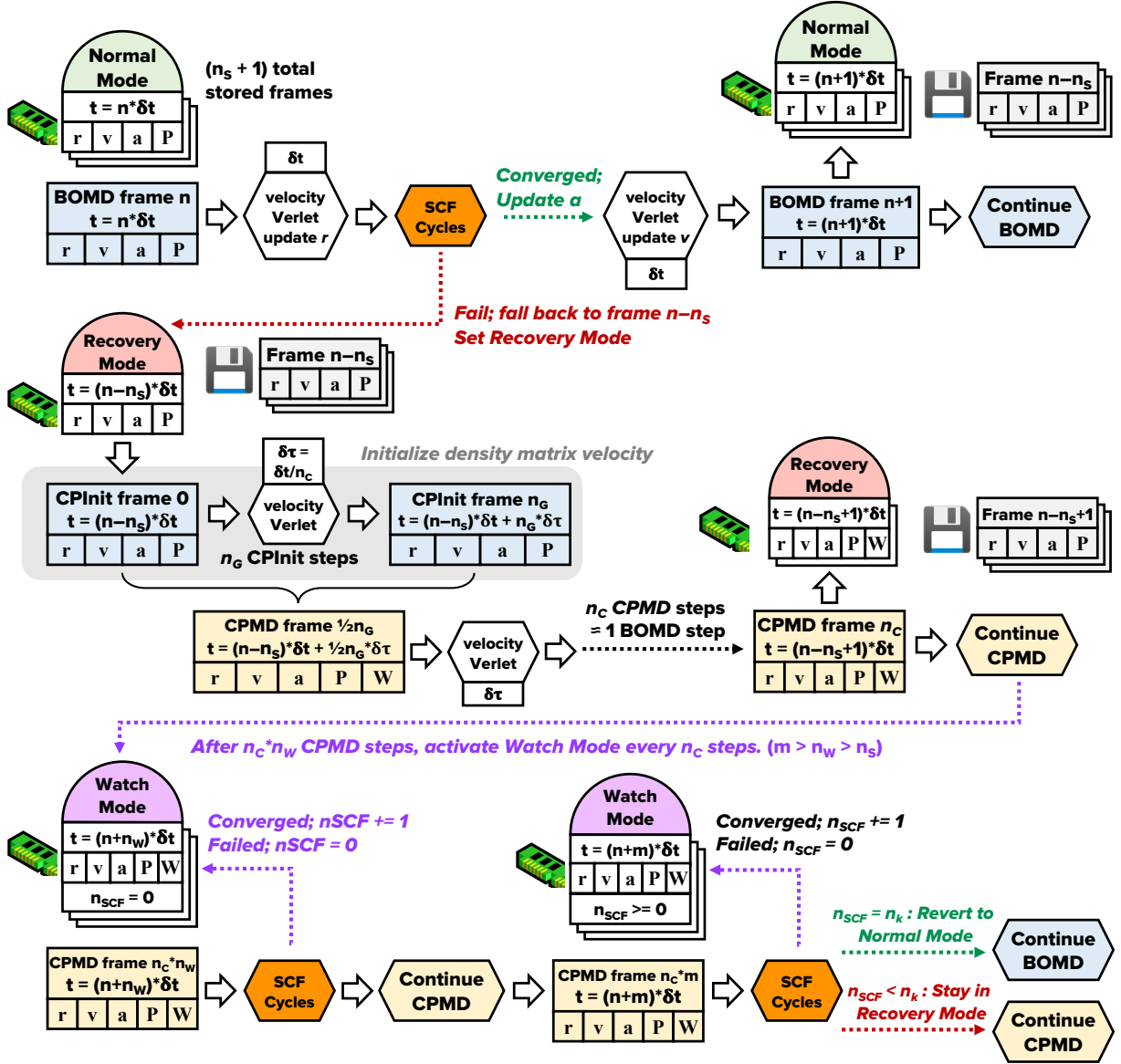
Figure 1: Flowchart illustrating main concepts of CPMonitor. Top row: CPMonitor running in normal mode (green) is the same as normal BOMD except the trajectory is written with a delay of n_S frames. Middle row: At the start of recovery mode (red), the simulation time is set back by n_S frames, the density matrix velocity is initialized, and the simulation continues with a reduced time step. Lower row: Watch mode (purple semicircle) is activated when CPMonitor is in recovery mode and the current time coincides with a BOMD time step. A SCF calculation is started and successful convergence increments a counter. When the number of consecutive successful SCF calculations exceeds a threshold, CPMonitor reverts to normal mode.

Table 1: Table of adjustable parameters for CPMonitor and recommended values.

Name	Definition	Recommended
$\delta t > 0.0$	Simulation time step (fs)	1.0
$n_S \geq 0$	Time steps between current frame in Normal mode and starting point of Recovery mode	2
$n_G \geq 2$	Even number of BOMD microiterations for initializing CPMD density matrix velocity via finite difference	2
$n_C \geq 2$	Subdivisions of the BOMD time step to be used in recovery mode (micro-iterations)	10
$n_W > n_S$	Equivalent number of macro-iterations in Recovery mode needed to activate Watch mode	3
$n_K \geq 1$	Consecutive SCF successes in Watch mode needed to revert to Normal mode	3

The goal of the recovery mode is to use CPMD to get across the region where BOMD fails due to SCF convergence failure. It is important for CPMonitor to keep track of the number of micro-iterations N_{CPMD} and the simulation time. Whenever N_{CPMD} is an integer multiple of n_C, the CPMD simulation time is an integer multiple of the BOMD time step, i.e. n_C CPMD steps is equivalent to a macro-iteration. When this happens, CPMonitor will write and save the current simulation frame, such that the trajectory frames correspond to equally spaced and monotonically increasing simulation times, regardless of whether BOMD or CPMD is used. In addition, if $N_{\text{CPMD}} \geq n_W n_C$, the CPMonitor will enter *watch mode* (colored as purple) that considers when the simulation should go back from CPMD to BOMD. $n_W > n_S$ is a parameter that sets the minimum CPMD simulation length.

The main criterion for switching back to BOMD is that after a certain number of CPMD steps, the system should have exited the region where SCF is hard to converge. Based on this idea, CPMonitor in watch mode will run SCF calculations at CPMD time steps corresponding to the macro-iterations, i.e. where N_{CPMD} is multiple of n_C. A counter is incremented in CPMonitor for a successful SCF calculation and reset to zero for a convergence failure.

If successful SCF convergence is recorded for n_K consecutive macro-iterations, CPMonitor will terminate the CPMD, and return to the normal mode that runs BOMD. Otherwise, CPMonitor will continue the CPMD trajectory normally regardless of the SCF convergence result, remaining in the watch mode that carries out an additional SCF calculation at CPMD time steps corresponding to macro-iterations. The minimum number of CPMD steps needed to recover from a single SCF convergence failure is thus $(n_W + n_K) * n_C$, or 60 CPMD steps using the recommended parameters.

The parameter n_S controls the distance between the current BOMD frame and the written trajectory frame / recovery point. Because the recovery mode requires n_G BOMD micro-iterations to initialize the density matrix velocity, it is important that SCF convergence failures are not encountered near the recovery point itself. The smallest possible value of $n_S = 0$ implies that no additional frames are stored and the recovery mode starts from the immediately preceding BOMD frame. Choosing a larger value of n_S helps to ensure the CPMD simulation can successfully start by increasing the distance from the convergence failure, but it also increases the number of CPMD steps needed to recover before BOMD can resume. In practice we found that setting $n_S = 2$ was sufficient for all applications.

When CPMD is initialized, there is a discontinuous change in the total energy due to the addition of the fictitious electronic kinetic energy term in the Hamiltonian, $\frac{1}{2}\mu\text{Tr}[\mathbf{WW}]$. Similarly, when the recovery mode is complete and CPMD reverts to BOMD, this fictitious term is removed. Thus, one cycle of recovery mode (i.e. BOMD-CPMD-BOMD) leads to a change in the total energy because of energy transfer between the fictitious electronic kinetic energy and the other terms during the CPMD phase. The fictitious term usually increases during the first steps of CPMD following the initialization of $\mathbf{W}$, thus the total energy change of one recovery cycle is negative. In order to approximately preserve energy conservation during our simulation, the energy transferred to the fictitious term is applied the nuclear kinetic energy at the end of recovery mode as a velocity rescaling. The amount of energy transfer is calculated by taking the difference in the BOMD total energy $E_{\text{SCF}}(\mathbf{r})+$

$\frac{1}{2} \sum_{i=1}^{3N_{\text{atom}}} m_i v_i^2$ between the starting and ending time points of recovery mode .

CPMonitor requires that $n_S + 1$ frames are stored in memory in order to recover successfully from SCF convergence failures. This condition is met as long as the BOMD simulation is able to successfully complete at least n_S frames from the initial condition. It is also possible that a SCF convergence failure occurs shortly after clearing the recovery mode, such that fewer than n_S frames have elapsed since resuming the BOMD simulation. In this case, the recovery point falls within a CPMD time segment and the existing $\mathbf{W}$ is used without needing to initialize it from BOMD micro-iterations.

