Self-consistent predictor/corrector algorithms for stable and efficient integration of the time-dependent Kohn-Sham equation

Ying Zhu and John M. Herbert[a)]
Department of Chemistry and Biochemistry, and Chemical Physics Program, The Ohio State University, Columbus, Ohio 43210, USA

(Received 14 September 2017; accepted 15 January 2018; published online 31 January 2018)

The "real time" formulation of time-dependent density functional theory (TDDFT) involves integration of the time-dependent Kohn-Sham (TDKS) equation in order to describe the time evolution of the electron density following a perturbation. This approach, which is complementary to the more traditional linear-response formulation of TDDFT, is more efficient for computation of broad-band spectra (including core-excited states) and for systems where the density of states is large. Integration of the TDKS equation is complicated by the time-dependent nature of the effective Hamiltonian, and we introduce several predictor/corrector algorithms to propagate the density matrix, one of which can be viewed as a self-consistent extension of the widely used modified-midpoint algorithm. The predictor/corrector algorithms facilitate larger time steps and are shown to be more efficient despite requiring more than one Fock build per time step, and furthermore can be used to detect a divergent simulation on-the-fly, which can then be halted or else the time step modified. *Published by AIP Publishing.* https://doi.org/10.1063/1.5004675

I. INTRODUCTION

The development of computational methods to study excited states and electron dynamics under external electromagnetic fields is essential for mechanistic studies of many physico-chemical processes. Weak-field processes include generation of photo-lesions in DNA[1] and light harvesting in dye sensitized solar cells,[2,3] both of which involve photo-excitation, charge transfer, and photo-ionization processes. Strong-field processes include high harmonic generation,[4] multi-photon ionization,[5] and above-threshold ionization.[6] The emergence of attosecond spectroscopy makes probing electronic motions possible,[7,8] and experiments can now directly probe electron dynamics to study bond breaking,[9] ionization,[10,11] and electron solvation.[12,13] Development of improved computational methods to complement these experiments is essential.

Time-dependent density functional theory (TDDFT)[14,15] is the presently the most popular method for computing excited states and response properties, due to its reasonable tradeoff between accuracy and efficiency.[16] The most widely used and implemented version of TDDFT is the linear response (LR) approach based on Casida's equation,[17] but while LR-TDDFT has proven to be an economical tool to study a few low-lying excited states of an isolated molecule, it is limited to the weak-field perturbative regime and cannot be used to study strong-field processes[18,19] or non-linear optical properties.[20–22] Furthermore, Casida's equation is solved iteratively in a basis of (occupied) × (virtual) dimension,[23] which is suitable for computing a few low-lying excited states but becomes prohibitive in both memory and computer time when the density of states is high, e.g., in a semiconductor,[24] or when core-excited states or broad-band spectra are of interest.[25]

An alternative to solving Casida's equation in the frequency domain is to solve the time-dependent Kohn-Sham (TDKS) equation in the time domain, in what has been dubbed the "real-time" (RT) approach to TDDFT.[26–30] This approach is intrinsically non-perturbative and is thus suitable for both weak and strong fields. In the weak-field limit, RT-TDDFT and LR-TDDFT results should be equivalent. Unlike the LR approach, where the energy spectrum is obtained iteratively starting with the lowest excited state, in RT-TDDFT the entire broad-band spectrum is obtained at once, with resolution that improves upon further time propagation. The RT approach has been implemented in various DFT codes including Octopus,[31] Siesta,[20] Qbox,[32] Gaussian,[33,34] NWChem,[35] and Q-Chem,[36] and has been formulated using real-space grids,[20,31] plane waves,[32] and Gaussian-type orbitals.[33–36] We focus exclusively on Gaussian-type orbitals in this work.

Both the accuracy and efficiency of a RT-TDDFT calculation depend critically on the integrator that is used to propagate the density based on the TDKS equation. It is *electron* dynamics that is propagated, and often the time step Δt for this propagation is as small as 0.05 a.u. ($\approx$1 as),[36–38] although somewhat larger time steps (as large as 0.5 a.u.) have been used in some applications.[35,39–41] The numerical validity must always be carefully checked during the calculation. In calculations based on Gaussian orbitals, which are the exclusive basis sets examined here, time integration involves evaluation of the Fock matrix at every time step, for a simulation that might propagate for tens of femtoseconds at a minimum. Thus, while the RT approach avoids the memory bottleneck of the LR approach, requiring only about twice the memory footprint of the ground-state DFT calculation (because the

[a)] herbert@chemistry.ohio-state.edu

Fock and density matrices become complex), the numerous "Fock builds" may increase the computational time.

In the present work, we introduce two predictor/corrector (PC) methods that significantly improve both the efficiency of time propagation (as characterized by the number of Fock builds per unit of simulated time) and the stability of the numerical integration. One of these algorithms can be viewed as a self-consistent extension of the modified-midpoint algorithm of Li *et al.*,[33] and the purpose of the self-consistent PC design is two-fold. First, the self-consistent step between the predicted and corrected values of the density matrix serves as a self-checking procedure that can be used to halt a divergent calculation immediately so that computational time is not wasted on an inappropriate calculation. In principle, adaptive time-step controls could also be incorporated based on this self-checking, though we have not done so here. Second, through exact evaluation of the matrix exponentials that appear in the propagator algorithm, which is possible when using Gaussian orbitals but likely is not possible when using plane waves or real-space grids, we rule out errors generated by this step and therefore gain a better view of how error propagates via the integrator itself.

The remainder of this paper is structured as follows. In Sec. II, we briefly review the TDKS formalism and then detail the PC algorithms and discuss their mathematical convergence properties. Section III provides numerical results that compare the accuracy and stability of different algorithms, including examination of energy and density conservation, and comparison of excitation spectra.

II. THEORY

A. TDKS equation

In TDDFT, the time-dependent electron density $\rho(\mathbf{r}, t)$ is obtained by solving the TDKS equation. In atomic units (a.u.), this equation is

$$-\tfrac{1}{2}\hat{\nabla}^2\psi_k(\mathbf{r},t) + \upsilon_{\mathrm{KS}}(\mathbf{r},t)\,\psi_k(\mathbf{r},t) = i\frac{\partial}{\partial t}\psi_k(\mathbf{r},t), \tag{1}$$

where the functions ψ_k are time-dependent molecular orbitals (MOs) for a fictitious, non-interacting reference system. The Kohn-Sham potential

$$\upsilon_{\mathrm{KS}}(\mathbf{r},t) = \upsilon_{\mathrm{ext}}(\mathbf{r},t) + \int d\mathbf{r}'\,\frac{\rho(\mathbf{r}',t)}{|\mathbf{r}-\mathbf{r}'|} + \upsilon_{\mathrm{xc}}(\mathbf{r},t) \tag{2}$$

includes the external potential (υ_{ext}), the Hartree or mean-field Coulomb potential, and the exchange-correlation potential (υ_{xc}).

We assume that the exact density is representable in terms of a single Slater determinant,

$$\rho(\mathbf{r},t) = \sum_k^{\mathrm{occ}} |\psi_k(\mathbf{r},t)|^2 \ . \tag{3}$$

The time-dependent MOs can be expanded in a fixed basis $\{\psi_k^0(\mathbf{r})\}$ provided by the ground-state Kohn-Sham orbitals, with time-dependent coefficients,

$$\psi_k(\mathbf{r},t) = \sum_j C_{jk}(t)\,\psi_j^0(\mathbf{r}). \tag{4}$$

The sum in this equation runs over both occupied and virtual MOs of the ground-state calculation, regardless of whether $\psi_k(\mathbf{r}, t)$ is an occupied or a virtual orbital. (Only the occupied orbitals must be propagated in time, however.) Fixing the $\{\psi_k^0(\mathbf{r})\}$ basis, Eq. (1) becomes

$$\sum_n F_{jn}C_{nk}(t) = i\frac{\partial}{\partial t}C_{jk}(t), \tag{5}$$

where $F_{jk} = \langle\psi_j^0|\hat{f}_{\mathrm{KS}}|\psi_k^0\rangle$ and

$$\hat{f}_{\mathrm{KS}} = -\tfrac{1}{2}\hat{\nabla}^2 + \upsilon_{\mathrm{KS}}(\mathbf{r},t) \tag{6}$$

is the Fock operator. The time-dependent density is thus

$$\rho(\mathbf{r},t) = \sum_{jk} P_{kj}(t)\,\psi_k^0(\mathbf{r})\,\psi_j^0(\mathbf{r})^*, \tag{7}$$

where $\mathbf{P} = \mathbf{C}_{\mathrm{occ}}\mathbf{C}_{\mathrm{occ}}^\dagger$ is the one-electron density matrix, with matrix elements

$$P_{jk}(t) = \sum_n^{\mathrm{occ}} C_{jn}(t)\,C_{kn}(t)^* \ . \tag{8}$$

Henceforth, we assume that time propagation starts from the ground state and that the MOs are represented as linear combinations of atomic orbitals (AOs). The initial density matrix at $t = 0$ is

$$\mathbf{P}(0) = \mathrm{diag}\Big(\overbrace{\underbrace{1,1,1,\ldots,1}_{\text{occupied}},\underbrace{0,0,\ldots,0}_{\text{virtual}}}^{\text{MOs}}\Big) \tag{9}$$

in the MO basis $\{\psi_k^0\}$.

Equation (5) can be simplified into a form of the Liouville-von Neumann (LvN) equation,

$$i\frac{\partial\mathbf{P}}{\partial t} = \mathbf{FP} - \mathbf{PF}, \tag{10}$$

whose solution $\mathbf{P}(t)$ can be written, formally, using a unitary propagator

$$\mathbf{U}(t_2,t_1) = \hat{\mathcal{T}}\exp\left(-i\int_{t_1}^{t_2}\mathbf{F}(t)\,dt\right), \tag{11}$$

whose definition involves time-ordered integration,[42] indicated by $\hat{\mathcal{T}}$ in Eq. (11). The density matrix at time t is

$$\mathbf{P}(t) = \mathbf{U}(t,0)\,\mathbf{P}(0)\,[\mathbf{U}(t,0)]^\dagger \ . \tag{12}$$

Unitarity of the propagator ensures that the density matrix satisfies the properties $\mathbf{P}^2 = \mathbf{P}$, $\mathbf{P}^\dagger = \mathbf{P}$, and $\mathrm{tr}(\mathbf{P}) = n_{\mathrm{e}}$ (the number of electrons) at each moment in time, provided that the density matrix satisfies these properties at $t = 0$.