3   Computational Details

We implemented the Car-Parrinello MD and CPMonitor methods into a development version of the TeraChem quantum chemistry package.[51–53] The implementation supports both restricted and unrestricted HF and KS wavefunctions, generalized mass matrices for propagating the orthogonalized AOs, as well as the use of pseudopotentials.[54,55] The matrix $\mathbf{U}$ was calculated using symmetric Löwdin orthonormalization. We also developed an improved procedure for calculating $\mathbf{P}_+, \mathbf{W}_+$ and a new mass-weighting scheme, described in detail in the Appendix. TeraChem accelerates the MD simulations by carrying out the Fock builds on graphics processing units (GPUs) using a mixed-precision model that computes smaller integrals with reduced precision.[56] All other calculations specific to CPMD and CPMonitor, such as the calculation of $\mathbf{P}_+$ and $\mathbf{W}_+$, are performed on the CPU. The correctness of our CPMD implementation is verified by testing energy conservation as shown in Supporting Figure S1. The CPMD and CPMonitor codes are written as separate modules that hook into the main MD loop and SCF routines, thus they do not require making any significant modifications to the original MD / SCF codes.

4 Results and Discussion

This section contains three simulation studies that use CPMonitor on systems of increasing complexity. The first study compares the thermodynamic properties of the CPMonitor simulation with "pure" BOMD and CPMD simulations. The simulated system consists of a single malonate anion and does not exhibit any SCF convergence difficulties, thus CPMonitor's recovery mode is artificially invoked at random intervals. The second study involves an iron carbonyl cluster that contains an actual convergence failure. Here we compare the trajectories from CPMonitor vs. a reference trajectory that uses BOMD with advanced convergence strategies and level shifting as a way to provide insight into how CPMonitor affects the simulation trajectory. The third study is a "nanoreactor" simulation containing an iron complex with a large number of small molecules, with an oscillating boundary condition that artificially induces large numbers of reaction events. The CPMonitor simulation demonstrated improved simulation speeds by reducing wall times by a factor of 3-4 compared to BOMD with level shifting. Moreover, CPMonitor improved robustness against SCF convergence difficulties, as the conventional BOMD simulation was terminated due to SCF convergence failure even when advanced convergence strategies and level shifting were used. Taken together, these studies show that CPMonitor provides improved performance and reliability of BOMD in reaction discovery applications.

4.1 Malonate ion

We investigated how CPMonitor might affect the thermodynamic ensemble of a system by calculating free energy profiles of a proton transfer reaction in a singly deprotonated malonate ion. In the gas phase, this system contains an intramolecular hydrogen bond with a small free energy barrier to proton transfer that is accessible from running unbiased simulations at elevated temperature. The system was modeled using the restricted Hartree-Fock / 6-31G* level of theory and dynamics was simulated using a Langevin integrator with the temperature

set to 500 K. The simulation temperature was chosen in order to sample a significant number of proton transfer events; low temperatures are unable to sample energies near the reaction barrier whereas overly high temperatures tend to dissociate the intramolecular hydrogen bond.

Three simulations were carried out; first, a reference BOMD simulation used a time step of 1.0 fs, and the CPMD simulation used a time step of 0.1 fs. Both simulations used a Langevin thermostat with a collision interval of 100 time steps for temperature control. CPMonitor used the same time step settings as the BOMD and CPMD simulations described above. This system does not have any pathological behavior in terms of SCF convergence, thus the CPMonitor recovery mode was artificially activated using a random number generator with a probability of 10% per MD step. As a result, the CPMonitor trajectory contains significant portions of both BOMD and CPMD simulation steps, amounting to 60% and 40% of the total simulation time respectively.
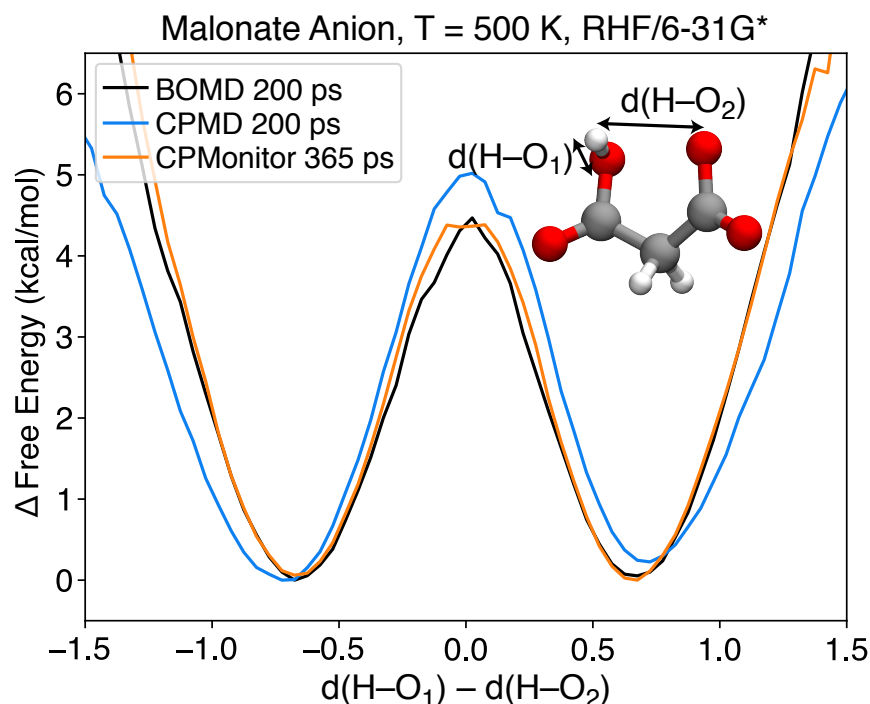

Figure 2: Free energy profile of malonate ion along the hydrogen transfer coordinate.

Figure 2 shows the comparison of free energy profiles obtained using the BOMD, CPMD,

and CPMonitor methods. All three simulations were at least 200 ps in length. The number of transition events recorded for the three simulations were 102, 21, and 152 respectively. The CPMonitor free energy profile closely follows the BOMD profile with a difference of < 0.2 kcal/mol everywhere; this is a positive indicator that introducing many short segments of Car-Parrinello simulation - about 6 fs per segment - does not significantly modify the thermodynamic ensemble. On the other hand, the pure CPMD simulation does show some differences with the BOMD and CPMonitor simulations. The free energy barrier for proton transfer is higher by ≈ 0.5 kcal/mol (also reflected in the reduced number of transfer events), and the sides of the profile have a reduced slope. The CPMD free energy profile also has a greater amount of statistical error due to the slightly different heights of the two free energy basins, possibly owing to the reduced number of transfer events. Evidently, CPMD has a non-negligible effect on the thermodynamic properties of the system, perhaps owing to energy transfer to the fictitious electronic degrees of freedom. Over the 200 ps of total CPMD simulation time, the fictitious kinetic energy increases to ≈ 1.5 a.u. at a rate of approximately 0.01 a.u. per picosecond (10000 time steps). The rate of fictitious kinetic energy increase is consistent with previous studies,[42] however, our much longer simulation times show that the total fictitious kinetic energy can build up to significant quantities. Although around 40% of the CPMonitor trajectory is spent in CPMD, the fictitious kinetic energy never accumulates in excess of 0.003 a.u. due to the very short lengths of CPMD trajectory segments. We think this is the fundamental reason why the CPMonitor results are able to reproduce BOMD more closely than pure CPMD.

4.2   Iron carbonyl cluster

Our second simulation study used an iron carbonyl cluster $Fe_3(CO)_{11}$ as an example system that can encounter SCF convergence difficulties during MD simulations. This cluster is a simplified model of iron-carbonyl clusters that may be capable of energy conversion catalysis; for example, $[Fe_4N(CO)_{12}]^-$ is a known electrocatalyst for CO_2 reduction to formate.[57,58]

These simulations used the BP86 GGA density functional and an all-electron double-zeta basis set, following the work of Ref. 59 that characterized several local energy minima of this system. We compared the results of a BOMD run and a CPMonitor run which used the same set of initial energy-minimized coordinates. A single set of initial velocities was drawn randomly from the Maxwell-Boltzmann distribution and used for both runs. The convergence threshold in the maximum value of the DIIS error vector ϵ was set to 10^{-4}, which corresponds to energy changes on the order of 10^{-6} in the final SCF cycle. The BOMD simulation used the DIIS+A convergence algorithm, which works as follows:

1. If $\epsilon > 10^{-3}$ after 50 total DIIS iterations, ADIIS is activated until the error drops below 10^{-3}, then DIIS is reactivated.

2. If $\epsilon > 10^{-4}$ after 250 total DIIS iterations, ADIIS is activated until SCF convergence is achieved or the limit on the number of SCF cycles is reached (10^5 cycles).