B. Time integration of the propagator

The LvN equation is a "stiff" differential equation. Once the system is excited from the ground state, matrix elements of $\mathbf{P}(t)$ oscillate at a broad range of frequencies. The Fourier components of these oscillations correspond to the Bohr frequencies, including both valence-to-valence and core-to-valence excitations. To maintain numerical stability, Eq. (10) must therefore be integrated using a very small time step, especially if Taylor-expansion-based propagation algorithms (e.g., Euler, Runge-Kutta, Crank-Nicolson, etc.[43,44])

are employed. If excitations out of core orbitals or other high energy excitations are not of interest, then a frozen core approximation,[45] or else a more general active-space implementation,[34] can be used remove the highest-frequency components. On the other hand, these high-lying excitations are actually attractive targets for a real-time implementation due to the iterative nature of LR-TDDFT algorithms, whose cost scales with the number of excited states that are needed and thus becomes very expensive for high-energy excitations.

We will not use frozen orbitals in this work and will employ propagators based on approximations to Eq. (11). Note that the integration in this equation is non-trivial because $\mathbf{F}(t)$ does not commute with $\mathbf{F}(t')$ for $t \neq t'$. This fact itself necessitates the use of a small time step, stiffness of the equation notwithstanding. Let $t_N = N\Delta t$ denote the value of t at the Nth time step, and for simplicity denote by $\mathbf{U}_N$ the propagator from t_N to t_{N+1},

$$\mathbf{U}_N = \mathbf{U}(t_{N+1}, t_N) = \hat{\mathcal{T}}\exp\left(-i\int_{t_N}^{t_{N+1}} \mathbf{F}(t)\,dt\right) , \qquad (13)$$

which takes $\mathbf{P}_N$ to $\mathbf{P}_{N+1}$. In detail,

$$\mathbf{P}_{N+1} = \mathbf{U}_N\mathbf{U}_{N-1}\cdots\mathbf{U}_1\mathbf{U}_0\mathbf{P}_0\mathbf{U}_0^\dagger\mathbf{U}_1^\dagger\cdots\mathbf{U}_{N-1}^\dagger\mathbf{U}_N^\dagger . \qquad (14)$$

Discretizing time allows us to evaluate the integral in Eq. (13) and to ignore the time-ordering, with the caveat that the exact time-ordered propagator is obtained only in the limit $\Delta t \to 0$.

A second consideration is the exponential in Eq. (13). Unitarity of the propagator is essential to preserving the trace and idempotency of the density matrix. If the dimension of $\mathbf{U}$ is large, as in a plane-wave or grid-based calculation, then the matrix exponential can only be approximated, using methods such as polynomial expansions or subspace projection.[46] This introduces additional error in each integration step. When Gaussian orbitals are used, however, the dimension of $\mathbf{U}$ is relatively small and exact evaluation of the exponential (by diagonalization) incurs negligible cost as compared to a single Fock build. As such, we always compute exact matrix exponentials.

C. Modified midpoint algorithm

Schlegel and co-workers[33] introduced the modified-midpoint unitary transform (MMUT) time-propagation algorithm that was mentioned in Sec. I and which is based on the approximation

$$\mathbf{U}_N \approx \exp(-i\Delta t\, \mathbf{F}_{N+1/2}), \qquad (15)$$

in conjunction with an exact evaluation of the matrix exponential. Although Eq. (15) only approximates $\mathbf{U}_N$, this approximate propagator is unitary by construction. A diagram of a MMUT time step appears in Fig. 1. Starting at time t_N, with initial inputs being the density matrix at the previous half time step ($\mathbf{P}_{N-1/2}$) and at the current time ($\mathbf{P}_N$), $\mathbf{F}_N$ is first constructed based on $\mathbf{P}_N$. Next, $\mathbf{P}_{N-1/2}$ is propagated to $\mathbf{P}_{N+1/2}$ via midpoint approximation, i.e.,

$$\mathbf{P}_{N+1/2} = (e^{-i\Delta t\mathbf{F}_N})\mathbf{P}_{N-1/2}(e^{i\Delta t\mathbf{F}_N}). \qquad (16)$$

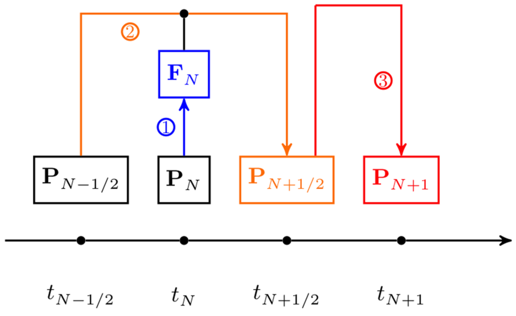

FIG. 1. Flow chart for the MMUT method. Given initial values of $\mathbf{P}_{N-1/2}$ and $\mathbf{P}_N$, the quantity $\mathbf{F}_N$ is constructed and $\mathbf{P}_{N+1/2}$ is calculated as a full time step propagation from $\mathbf{P}_{N-1/2}$, using the midpoint Fock matrix $\mathbf{F}_N$ [Eq. (16)]. Finally, $\mathbf{P}_{N+1}$ is obtained as a half time step propagation from $\mathbf{P}_{N+1/2}$, again using $\mathbf{F}_N$ [Eq. (17)].