The total simulation lengths of the BOMD and CPMonitor runs were 8479 and 10000 steps respectively (8.479 ps and 10.0 ps). The BOMD simulation was terminated due to a SCF convergence failure whereas the CPMonitor simulation ran to completion. The CPMonitor simulation used a standard DIIS convergence algorithm and the CPMD recovery mode was activated when the number of SCF cycles exceeded 100. Using these settings, the CPMonitor spent 262 fs of the simulation in recovery mode (about 2.6% of the whole trajectory). The ADIIS algorithm was activated a total of 1242 times in the BOMD simulation, and was not able to converge in the final simulation step even though a very high limit of 10^5 SCF cycles was provided. In the final simulation step that caused the simulation to terminate, the DIIS error fluctuated in the range of $0.0003 - 0.3$ without any discernible trend, indicating the convergence failure was not recoverable using DIIS+A alone. In terms of performance, the CPMonitor required 58 s of wall time per 1 fs of simulation time, whereas DIIS+A required 56 s of wall time (excluding the last step of DIIS+A containing the SCF convergence failure); this indicates that both methods have comparable overall performance.
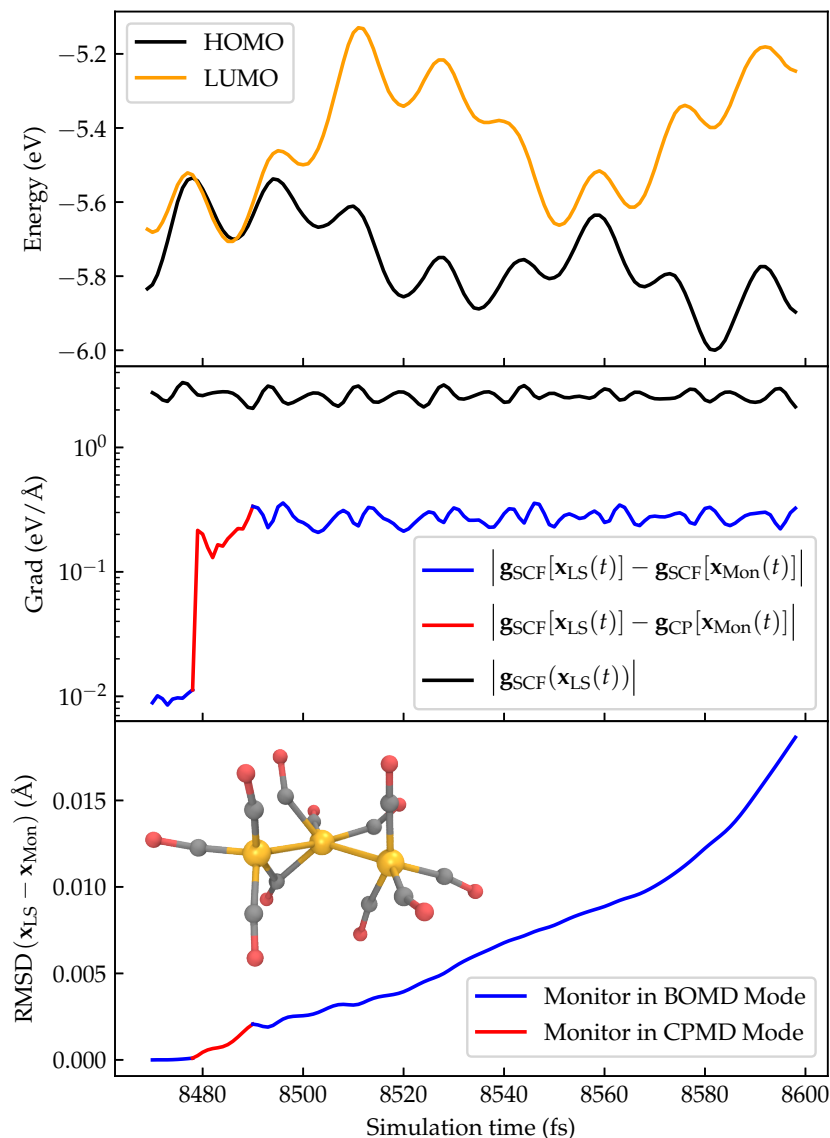
Figure 3: Comparison of CPMonitor simulation trajectory and reference BOMD trajectory with level-shifting. Top panel: HOMO and LUMO energy levels of the reference trajectory, showing the closing of the energy gap at $t = 8480$ fs. Middle panel: Difference in nuclear gradient between the CPMonitor and reference trajectory (blue, CPMonitor in normal mode; red, CPMonitor in recovery mode). The norm of the nuclear gradient of the reference trajectory (black) is shown for comparison. Bottom panel: RMSD of atomic positions between the CPMonitor and reference trajectory (blue, CPMonitor in normal mode; red, CPMonitor in recovery mode.)

Supporting Figure S2 compares the radial distribution functions of the Fe-C, Fe-Fe and C-C atom pairs in the CPMonitor and BOMD simulations of this system. The closely

19

overlapping distributions indicate that the simulation trajectories are similar along these reduced coordinates. On the other hand, we expect any small difference between the trajectories grow exponentially due to Lyapunov instability. Thus, it is helpful to compare BOMD and CPMonitor simulations at the point where they initially diverge.

Figure 3 illustrates such a case where two short simulations were started from a few time steps preceding the SCF convergence failure in the original BOMD simulation. In one simulation, a new BOMD simulation was carried out with a level shift of 1.6 a.u. added, which enabled the SCF calculations to propagate through the challenging region but also increased the number of iterations by a factor of 10. The other simulation used CPMonitor which propagated through the same challenging region using recovery mode. We evaluated the difference in the nuclear gradients between the two trajectories at the same time point and found that it was on the order of 0.2-0.3 eV/Å, one order of magnitude smaller than the norm of the gradient itself. In particular, the gradient differences between CPMonitor and BOMD are similar in size during and after recovery mode, where the gradient differences after recovery mode are solely due to the differences in nuclear configurations. The RMSD between the trajectories after existing recovery mode is around 0.002 Å, which provides some assurance that one recovery cycle of CPMonitor only has a minor impact on the simulation trajectory. Although the BOMD and CPMonitor trajectories are still going to diverge exponentially, we think the two simulation methods should still yield similar results when used in reaction discovery applications, as we will study in the next system.

4.3   Nanoreactor simulation with transition metal complex

Our third model system is intended to measure computational performance and demonstrate the usefulness of CPMonitor in reactive molecular dynamics simulations. The system was set up by modifying the initial configuration of an *ab initio* nanoreactor simulation reported in Ref. 13. The initial coordinates consisted of the following small molecules distributed in a spherical region with a 14 Å radius: 16 H_2, 14 H_2O, 14 CH_4, 14 NH_3, 14 CO. A single

[Fe(H$_2$O)$_6$]$^{2+}$ complex was placed in the center of the sphere, which set the net charge of the system to +2 with a spin multiplicity of 5. The transition metal complex was not included in our previous studies, and introduces the possibility of observing transition metal-catalyzed reaction pathways. These simulations used the B3LYP hybrid functional, the LANL2DZ basis set / ECP combination for Fe, and the 3-21G basis set for all other elements, whereas previous nanoreactor simulations used Hartree-Fock theory and avoided transition metals in the system to facilitate SCF convergence. Thus, the current system is intended to show that CPMonitor provides additional capabilities to expand the domain of applicability of reactive *ab initio* molecular dynamics.

We carried out four simulations of this model system that varied in their SCF convergence strategy. All simulations were started from the same initial conditions containing 247 atoms, 722 electrons and 1048 atomic basis functions. The MD simulations used a velocity Verlet integrator with a time step of 1.0 fs and a Langevin thermostat with an equilibrium temperature of 1000 K and collision interval of 300 time steps. The initial guess for each BOMD step was set using the converged solution of the previous step and used directly to initiate SCF cycles without any density matrix purification or mixing of orbital occupations. A spherically symmetric one-body potential was added to all atoms the system (except Fe) as follows:

$$V(r,t) = \frac{m}{2} \left[k(t)\rho(t)^2 - \kappa(t)r^2 \right]; \quad \rho = \begin{cases} r - R(t) & \text{if } r > R(t) \\ 0 & \text{otherwise} \end{cases}$$

$$k(t), \kappa(t), R(t) = \begin{cases} k_1, \kappa_1, R_1 & \text{if } t \bmod (t_1 + t_2) < t_1 \\ k_2, \kappa_2, R_2 & \text{otherwise} \end{cases}$$

(12)

where m is the atomic mass, and the parameters are given as: $t_1 = t_2 = 1000 \, \delta t$; $k_1, k_2 = 2.0, 1.0$ kcal mol^{-1} amu^{-1}; $R_1, R_2 = 14.0, 9.0$ Å; $\kappa_1, \kappa_2 = 0.02, 0.0$ kcal mol^{-1} amu^{-1}. The boundary parameters are set to oscillate between values of k_1, κ_1, R_1 and k_2, κ_2, R_2 with time

21

intervals of t_1 and t_2, causing the atoms to be pushed outward and inward with a force proportional to their mass. The Fe atom is not subject to the boundary potential, and a harmonic restraint is used instead with a force constant of $k_{\mathrm{F}} = 2.0 \, \mathrm{kcal \, mol^{-1}}$ ensuring that it stays near the sphere center. The CPMonitor simulation used a standard DIIS algorithm with a level shift parameter of 0.1 and a maximum iteration number of 300. For comparison purposes, three BOMD simulations were carried out that used the DIIS+A convergence algorithm (described in §4.2) with different choices of level shifting parameter (0.1, 0.5 and 1.0 a.u.).