Finally, $\mathbf{P}_{N+1}$ is calculated according to

$$\mathbf{P}_{N+1} = (e^{-(i\Delta t/2)\mathbf{F}_N})\mathbf{P}_{N+1/2}(e^{(i\Delta t/2)\mathbf{F}_N}). \qquad (17)$$

The MMUT algorithm is actually a modified explicit Euler method, where "explicit" means that the algorithm employs only that information that is available at the current and previous times. The Euler method is used in the exponential parts. The stability of this algorithm can be improved by switching from an explicit to an implicit scheme, wherein one iteratively solves equations that depend on both current and future information. In the following, we illustrate several different implicit predictor/corrector (PC) algorithms for time propagation of $\mathbf{P}(t)$.

D. Linear Fock, linear density (LFLP) PC algorithms

The first of our proposed PC algorithms is illustrated schematically in Fig. 2 and detailed in Algorithm 1. It is based on the assumption that when Δt is small, changes in the Fock and density matrices can be obtained by linear extrapolation. Starting from t_N with initial information $\mathbf{F}_{N-1/2}$, $\mathbf{F}_N$, and $\mathbf{P}_N$, the algorithm consists of the following steps, which are labeled by number in Fig. 2.

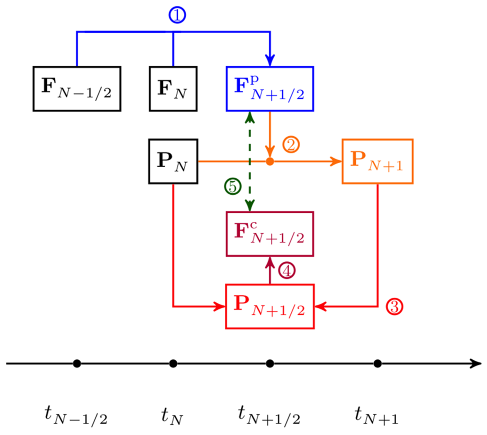

FIG. 2. Flow chart for the LFLP-PC propagator algorithm. The predicted Fock matrix $\mathbf{F}^{\rm p}_{N+1/2}$ is obtained by linear extrapolation of initial values of $\mathbf{F}_{N-1/2}$ and $\mathbf{F}_N$ and a corrected version is constructed based on a linear interpolation of the density matrix. Predicted and corrected values of the Fock matrix must reach consistency before proceeding to the next time step.

Algorithm 1. LFLP-PC.

```
1:  Inputs:
        $\mathbf{P}_0 \leftarrow$ ground state DFT result
        $\mathbf{F}_0 \leftarrow$ calculated by using $\mathbf{P}_0$
        $t_{\text{initial}}, t_{\text{final}}, \Delta t$
2:  Initialize:
        $N_{\text{f}} \leftarrow (t_{\text{final}} - t_{\text{initial}})/\Delta t$
        $\mathbf{F}_{-1/2} \leftarrow \mathbf{F}_0$
3:  for step $N = 1$ to $N_{\text{f}}$ do                              ▷ Propagation loop
4:      $\mathbf{F}^{\text{p}}_{N+1/2} \leftarrow 2\mathbf{F}_N - \mathbf{F}_{N-1/2}$                           ▷ Step 1
5:      $\mathbf{U}_N \leftarrow \exp(-i\Delta t\, \mathbf{F}^{\text{p}}_{N+1/2})$                        ▷ Step 2
6:      $\mathbf{P}_{N+1} \leftarrow \mathbf{U}_N \mathbf{P}_N \mathbf{U}^{\dagger}_N$
7:      $\boldsymbol{P}_{N+1/2} \leftarrow (\boldsymbol{P}_N + \boldsymbol{P}_{N+1})/2$                        ▷ Step 3
8:      $\mathbf{F}^{\text{c}}_{N+1/2} \leftarrow$ calculated by using $\mathbf{P}_{N+1/2}$              ▷ Step 4
9:      $\delta\mathbf{F}_{N+1/2} \leftarrow \mathbf{F}^{p}_{N+1/2} - \mathbf{F}^{\text{c}}_{N+1/2}$                   ▷ Step 5
10:     if $\|\delta\mathbf{F}_{N+1/2}\|_{\text{F}} > \xi\alpha n$ then                  ▷ not consistent
11:         $\mathbf{F}^{\text{p}}_{N+1/2} \leftarrow \mathbf{F}^{\text{c}}_{N+1/2}$
12:         go to line 5 (Step 2)
13:     else
14:         switch method do                                       ▷ consistent
15:             case LFLP-PC1
16:                 $\mathbf{F}_{N+1/2} \leftarrow \mathbf{F}^{\text{c}}_{N+1/2}$
17:             case LFLP-PC2
18:                 $\mathbf{F}_{N+1/2} \leftarrow (\mathbf{F}^{\text{p}}_{N+1/2} + \mathbf{F}^{\text{c}}_{N+1/2})/2$
19:             case LFLP-PC3
20:                 $\mathbf{F}_{N+1/2} \leftarrow (\mathbf{F}^{\text{p[0]}}_{N+1/2} + \mathbf{F}^{\text{c}}_{N+1/2})/2$
21:     end if
22: end for
```