Table 2 and Figure 4 compare the performance of the CPMonitor and reference BOMD simulations. As the level shifting parameter is increased in the BOMD simulations, the mean number of SCF cycles per MD step increases as well. Each of the BOMD simulations was terminated after encountering a configuration that did not converge after 10^5 SCF iterations. The BOMD simulation that used a level shift parameter of $\epsilon = 0.1$ had the smallest mean number of SCF cycles, but the simulation was terminated after only 266 steps. When level shift parameters of $\epsilon = 0.5$ and $\epsilon = 0.1$ were used, improved stability came at the cost of a higher number of SCF iterations on average, yet both simulations were terminated after 7278 and 7170 steps respectively. The CPMonitor simulation ran for the full length of 25000 steps and entered recovery mode 144 times. In more than half of these instances, the recovery mode was able to return to BOMD within the minimum number of 60 micro-iterations (6 macro-iterations) and the longest recovery segment was 80 macro-iterations near the beginning of the simulation starting at step 1298. Overall, 6.2% of the simulation time was spent in recovery mode, corresponding to 28.9% of the total wall time.

Table 2: Computational cost characteristics of BOMD and CPMonitor trajectories.

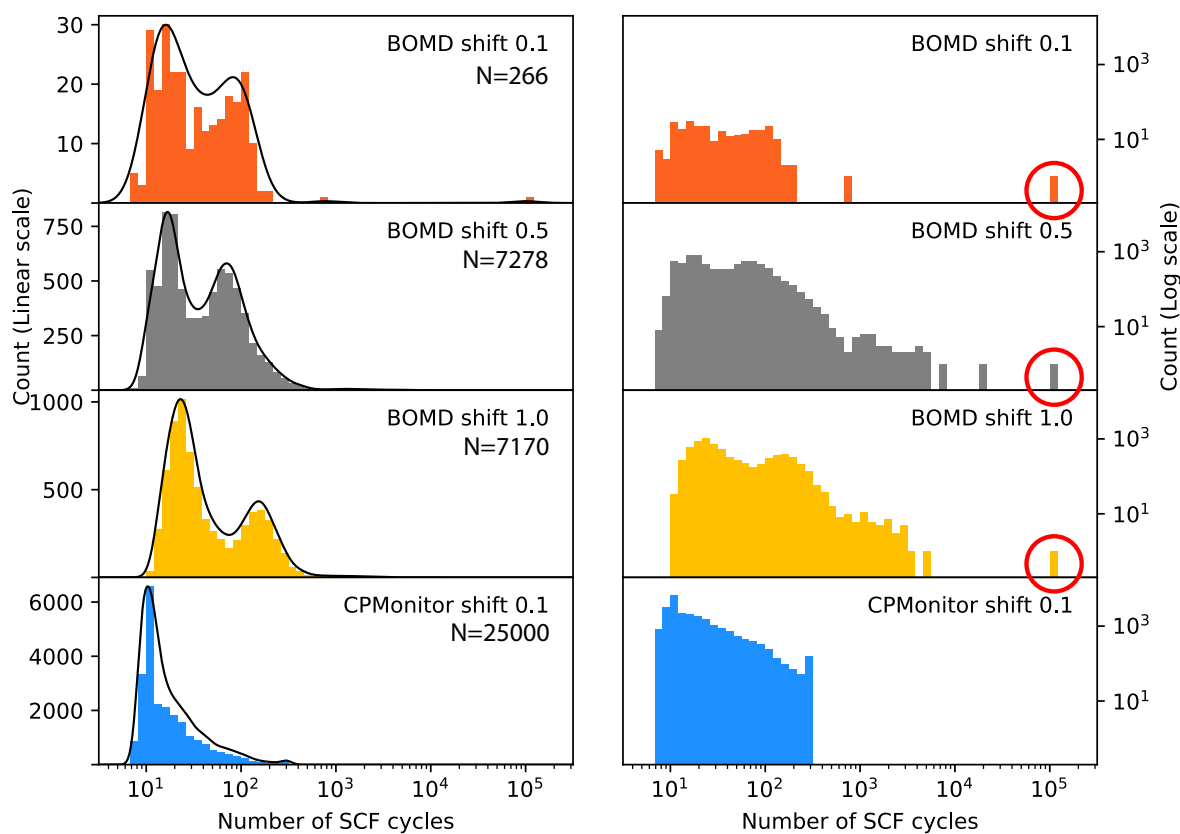| Method	Level Shift	Time Steps	Number of SCF cycles			Time per MD step		
			Mean	Median	Max	Mean	Median	Max
BOMD	0.1	266	423 (49)	29	100000	2373 (162)	94	5.8×10^5
BOMD	0.5	7278	83 (69)	34	100000	356 (274)	120	5.9×10^5
BOMD	1.0	7170	98 (84)	32	100000	396 (314)	110	5.9×10^5
CPMonitor	0.1	25000	23	13	301	117	51	6733

Figure 4: Distribution of the number of SCF cycles per AIMD time step (macro-iteration). The left column and right column show the same data using linear and log scales on the y-axis respectively. The kernel density estimates in the left column are guides for the eye. The circled bars in the right column indicate SCF calculations that did not converge.

The results show that CPMonitor provides improved performance over BOMD in terms of both simulation stability and overall speed. Compared to the best-case scenario for BOMD with level shifting ($\epsilon = 0.5$), CPMonitor provides a roughly 2.5-3.5$\times$ speed increase in terms of wall clock time per MD step. The improved performance mainly comes from a smaller number of SCF cycles needed to reach convergence in the BOMD simulation steps due to a smaller level shifting parameter being used ($\epsilon = 0.1$). Moreover, because CPMonitor enters into recovery mode when a SCF convergence error is encountered, the maximum number of SCF iterations per MD step is fixed, thus there are no time steps where very large numbers of SCF cycles are needed. By comparison, the BOMD simulations with level shifting often required in excess of 1000 SCF cycles to converge, which adversely impacts performance and may lead to termination.
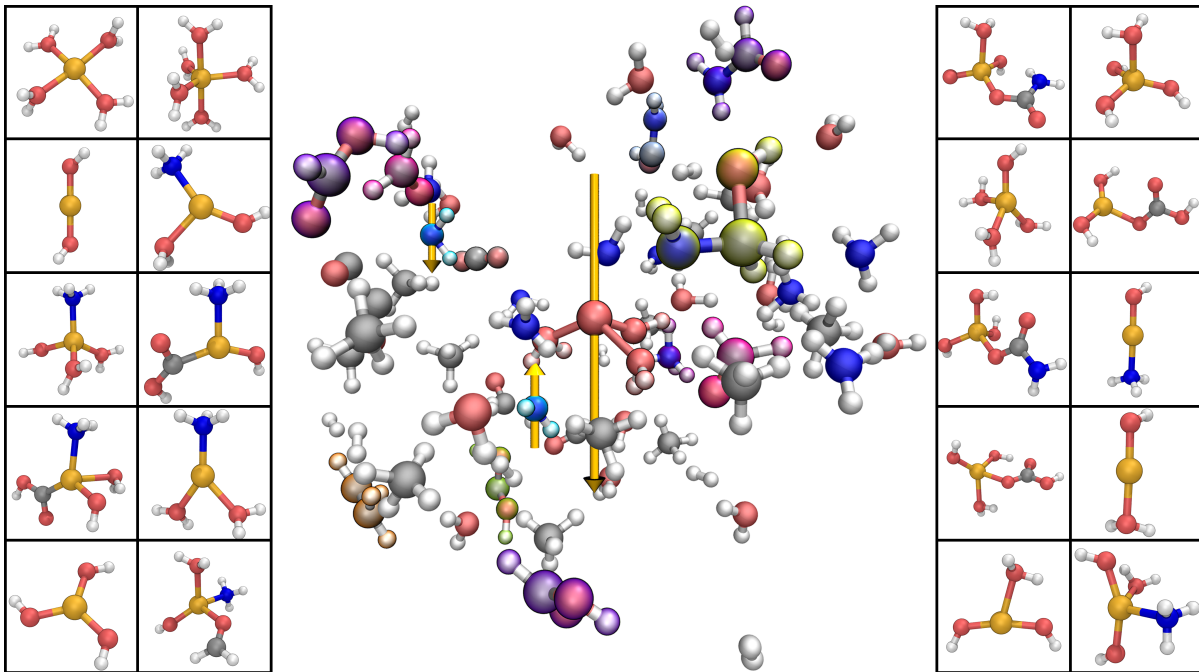

Figure 5: Center: 3-D rendering of a trajectory snapshot from the CPMonitor nanoreactor simulation. Side panels: Optimized structures of Fe-containing coordination complexes that were encountered over the course of the simulation trajectory.

We examined the molecular species in the BOMD and CPMonitor trajectories to understand whether the two simulation methods had any differences in their ability to discover

new chemical species and reactivity. Figure 5 shows a snapshot of the CPMonitor simulation trajectory and lists the distinct coordination complexes of Fe that occurred throughout the simulation. More than 20 different coordination complexes of Fe were formed throughout the 25 ps simulation, as shown by the optimized structures in the side panels. A Mulliken population analysis shows that the number of unpaired electrons on the iron complex is 4 +- 1 throughout the entire trajectory.

Figure 6: Organic molecules and radicals that were found in the nanoreactor simulations, grouped by empirical formula of heavy atoms. Each molecule is color coded by the simulation trajectories that it occurred in. The Venn diagram counts the number of molecules that were found in the different possible combinations of simulations.

Figure 6 lists the organic molecules and radicals that were found. A total of 62 organic molecules were formed in the CPMonitor trajectory, compared to the 14 that were found in each of the BOMD trajectories (a total of 22 combined). The ratio in the number of compounds formed ($64 : 14 = 4.4 : 1$) is close to the ratio of trajectory lengths (25 ps $: 7$ ps $= 3.6 : 1$). The majority of molecules found in the BOMD trajectories were also found in the CPMonitor trajectory (16 out of 22), which indicates CPMonitor simulations have similar

reactivity to BOMD. The six species that appeared in BOMD but not in the CPMonitor trajectory are attributed to incomplete sampling in the simulations.

5   Conclusion

As shown by our simulation studies, CPMonitor can reliably augment BOMD simulations with the capability to propagate through regions of configuration space that contain difficult SCF convergence behavior. We expect CPMonitor to be especially useful for AIMD simulations under extreme conditions and in reaction discovery contexts, because these simulations have especially high probabilities of entering into these computationally challenging regions. Although this work addresses an important technical problem, there is still the underlying issue that a large portion of systems and configurations that exhibit poor SCF convergence behavior are not well-described by single-reference methods. While it is desirable to incorporate multi-reference character into these simulations, most multireference methods such as CASSCF are not yet suitable for AIMD simulations in extreme conditions or reaction discovery settings because they are computationally expensive and require system-dependent fine-tuning of parameters such as the active space.

One promising approach to improve the accuracy of reaction discovery is to first apply CPMonitor to discover pathways at the SCF level, followed by multireference calculations to obtain more accurate estimates of reaction energies and barrier heights. The multireference data could then be deposited into community databases, such as the MolSSI QCArchive, to inform the development of future single-reference models that could more accurately approximate the multireference potential energy surface. In terms of applications, we think that BOMD simulations with CPMonitor can be used to understand reactivity in more complex systems with open mechanistic questions, such as the computational design of more efficient transition metal-containing catalysts for energy conversion or investigating the potential role of minerals in the prebiotic synthesis of the molecules of life.

6   Appendix

6.1   Calculation of $\mathbf{P}_+$ and $\mathbf{W}_+$

Here we provide additional details for how the quantities in Eqs. 8- 11 are evaluated in order to numerically integrate the dynamical variables of CPMD. The energy partial derivatives in Eqs. 8-9 are given by:

$$
\begin{aligned}
\left.\frac{\partial E(\mathbf{r}, \mathbf{P})}{\partial \mathbf{r}}\right|_{\mathbf{P}} &= \mathrm{Tr}\left[\mathbf{U}^{-T}\frac{d\mathbf{H}_0'}{d\mathbf{r}}\mathbf{U}^{-1}\widetilde{\mathbf{P}} + \frac{1}{2}\mathbf{U}^{-T}\left.\frac{\partial \mathbf{G}'}{\partial \mathbf{r}}\right|_{\mathbf{P}'}\mathbf{U}^{-1}\widetilde{\mathbf{P}}\right] \\
&\quad - \mathrm{Tr}\left[\mathbf{F}\left(\frac{d\mathbf{U}}{d\mathbf{r}}\mathbf{U}^{-1}\right)\widetilde{\mathbf{P}} + \widetilde{\mathbf{P}}\left(\mathbf{U}^{-T}\frac{d\mathbf{U}^T}{d\mathbf{r}}\right)\mathbf{F}\right] + \frac{dV_{NN}}{d\mathbf{r}} \quad (13)
\end{aligned}
$$

$$
\left.\frac{\partial E(\mathbf{r}, \mathbf{P})}{\partial \mathbf{P}}\right|_{\mathbf{r}} = 3\mathbf{FP} + 3\mathbf{PF} - 2\mathbf{FP}^2 - 2\mathbf{PFP} - 2\mathbf{P}^2\mathbf{F} \quad (14)
$$

where $\widetilde{\mathbf{P}} = 3\mathbf{P}^2 - 2\mathbf{P}^3$ is the McWeeny purification transformation of the density, and the trace is taken over electronic variables. The orthogonalization matrix derivative $d\mathbf{U}/d\mathbf{r}$ is described in Ref. 41.

Instead of finding the Lagrange multipliers explicitly, an iterative formula is used to ensure $\mathbf{P}_+$ is idempotent. We express the density matrix at the next time step as:

$$
\begin{aligned}
\mathbf{P}_+ &= \left[\mathbf{P} + \mathbf{W}\cdot\delta t - \frac{1}{2\mu}\left.\frac{\partial E(\mathbf{r}, \mathbf{P})}{\partial \mathbf{P}}\right|_{\mathbf{r}}\cdot\delta t^2\right] - \frac{1}{2\mu}\left(\mathbf{\Lambda P} + \mathbf{P\Lambda} - \mathbf{\Lambda}\right)\cdot\delta t^2 \\
&= \mathbf{P}_+^* + \mathbf{\Delta}
\end{aligned}
\quad (15)
$$

where $\mathbf{P}_+^*$ is the component of $\mathbf{P}_+$ in square brackets that may be calculated directly from Eq. 13, and $\mathbf{\Delta}$ is the term to be solved for that contains the Lagrange multipliers. Equations 16 through 19 describe the iterative procedure to solve for $\mathbf{P}_+$ and $\mathbf{\Delta}$, and $\mathbf{P}_{+,n} \equiv \mathbf{P}_+^* + \mathbf{\Delta}_n$ are their values at iteration n. The objective function being minimized is the norm-squared of the deviation from idempotency, $\mathbf{P}_+^2 - \mathbf{P}_+$:

$$
\mathrm{Tr}[(\mathbf{P}_+^2 - \mathbf{P}_+)^2] = \mathrm{Tr}[\mathbf{P}_+^4 - 2\mathbf{P}_+^3 + \mathbf{P}_+^2]. \quad (16)
$$

The gradient is given by:

$$\mathbf{\Gamma} \equiv \frac{\partial[(\mathbf{P}_+^2 - \mathbf{P}_+)^2]}{\partial \mathbf{\Delta}} = \frac{\partial[(\mathbf{P}_+^2 - \mathbf{P}_+)^2]}{\partial \mathbf{P}_+} = 4\mathbf{P}_+^3 - 6\mathbf{P}_+^2 + 2\mathbf{P}_+ \tag{17}$$

We also define $\mathbf{T} \equiv -1/2\,\mathbf{\Gamma}$ as a steepest descent optimization step scaled by $1/2$.

Using the idempotency of the density matrix at the current time step, i.e. $\mathbf{P}^2 = \mathbf{P}$, an additional condition on $\mathbf{\Delta}$ may be derived as:

$$\begin{aligned}
\mathbf{Q} \equiv \mathbf{I} - \mathbf{PP}[\mathbf{\Lambda P} + \mathbf{P\Lambda} - \mathbf{\Lambda}]\mathbf{Q} &= [\mathbf{P\Lambda PQ} + \mathbf{P}^2\mathbf{\Lambda Q} - \mathbf{P\Lambda Q}] \\
&= [\mathbf{P\Lambda P}(\mathbf{I} - \mathbf{P})] \\
&= [\mathbf{P\Lambda}(\mathbf{P}^2 - \mathbf{P})] = 0
\end{aligned} \tag{18}$$

Thus, $\mathbf{\Delta}$ should satisfy $\mathbf{P\Delta Q} = 0$. This condition is enforced by projection, and the next iterate is given in terms of the current one as:

$$\begin{aligned}
\mathbf{\Delta}_{n+1} &= \mathbf{\Delta}_n + (\mathbf{PT}_n\mathbf{P} + \mathbf{QT}_n\mathbf{Q}) \\
\mathbf{P}_{+,n+1} &= \mathbf{P}_{+,n} + (\mathbf{PT}_n\mathbf{P} + \mathbf{QT}_n\mathbf{Q})
\end{aligned} \tag{19}$$

The iterations are terminated when $\mathrm{Tr}[(\mathbf{P}_+^2 - \mathbf{P}_+)^2]$ falls below a threshold; we used the previously established convergence threshold of 10^{-12}.