Step 1: Predict $\mathbf{F}^{\text{p}}_{N+1/2}$ by linear extrapolation,

$$\mathbf{F}^{\text{p}}_{N+1/2} - \mathbf{F}_N = \mathbf{F}_N - \mathbf{F}_{N-1/2}. \tag{18}$$

Step 2: Propagate $\mathbf{P}_N$ to $\mathbf{P}_{N+1}$ using

$$\mathbf{U}_N \approx \exp(-i\Delta t\, \mathbf{F}^{\text{p}}_{N+1/2}). \tag{19}$$

This is the same as the MMUT update in Eq. (15).

Step 3: The density matrix at the midpoint, $\mathbf{P}_{N+1/2}$, is obtained by (backwards) linear interpolation.

Step 4: A corrected Fock matrix for the midpoint, $\mathbf{F}^{\text{c}}_{N+1/2}$, is constructed using $\mathbf{P}_{N+1/2}$.

Step 5: Evaluate the difference between $\mathbf{F}^{\text{p}}_{N+1/2}$ and $\mathbf{F}^{\text{c}}_{N+1/2}$. If consistency has not been achieved, update the former and return to step 2.

In the final step of this procedure, we evaluate consistency in terms of the Frobenius norm,

$$\|\mathbf{A}\|_{\text{F}} = [\text{tr}(\mathbf{A}\mathbf{A}^{\dagger})]^{1/2} = \Bigg(\sum_{jk} A^2_{jk}\Bigg)^{1/2}. \tag{20}$$

Specifically, if $\mathbf{A}_1$ and $\mathbf{A}_2$ are two $n \times n$ matrices that we wish to compare, then we judge them to be numerically equivalent if

$$\frac{\|\mathbf{A}_1 - \mathbf{A}_2\|_{\text{F}}}{n\alpha} < \xi, \tag{21}$$

where ξ is a specified threshold and α is the largest eigenvalue of $\mathbf{A}_1$.

We designate the approach outlined above as the "linear Fock, linear density predictor/corrector" (LFLP-PC) method. Even within this paradigm, however, there are several possible ways to update $\mathbf{F}^{\text{p}}_{N+1/2}$ in the final step, if self-consistency has not been achieved. We will examine the stability and efficiency of three possible updates, leading to three possible algorithms.

- LFLP-PC1: Use the latest value of the corrector matrix, $\mathbf{F}^{\text{c}}_{N+1/2}$.
- LFLP-PC2: Use the average of the last two approximations to $\mathbf{F}_{N+1/2}$.
- LFLP-PC3: Use the average of the latest $\mathbf{F}^{\text{c}}_{N+1/2}$ and the initial prediction $\mathbf{F}^{\text{p}}_{N+1/2}$. We denote this average as $\mathbf{F}^{\text{p[0]}}_{N+1/2}$.

The efficiency and stability of each will be examined below.

E. Exponential density predictor/corrector (EP-PC) algorithms

An alternative to linear extrapolation of $\mathbf{P}$ is to use strictly exponential updates of the density matrix, as shown schematically in Fig. 3 and detailed in Algorithm 2. Starting from t_N, with initial information $\mathbf{P}_N$, the algorithm proceeds as follows.

Step 1: Build $\mathbf{F}_N$ using $\mathbf{P}_N$.

Step 2: Obtain $\mathbf{P}^{\text{p}}_{N+1}$ by using $\mathbf{F}_N$ according to

$$\mathbf{P}^{\text{p}}_{N+1} = (e^{-i\Delta t\,\mathbf{F}_N})\,\mathbf{P}_N\,(e^{i\Delta t\,\mathbf{F}_N}) \tag{22}$$

Step 3: Build $\mathbf{F}^{\text{p}}_{N+1}$ using $\mathbf{P}^{\text{p}}_{N+1}$.

Step 4: Build the propagator by using the trapezoidal rule to perform the integral in Eq. (11),

$$\mathbf{U}_N = \exp[-(i\Delta t/2)(\mathbf{F}_N + \mathbf{F}^{\text{p}}_{N+1})], \tag{23}$$

and then update the corrected density matrix, $\mathbf{P}^{\text{c}}_{N+1} = \mathbf{U}_N \mathbf{P} \mathbf{U}^{\dagger}_N$.

Step 5: Check the consistency between $\mathbf{P}^{\text{p}}_{N+1}$ with $\mathbf{P}^{\text{c}}_{N+1}$. If consistency is reached, proceed to the next time step, otherwise substitute $\mathbf{P}^{\text{p}}_{N+1}$ with $\mathbf{P}^{\text{c}}_{N+1}$ and return to step 3.

Consistency checking for $\mathbf{P}$ uses the same procedure as that used to check the consistency of $\mathbf{F}$ in the LFLP-PC algorithms [Eq. (21)], except that in the present case $\alpha = 1$, as that is the maximum eigenvalue of $\mathbf{P}$.

As with LFLP-PC, we will examine three variants of EP-PC. These differ in how we choose $\mathbf{P}_{N+1}$ in step 5 of the

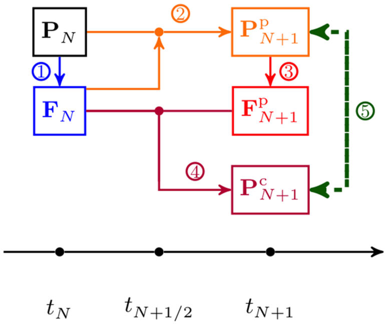
$\mathbf{P}_N$
①
$\mathbf{F}_N$
②
$\mathbf{P}^{\text{p}}_{N+1}$
③
$\mathbf{F}^{\text{p}}_{N+1}$
④
$\mathbf{P}^{\text{c}}_{N+1}$
⑤
t_N
$t_{N+1/2}$
t_{N+1}

FIG. 3. Flow chart for the EP-PC propagator algorithm. The density matrix $\mathbf{P}$ is predicted exponentially, and the propagator is approximated using the trapezoidal rule. Predicted and corrected density matrices must reach consistency before proceeding to the next time step.

Algorithm 2. EP-PC.

```
1:  Inputs:
        $\mathbf{P}_0 \leftarrow$ ground state DFT result
        $t_{\text{initial}}, t_{\text{final}}, \Delta t$
2:  Initialize:
        $N_{\text{f}} \leftarrow (t_{\text{final}} - t_{\text{initial}})/\Delta t$
3:  for step $N = 1$ to $N_{\text{f}}$ do                          ▷ Propagation loop
4:      build $\mathbf{F}_N$ using $\mathbf{P}_N$                          ▷ Step 1
5:      $\mathbf{P}^{\text{p}}_{N+1} \leftarrow \exp(-i\Delta t\mathbf{F}_N)\mathbf{P}_N\exp(i\Delta t\mathbf{F}_N)$       ▷ Step 2
6:      build $\mathbf{F}^{\text{p}}_{N+1}$ using $\mathbf{P}^{\text{p}}_{N+1}$                ▷ Step 3
7:      $\mathbf{U}_N \leftarrow \exp[-i(\Delta t/2)(\mathbf{F}_N + \mathbf{F}^{\text{p}}_{N+1})]$            ▷ Step 4
8:      $\mathbf{P}^c_{N+1} \leftarrow \mathbf{U}_N\mathbf{P}_N\mathbf{U}^\dagger_N$
9:      $\delta\mathbf{P}_{N+1} \leftarrow \mathbf{P}^p_{N+1} - \mathbf{P}^{\text{c}}_{N+1}$                    ▷ Step 5
10:     if $\|\delta\mathbf{P}_{N+1}\|_{\text{F}} > \xi\alpha n$ then                  ▷ PC not converged
11:         $\mathbf{P}^{\text{p}}_{N+1} \leftarrow \mathbf{P}^{\text{c}}_{N+1}$
12:         go to line 6 (Step 3)
13:     else                                               ▷ PC converged
14:         switch method do
15:             case EP-PC1
16:                 $\mathbf{P}_{N+1} \leftarrow \mathbf{P}^{\text{c}}_{N+1}$
17:             case EP-PC2
18:                 $\mathbf{P}_{N+1} \leftarrow (\mathbf{P}^{\text{p}}_{N+1} + \mathbf{P}^{\text{c}}_{N+1})/2$
19:             case EP-PC3
20:                 $\mathbf{P}_{N+1} \leftarrow (\mathbf{P}^{\text{p}[0]}_{N+1} + \mathbf{P}^{\text{c}}_{N+1})/2$
21:     end if
22: end for
```

algorithm, before proceeding to the next time step. The three alternatives are as follows:

- EP-PC1: Use the latest corrector matrix, $\mathbf{P}^{\text{c}}_{N+1}$.
- EP-PC2: Use the average of the two most recent values of $\mathbf{P}^{\text{c}}_{N+1}$.
- EP-PC3: Use the average of the last $\mathbf{P}^{\text{c}}_{N+1}$ and the initial $\mathbf{P}^{\text{p}}_{N+1}$, which is denoted as $\mathbf{P}^{\text{p}[0]}_{N+1}$.