After $\mathbf{P}_+$ is found, the density matrix velocities $\mathbf{W}_+$ are given by:

$$\begin{aligned}
\mathbf{W}_+ &= \left[\frac{\mathbf{P}_+ - \mathbf{P}}{\delta t} - \frac{1}{2\mu}\left.\frac{\partial E(\mathbf{r}, \mathbf{P})}{\partial \mathbf{P}}\right|_{\mathbf{r}} \cdot \delta t\right] - \frac{1}{2\mu}(\mathbf{\Lambda P} + \mathbf{P\Lambda} - \mathbf{\Lambda}) \cdot \delta t \\
&= \mathbf{W}_+^* - \frac{1}{2\mu}(\mathbf{\Lambda P} + \mathbf{P\Lambda} - \mathbf{\Lambda}) \cdot \delta t
\end{aligned} \tag{20}$$

Here the second term on the RHS is found by setting the time derivative of the idempotency condition to zero, i.e. $\mathbf{W}_+\mathbf{P}_+ + \mathbf{P}_+\mathbf{W}_+ - \mathbf{W}_+ = 0$. This allows Λ and $\mathbf{W}_+$ to be solved for

as:

$$\mathbf{\Lambda} = \frac{2\mu}{\delta t}(2\mathbf{P}_+\mathbf{W}_+^*\mathbf{P}_+ - \mathbf{W}_+^*)$$

$$\mathbf{W}_+ = \mathbf{W}_+^* - \mathbf{P}_+\mathbf{W}_+^*\mathbf{P}_+ - \mathbf{Q}_+\mathbf{W}_+^*\mathbf{Q}_+ \tag{21}$$

In the mass weighted generalization, Eq. 21 requires iterations to converge $\mathbf{W}_+$ similar to the case of Eq. 19 for $\mathbf{P}_+$. Thus, the conjugate gradient minimization described below applies equally well to accelerate the iterations to converge either $\mathbf{P}_+$ or $\mathbf{W}_+$.

6.2 Conjugate gradient minimization

Here we expand upon the discussion of Ref. 41 and introduce a new conjugate gradient formula that reaches idempotency more efficiently. The derivation is best explained by first deriving the scaling factor of $1/2$ that appears in Equation 17. To simplify notation, Equations 22 through 30 will use $\mathbf{P}$ to denote the current value of $\mathbf{P}_{+,n}$ being optimized. The following identities are noted for derivatives of powers of $\mathbf{P}$:

$$\frac{\partial(\mathbf{P}^3)_{pq}}{\partial P_{rs}} = \delta_{pr}(P^2)_{sq} + P_{pr}P_{sq} + (P^2)_{pr}\delta_{sq}$$

$$\frac{\partial(\mathbf{P}^2)_{pq}}{\partial P_{rs}} = \delta_{pr}P_{sq} + P_{pr}\delta_{qs} \tag{22}$$

$$\frac{\partial P_{pq}}{\partial P_{rs}} = \delta_{pr}\delta_{qs}$$

This provides an expression for the second derivative of $\mathrm{Tr}[(\mathbf{P}^2 - \mathbf{P})^2]$ as:

$$H_{pqrs} \equiv \frac{\partial^2}{\partial P_{pq}\partial P_{rs}}\mathrm{Tr}[(\mathbf{P}^2 - \mathbf{P})^2] = \frac{\partial}{\partial P_{rs}}(4\mathbf{P}^3 - 6\mathbf{P}^2 + 2\mathbf{P})_{pq} \approx 2(P_{pr}P_{qs} + Q_{pr}Q_{qs}) \tag{23}$$

In the last step, we obtain a simplification by assuming that $\mathbf{P}^2 - \mathbf{P} \approx 0$, i.e. $\mathbf{P}$ is nearly idempotent. With the second derivative, we may use Taylor series to estimate the step size in a steepest descent minimization. Setting the search direction $\mathbf{x} = \lambda\mathbf{\Gamma}$ where λ is to be

determined, we define the following function of λ :

$$\text{Tr}[((\mathbf{P} + \mathbf{x})^2 - (\mathbf{P} + \mathbf{x}))^2] = c + \mathbf{\Gamma} \cdot \mathbf{x} + \frac{1}{2}\mathbf{x} \cdot \mathbb{H} \cdot \mathbf{x}$$

$$f(\lambda) \equiv \text{Tr}[((\mathbf{P} + \lambda\mathbf{\Gamma})^2 - (\mathbf{P} + \lambda\mathbf{\Gamma}))^2] = c + \lambda\mathbf{\Gamma} \cdot \mathbf{\Gamma} + \frac{\lambda^2}{2}\mathbf{\Gamma} \cdot \mathbb{H} \cdot \mathbf{\Gamma} \tag{24}$$

where $c \equiv \text{Tr}[(\mathbf{P}^2 - \mathbf{P})^2]$, the idempotency error at the current iteration. To find the value of λ that minimizes the error, we find the root of $\partial f / \partial \lambda$ as:

$$\frac{\partial}{\partial \lambda}f(\lambda) = \mathbf{\Gamma} \cdot \mathbf{\Gamma} + \lambda\mathbf{\Gamma} \cdot \mathbb{H} \cdot \mathbf{\Gamma} = 0$$

$$\lambda^* = \frac{\mathbf{\Gamma} \cdot \mathbf{\Gamma}}{\mathbf{\Gamma} \cdot \mathbb{H} \cdot \mathbf{\Gamma}}. \tag{25}$$

We can evaluate the numerator and denominator as:

$$\mathbf{\Gamma} \cdot \mathbf{\Gamma} = (4\mathbf{P}^3 - 6\mathbf{P}^2 + 2\mathbf{P})_{pq}(4\mathbf{P}^3 - 6\mathbf{P}^2 + 2\mathbf{P})_{pq}$$

$$\mathbf{\Gamma} \cdot \mathbb{H} \cdot \mathbf{\Gamma} \approx (4\mathbf{P}^3 - 6\mathbf{P}^2 + 2\mathbf{P})_{pq} * 2(P_{pr}P_{qs} + Q_{pr}Q_{qs}) * (4\mathbf{P}^3 - 6\mathbf{P}^2 + 2\mathbf{P})_{rs} \tag{26}$$

If we use $\mathbf{P}^2 - \mathbf{P} \approx 0$ again, we obtain a compact expression for inner products of $\mathbb{H}$ with $\mathbf{P}$ raised to any power:

$$\mathbb{H} \cdot \mathbf{P}^n = 2\sum_{rs}(P_{pr}P_{qs} + Q_{pr}Q_{qs})(P_{rs})^n = 2\mathbf{P}\mathbf{P}^n\mathbf{P} + \mathbf{Q}\mathbf{P}^n\mathbf{Q}$$

$$= 4\mathbf{P}^{n+2} - 4\mathbf{P}^{n+1} + 2\mathbf{P}^n = 2\mathbf{P}^n(2\mathbf{P}^2 - 2\mathbf{P} + 1) \approx 2\mathbf{P}^n \tag{27}$$

Thus, we have

$$\mathbf{\Gamma} \cdot \mathbb{H} \cdot \mathbf{\Gamma} \approx 2\mathbf{\Gamma} \cdot \mathbf{\Gamma}; \quad \lambda^* \approx \frac{1}{2}. \tag{28}$$

The actual steepest descent iterations are modified from $-\mathbf{\Gamma}/2$ by projections as in Equation 19.

In the nonlinear conjugate gradient approach, iterations of $\mathbf{P}$ after the initial steepest descent step employ a modified conjugate direction $\mathbf{s}$. The Polak-Ribiere formula is used for computing the factor β:

$$\mathbf{s}_n = \mathbf{x}_n - \beta\mathbf{s}_{n-1}; \quad \beta = \mathbf{x}_n(\mathbf{x}_n - \mathbf{x}_{n-1})/|\mathbf{x}_{n-1}|^2 \,. \tag{29}$$

Next, we need to minimize the error along this search direction. Given the approximate Hessian in Eq. 23 and the second-order Taylor expansion in Eq. 24, we may express the optimal step size as:

$$\lambda = \frac{\boldsymbol{\Gamma} \cdot \mathbf{s}}{\mathbf{s} \cdot \mathbb{H} \cdot \mathbf{s}}; \quad \mathbf{s} \cdot \mathbb{H} \cdot \mathbf{s} = 2(\mathbf{s}\mathbf{P} \cdot \mathbf{P}\mathbf{s} + \mathbf{s}\mathbf{Q} \cdot \mathbf{Q}\mathbf{s}). \tag{30}$$

Thus, the conjugate gradient algorithm for iterating $\mathbf{P}_+$ toward idempotency may be summarized as:

1. Start with $n = 0$ and $\mathbf{P}_n = \mathbf{P}_+^*$ from Equation 15.

2. (Begin Loop) If $\mathrm{Tr}[(\mathbf{P}_n^2 - \mathbf{P}_n)^2] <$ threshold, terminate loop and set $\mathbf{P}_+ = \mathbf{P}_n$.