F. Mathematical convergence properties of the PC methods

The iterative step that checks for consistency between the predicted and corrected values of either $\mathbf{F}_N$ or $\mathbf{P}_N$ (i.e., step 5 in Figs. 1 and 2) is critical for achieving stable and efficient simulations. This process ensures that the propagation going forward is close enough to the one going backward during one time step and thus acts as an on-the-fly self-check that monitors whether the whole calculation is well-behaved. Convergence of the PC iterations is not guaranteed and may either fail or else require numerous iterations to converge. As such, it is useful to explore the conditions under which one might expect good convergence behavior.

As shown in the Appendix, one can derive an upper bound on the largest time step that will lead to convergence of the PC iterations, namely,

$$\Delta t < \frac{2}{\left\| \left[\mathbf{P}_N, \hat{\boldsymbol{\nabla}}_{\mathbf{P}}\, \mathbf{F}\big|_{\mathbf{P}\to\mathbf{P}^{\star}_{N+1}} \right] \right\|_{\text{F}}}. \tag{24}$$

Here, $\mathbf{P}_N$ is the current density matrix, $\mathbf{P}^{\star}_{N+1}$ is the true value converged after numerous iterations, and $\hat{\boldsymbol{\nabla}}_{\mathbf{P}}\, \mathbf{F}|_{\mathbf{P}\to\mathbf{P}^{\star}_{N+1}}$ is the directional derivative of the Fock matrix with respect to the density matrix along a direction from $\mathbf{P}_N$ to $\mathbf{P}^{\star}_{N+1}$. The magnitude of the denominator in Eq. (24) also affects the rate of convergence of the PC iterations.

This convergence condition provides information about how the PC methods behave. When Δt is small and well below the upper bound in Eq. (24), the rate of convergence is fast and only one or two iterations are needed. More iterations are required when Δt is larger, and if the time step is too large (beyond the upper bound), the PC iterations will fail to converge and the calculation is no longer valid. Somewhere in between, there exists an optimal value of Δt that maximizes efficiency by minimizing the number of Fock builds per unit of simulated time. Considering that the Fock matrix depends on the exchange-correlation functional and basis set, the upper bound in Eq. (24) and also the rate of convergence are affected by both choices. Unfortunately, the denominator in Eq. (24) is difficult to compute analytically, but the optimal choice of Δt can be estimated by running a few short test calculations with different values of Δt.

G. Convergence of approximations for the time-ordered integral

The time-ordered integration in Eq. (13) needs to be approximated at each time step. In LFLP-PC, a midpoint rule is used [Eq. (15)] and in EP-PC a trapezoidal rule is used [Eq. (23)], but in either case, the convergence of these approximations to the true time-ordered integration is unclear. Note that in Sec. II F, we used the word "consistency" to refer to self-consistency of the PC algorithms, and it is important to distinguish that in the present section we use the term "convergence" in a different context, specifically, to refer to consistency between the approximate integration scheme that is used in practice and the true time-ordered integration. The latter can be expressed in terms of the Magnus expansion that is introduced in Sec. II G 1, and in this context "convergence" means to estimate the truncation error that we introduce in this expansion. In Sec. II G 2, we use a transformation to the interaction representation to introduce an on-the-fly check, based on the norm of $\delta\mathbf{F}$ at each time step, as a way to monitor convergence of the integral approximations.

1. *Non-linear Magnus expansion*

The Magnus expansion[44,47] replaces the complicated time-ordered integration that defines the propagator by means of an infinite series of nested commutators involving the integrand (Fock matrix) evaluated at different times. For $t \in (t_N, t_{N+1})$, one solves a discretized version of either the TDKS equation [Eq. (5)] or the LvN equation [Eq. (10)] by propagating the MO coefficients ($\mathbf{C}_{N+1} = \mathbf{U}_N\mathbf{C}_N$) or the density matrix ($\mathbf{P}_{N+1} = \mathbf{U}_N\mathbf{P}_N\mathbf{U}^\dagger_N$), respectively, using

$$\mathbf{U}_N = \hat{\mathcal{T}}\exp\left(-i\int_{t_N}^{t_{N+1}} \mathbf{F}(\tau,\mathbf{P})\, d\tau\right). \tag{25}$$

[Note that this is the same as Eq. (13), but with the dependence of $\mathbf{F}$ on $\mathbf{P}$ made explicit.]

The Magnus expansion has been discussed previously in the context of the TDKS equation but only in its "linear" form,[35,46,48] which is appropriate when the operator appearing in the integrand is independent of the quantity that is being propagated in time, but not in the context of Eq. (25). The dependence of $\mathbf{F}$ on $\mathbf{P}$ makes the Magnus expansion a non-linear problem and thus we consider a generalized formalism that is appropriate in such cases.[44,47] This non-linear Magnus expansion replaces the time-ordered integration with an infinite series

$$\mathbf{U}_N = \exp(\boldsymbol{\Omega}) = \lim_{n\to\infty} \exp(\boldsymbol{\Omega}^{[n]}), \tag{26}$$

with

$$\boldsymbol{\Omega}^{[n]} = \sum_{k=1}^{n} \boldsymbol{\Omega}_k. \tag{27}$$

The terms $\boldsymbol{\Omega}_k$ in this expansion involve increasingly complicated integrals over nested commutators, but time ordering is not required. If truncated at $\boldsymbol{\Omega}^{[m]}$, the Magnus expansion is correct to $\mathcal{O}[(\Delta t)^m]$,[47] meaning that the difference $\boldsymbol{\Omega} - \boldsymbol{\Omega}^{[m]}$ introduces error of $\mathcal{O}[(\Delta t)^{m+1}]$. This implies that in evaluating each term $\boldsymbol{\Omega}_k$, one need only retain terms up to $\mathcal{O}[(\Delta t)^m]$ for mth-order truncation.