3. Compute the gradient $\boldsymbol{\Gamma}_n = 4\mathbf{P}_n^3 - 6\mathbf{P}_n^2 + 2\mathbf{P}_n$.

4. Set the projected steepest-descent direction: $\mathbf{x}_n = -(\mathbf{P}_0\boldsymbol{\Gamma}_n\mathbf{P}_0 + \mathbf{Q}_0\boldsymbol{\Gamma}_n\mathbf{Q}_0)$.

5. If $n = 1$: Set search direction $\mathbf{s}_n = \mathbf{x}_n$.

6. If $n > 1$: Calculate search direction $\mathbf{s}_n$ using Equation 29.

7. Estimate the step scaling λ using Equation 30.

8. Apply the step to the density matrix: $\mathbf{P}_{n+1} = \mathbf{P}_n + \lambda\mathbf{s}_n$

9. Increment n by 1.

10. Return to step 2. (End Loop)

In our performance tests on the malonate ion system from §4.1, the CG algorithm needed 5.1 ± 0.04 iterations to converge $\mathbf{P}$ and 7.5 ± 0.04 iterations to converge $\mathbf{W}$ to a precision of 10^{-12}. The SD algorithm needed 13.1 ± 0.2 iterations to converge $\mathbf{P}$ and 23.0 ± 0.2 iterations to converge $\mathbf{W}$ to the same precision.

6.3 Mass-weighting scheme

The basic principle of mass weighting is to assign larger masses to core orbitals, thereby forcing their coefficients to oscillate less rapidly and allowing for larger CPMD time steps. From Eq. 9, the fictitious force on the dynamical density matrix elements is proportional to the Fock matrix elements, thus we expect the most rapid oscillations to occur for the AOs that have the largest Fock matrix elements - the core orbitals. On the other hand, we also expect the core orbital occupations to have smaller variations over time compared to valence orbitals because they do not participate directly in chemical bonding and intermolecular interactions. Indeed, power spectral analysis of the time series of density matrix coefficients indicates the existence of very small, high-frequency oscillations from core orbitals that are absent from most valence orbitals (Supporting Figure S3).

Our mass-weighting scheme is based on the diagonal elements of the kinetic energy matrix rather than the full Fock matrix, which we regard as an improvement over the scheme in Ref.[42] The kinetic energy matrix has uniformly positive elements, and contains no two-electron contributions from chemical bonding that may change as reactions occur in the simulations; this contrasts with the Fock matrix whose elements vary from < 100 to > 10 a.u. and are subject to larger changes over time. We also found in preliminary tests that using the Fock matrix elements would sometimes fail to detect some density matrix elements with fast frequency components, i.e. some AOs had Fock matrix elements in the valence orbital range yet they possessed the same high-frequency components as other core orbitals. Though we could not find an underlying reason, we think it could be due to variations in the Fock matrix element over time due to changes in chemical bonding or the orthogonalization matrices.

The mass-weighting scheme used the following procedure:

$$T_i = \max\left(\{T_{jk} \mid j, k \in [1, N_{\mathrm{AO}}] \wedge (j = i \vee U_{ij} > k_U)\}\right)$$

$$\mu_i = \mu_0 * \left(1 + T_i * \frac{T_i^6}{T_i^6 + k_S^6}\right)$$

(31)

In the first line, T_i is calculated as the maximum over row i of the kinetic energy matrix, and all rows j whose orthogonalized AOs share the same non-orthogonal basis functions as function i. The latter is determined by applying a threshold k_U to the Löwdin symmetrized orthogonalization matrix $\mathbf{U}$. The second line scales up the mass of core orbitals (i.e. those with large T_i values) using a switching function. Valence orbitals with $T_i < k_S$ are given masses very close to μ_0 whereas core orbitals with $T_i > k_S$ are given masses that increase proportionally with T_i. We used parameter values of $k_S = 5.0$ a.u. and $k_U = 0.2$ throughout the study.

7 Acknowledgements

8 Supporting Information

Supporting Figures show the energy conservation of CPMD simulations, the comparison of radial distribution functions for the iron carbonyl system, and the spectral analysis of density matrix coefficients in CPMD simulations.

References

(1) Payne, M. C.; Teter, M. P.; Allan, D. C.; Arias, T. A.; Joannopoulos, J. D. Iterative minimization techniques for ab initio total-energy calculations: molecular dynamics and conjugate gradients. *Rev. Mod. Phys.* **1992**, *64*, 1045–1097.

(2) Thompson, D. L. *Modern Methods for Multidimensional Dynamics Computations in Chemistry*; WORLD SCIENTIFIC, 1998.

(3) Tuckerman, M. E. Ab initio molecular dynamics: basic concepts, current trends and novel applications. *J. Phys.-Condens. Mat.* **2002**, *14*, R1297–R1355.

(4) Leforestier, C. Classical trajectories using the full ab initio potential energy surface H+CH4→CH4+H. *J. Chem. Phys.* **1978**, *68*, 4406–4410.

(5) Ishida, K.; Morokuma, K.; Komornicki, A. The intrinsic reaction coordinate. An ab initio calculation for HNC→HCN and H+CH4→CH4+H. *J. Chem. Phys.* **1977**, *66*, 2153–2156.

(6) Warshel, A.; Karplus, M. Semiclassical trajectory approach to photoisomerization. *Chem. Phys. Lett.* **1975**, *32*, 11 – 17.

(7) McLaughlin, D. R.; Thompson, D. L. Ab initio dynamics: HeH+ + H2 → He + H3+ (C 2) classical trajectories using a quantum mechanical potential-energy surface. *J. Chem. Phys.* **1973**, *59*, 4393–4405.

(8) Chen, M.; Ko, H.-Y.; Remsing, R. C.; Calegari Andrade, M. F.; Santra, B.; Sun, Z.; Selloni, A.; Car, R.; Klein, M. L.; Perdew, J. P.; Wu, X. Ab initio theory and modeling of water. *Proc. Nat. Acad. Sci. USA* **2017**, *114*, 10846–10851.

(9) Gaigeot, M.-P.; Sprik, M. Ab Initio Molecular Dynamics Computation of the Infrared Spectrum of Aqueous Uracil. *J. Phys. Chem. B* **2003**, *107*, 10344–10358.

(10) Peters, L. D. M.; Kussmann, J.; Ochsenfeld, C. Efficient and Accurate BornOppenheimer Molecular Dynamics for Large Molecular Systems. *J. Chem. Theory Comput.* **2017**, *13*, 5479–5485.

(11) Ammal, S. C.; Yamataka, H.; Aida, M.; Dupuis, M. Dynamics-Driven Reaction Pathway in an Intramolecular Rearrangement. *Science* **2003**, *299*, 1555–1557.

(12) Saitta, A. M.; Saija, F. Miller experiments in atomistic computer simulations. *Proc. Nat. Acad. Sci. USA* **2014**, *111*, 13768–13773.

(13) Wang, L.-P.; Titov, A.; McGibbon, R.; Liu, F.; Pande, V. S.; Martinez, T. J. Discovering chemistry with an ab initio nanoreactor. *Nat. Chem.* **2014**, *6*, 1044–1048.

(14) Goldman, N.; Reed, E. J.; Fried, L. E.; Kuo, I.-F. W.; Maiti, A. Synthesis of glycine-containing complexes in impacts of comets on early Earth. *Nat. Chem.* **2010**, *2*, 949–954.

(15) Pietrucci, F.; Andreoni, W. Graph Theory Meets Ab Initio Molecular Dynamics: Atomic Structures and Transformations at the Nanoscale. *Phys. Rev. Lett.* **2011**, *107*, 085504.

(16) Rappoport, D.; Galvin, C. J.; Zubarev, D. Y.; Aspuru-Guzik, A. Complex Chemical Reaction Networks from Heuristics-Aided Quantum Chemistry. *Journal of Chemical Theory and Computation* **2014**, *10*, 897–907.

(17) Bergeler, M.; Simm, G. N.; Proppe, J.; Reiher, M. Heuristics-Guided Exploration of Reaction Mechanisms. *Journal of Chemical Theory and Computation* **2015**, *11*, 5712–5722.

(18) Suleimanov, Y. V.; Green, W. H. Automated Discovery of Elementary Chemical Reaction Steps Using Freezing String and Berny Optimization Methods. *J. Chem. Theory Comput.* **2015**, *11*, 4248–4259.

(19) Habershon, S. Automated Prediction of Catalytic Mechanism and Rate Law Using Graph-Based Reaction Path Sampling. *J. Chem. Theory Comput.* **2016**, *12*, 1786–1798.

(20) Maeda, S.; Harabuchi, Y.; Takagi, M.; Taketsugu, T.; Morokuma, K. Artificial Force Induced Reaction (AFIR) Method for Exploring Quantum Chemical Potential Energy Surfaces. *Chem. Rec.* **2016**, *16*, 2232–2248.

(21) Pendleton, I. M.; Prez-Temprano, M. H.; Sanford, M. S.; Zimmerman, P. M. Experimental and Computational Assessment of Reactivity and Mechanism in C(sp3)N Bond-Forming Reductive Elimination from Palladium(IV). *J. Am. Chem. Soc.* **2016**, *138*, 6049–6060.

(22) Schweigert, I. V.; Dunlap, B. I. Electronic structure and molecular dynamics of breaking the RONO2 bond. *J. Chem. Phys.* **2009**, *130*, 244110.

(23) Wang, L.-P.; McGibbon, R. T.; Pande, V. S.; Martinez, T. J. Automated Discovery and Refinement of Reactive Molecular Dynamics Pathways. *J. Chem. Theory Comput.* **2016**, *12*, 638–649.

(24) Mahler, A.; Janesko, B. G.; Moncho, S.; Brothers, E. N. Why are GGAs so accurate for reaction kinetics on surfaces? Systematic comparison of hybrid vs. nonhybrid DFT for representative reactions. *J. Chem. Phys.* **2017**, *146*, 234103.

(25) Pulay, P. Convergence acceleration of iterative sequences. the case of scf iteration. *Chem. Phys. Lett.* **1980**, *73*, 393 – 398.

(26) Gilbert, A. T. B.; Besley, N. A.; Gill, P. M. W. Self-Consistent Field Calculations of Excited States Using the Maximum Overlap Method (MOM). *J. Phys. Chem. A* **2008**, *112*, 13164–13171.

(27) Van Voorhis, T.; Head-Gordon, M. A geometric approach to direct minimization. *Mol. Phys.* **2002**, *100*, 1713–1721.

(28) Kudin, K. N.; Scuseria, G. E.; Cancs, E. A black-box self-consistent field convergence algorithm: One step closer. *J. Chem. Phys.* **2002**, *116*, 8255–8261.

(29) Hu, X.; Yang, W. Accelerating self-consistent field convergence with the augmented RoothaanHall energy function. *J. Chem. Phys.* **2010**, *132*, 054109.

(30) Cancs, E.; Le Bris, C. Can we outperform the DIIS approach for electronic structure calculations? *Int. J. Quantum Chem.* **2000**, *79*, 82–90.

(31) Saunders, V. R.; Hillier, I. H. A LevelShifting method for converging closed shell HartreeFock wave functions. *Int. J. Quantum Chem.* **1973**, *7*, 699–705.

(32) Rabuck, A. D.; Scuseria, G. E. Improving self-consistent field convergence by varying occupation numbers. *J. Chem. Phys.* **1999**, *110*, 695–700.

(33) Kollmar, C. Convergence optimization of restricted open-shell self-consistent field calculations. *Int. J. Quantum Chem.* **1997**, *62*, 617–637.

(34) Car, R.; Parrinello, M. Unified Approach for Molecular Dynamics and Density-Functional Theory. *Phys. Rev. Lett.* **1985**, *55*, 2471–2474.

(35) Galli, G.; Martin, R. M.; Car, R.; Parrinello, M. Carbon: The nature of the liquid state. *Phys. Rev. Lett.* **1989**, *63*, 988–991.

(36) Laasonen, K.; Pasquarello, A.; Car, R.; Lee, C.; Vanderbilt, D. Car-Parrinello molecular dynamics with Vanderbilt ultrasoft pseudopotentials. *Phys. Rev. B* **1993**, *47*, 10142–10153.

(37) Laasonen, K.; Sprik, M.; Parrinello, M.; Car, R. Ab initio liquid water. *J. Chem. Phys.* **1993**, *99*, 9080–9089.

(38) van der Weide, J.; Zhang, Z.; Baumann, P. K.; Wensell, M. G.; Bernholc, J.; Nemanich, R. J. Negative-electron-affinity effects on the diamond (100) surface. *Phys. Rev. B* **1994**, *50*, 5803–5806.

(39) Kuo, I.-F. W.; Mundy, C. J. An ab Initio Molecular Dynamics Study of the Aqueous Liquid-Vapor Interface. *Science* **2004**, *303*, 658–660.

(40) Hartke, B.; Carter, E. A. Spin eigenstate-dependent HartreeFock molecular dynamics. *Chem. Phys. Lett.* **1992**, *189*, 358 – 362.

(41) Schlegel, H. B.; Millam, J. M.; Iyengar, S. S.; Voth, G. A.; Daniels, A. D.; Scuseria, G. E.; Frisch, M. J. Ab initio molecular dynamics: Propagating the density matrix with Gaussian orbitals. *J. Chem. Phys.* **2001**, *114*, 9758–9763.

(42) Iyengar, S. S.; Schlegel, H. B.; Millam, J. M.; Voth, G. A.; Scuseria, G. E.; Frisch, M. J. Ab initio molecular dynamics: Propagating the density matrix with Gaussian orbitals. II. Generalizations based on mass-weighting, idempotency, energy conservation and choice of initial conditions. *J. Chem. Phys.* **2001**, *115*, 10291–10302.

(43) Herbert, J. M.; Head-Gordon, M. Curvy-steps approach to constraint-free extended-Lagrangian ab initio molecular dynamics, using atom-centered basis functions: Convergence toward BornOppenheimer trajectories. *The Journal of Chemical Physics* **2004**, *121*, 11542–11556.

(44) Schwegler, E.; Grossman, J. C.; Gygi, F.; Galli, G. Towards an assessment of the accuracy of density functional theory for first principles simulations of water. II. *J. Chem. Phys.* **2004**, *121*, 5400–5409.

(45) Grossman, J. C.; Schwegler, E.; Draeger, E. W.; Gygi, F.; Galli, G. Towards an assessment of the accuracy of density functional theory for first principles simulations of water. *J. Chem. Phys.* **2004**, *120*, 300–311.

(46) Niklasson, A. M. N. Extended Born-Oppenheimer Molecular Dynamics. *Phys. Rev. Lett.* **2008**, *100*, 123004.

(47) Khne, T. D. Second generation CarParrinello molecular dynamics. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **2014**, *4*, 391–406.

(48) Niklasson, A. M. N. Next generation extended Lagrangian first principles molecular dynamics. *J. Chem. Phys.* **2017**, *147*, 054103.

(49) Xiao, H.; Tahir-Kheli, J.; Goddard, W. A. Accurate Band Gaps for Semiconductors from Density Functional Theory. *The Journal of Physical Chemistry Letters* **2011**, *2*, 212–217.

(50) Schlegel, H. B.; Iyengar, S. S.; Li, X.; Millam, J. M.; Voth, G. A.; Scuseria, G. E.; Frisch, M. J. Ab initio molecular dynamics: Propagating the density matrix with Gaussian orbitals. III. Comparison with BornOppenheimer dynamics. *J. Chem. Phys.* **2002**, *117*, 8694–8704.

(51) 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*, 222–231.

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

(53) 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*, 2619–2628.

(54) Song, C.; Wang, L.-P.; Sachse, T.; Prei, J.; Presselt, M.; Martnez, T. J. Efficient implementation of effective core potential integrals and gradients on graphical processing units. *J. Chem. Phys.* **2015**, *143*, 014114.

(55) Song, C.; Wang, L.-P.; Martnez, T. J. Automated Code Engine for Graphical Processing Units: Application to the Effective Core Potential Integrals and Gradients. *J. Chem. Theory Comput.* **2016**, *12*, 92–106.

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

(57) Rail, M. D.; Berben, L. A. Directing the Reactivity of [HFe4N(CO)12] toward H+ or CO2 Reduction by Understanding the Electrocatalytic Mechanism. *J. Am. Chem. Soc.* **2011**, *133*, 18577–18579.

(58) Jang, H.; Qiu, Y.; Hutchings, M. E.; Nguyen, M.; Berben, L. A.; Wang, L.-P. Quantum chemical studies of redox properties and conformational changes of a four-center iron CO2 reduction electrocatalyst. *Chem. Sci.* **2018**, *9*, 2645–2654.

(59) Wang, H.; Xie, Y.; King, R. B.; Schaefer, H. F. Remarkable Aspects of Unsaturation in Trinuclear Metal Carbonyl Clusters: The Triiron Species Fe3(CO)n (n = 12, 11, 10, 9). *J. Am. Chem. Soc.* **2006**, *128*, 11376–11384.