For mth-order truncation, one finds[47] that $\boldsymbol{\Omega}^{[0]}(t) \equiv \mathbf{0}$,

$$\boldsymbol{\Omega}^{[1]}(t) = -i \int_{t_N}^{t} \mathbf{F}(\tau, \mathbf{P}_N)\, d\tau, \tag{28}$$

and for $m \geq 2$,

$$\boldsymbol{\Omega}^{[m]}(t) = \sum_{k=0}^{m-2} \frac{B_k}{k!} \int_{t_N}^{t} d\tau \ \mathrm{ad}^k_{\boldsymbol{\Omega}^{[m-1]}(\tau)} \times \left(-i\mathbf{F}\left(\tau, e^{\boldsymbol{\Omega}^{[m-1]}(\tau)}\mathbf{P}_N \left[e^{\boldsymbol{\Omega}^{[m-1]}(\tau)}\right]^{\dagger}\right)\right). \tag{29}$$

Here, B_k are the Bernoulli numbers and ad^k is the nested commutator with the trivial definition

$$\mathrm{ad}^0_{\boldsymbol{\Omega}}\,(-i\mathbf{F}) = -i\mathbf{F}, \tag{30}$$

for $k = 0$ and

$$\mathrm{ad}^k_{\boldsymbol{\Omega}}\,(-i\mathbf{F}) = \left[\boldsymbol{\Omega},\ \mathrm{ad}^{k-1}_{\boldsymbol{\Omega}}\,(-i\mathbf{F})\right], \tag{31}$$

for $k \geq 1$.[47] For a second-order approximation, $\boldsymbol{\Omega}^{[1]}$ [Eq. (28)] needs to be approximated up to $\mathcal{O}(\Delta t)$, meaning that the errors enter at $\mathcal{O}[(\Delta t)^2]$, and

$$\boldsymbol{\Omega}^{[2]} = -i \int_{t_N}^{t_{N+1}} d\tau\ \mathbf{F}\left(\tau, e^{\boldsymbol{\Omega}^{[1]}(\tau)}\mathbf{P}_N \left[e^{\boldsymbol{\Omega}^{[1]}(\tau)}\right]^{\dagger}\right) \tag{32}$$

should be approximated up to $\mathcal{O}[(\Delta t)^2]$.

The simple Euler method amounts to the approximation $\boldsymbol{\Omega}^{[1]} \approx -i\mathbf{F}(t_N, P_N)\Delta t$, without taking account of $\boldsymbol{\Omega}^{[2]}$, and is therefore a first-order method. The MMUT, LFLP-PC, and EP-PC algorithms are second-order approximations that involve evaluation of both $\boldsymbol{\Omega}^{[1]}$ and $\boldsymbol{\Omega}^{[2]}$. In the case of MMUT, for $t \in [t_{N-1/2}, t_{N+1/2}]$, $\boldsymbol{\Omega}^{[1]}$ at the midpoint t_N is approximated by

$$\boldsymbol{\Omega}^{[1]}(t_N) = -i \int_{t_{N-1/2}}^{t_N} \mathbf{F}\, d\tau = -i\mathbf{F}_{N-1}\Delta t/2 + \mathcal{O}[(\Delta t)^2]\ . \tag{33}$$

Obviously the Fock matrix $\mathbf{F}_{N-1}$ used in this expression lags $\mathbf{F}_N$ at the end of the propagation interval, but we may estimate the error as

$$(\mathbf{F}_{N-1} - \mathbf{F}_N)\Delta t \sim \mathcal{O}[(\Delta t)^2]\ . \tag{34}$$

The midpoint $\mathbf{P}_N$ is then updated using $\boldsymbol{\Omega}^{[1]}$ and midpoint integration is applied to the integral appearing in $\boldsymbol{\Omega}^{[2]}$, with total error of $\mathcal{O}[(\Delta t)^3]$. Similarly, for LFLP-PC, $\boldsymbol{\Omega}^{[1]}$ is approximated by linear extrapolation and $\boldsymbol{\Omega}^{[2]}$ is calculated using the midpoint rule. For EP-PC, $\boldsymbol{\Omega}^{[1]}$ is approximated using the Euler method and $\boldsymbol{\Omega}^{[2]}$ is calculated using the trapezoidal rule.

Magnus expansions beyond second order are not generally employed in TDKS calculations because they dramatically increase the required number of Fock builds for each time step, and construction of the Fock matrix is the overwhelming bottleneck for TDKS calculations in Gaussian basis sets. To ensure that a second-order truncation of Eq. (26) is converged, the time interval must be much smaller than the convergence radius, which implies that[44]

$$\int_{t_N}^{t_{N+1}} \|\mathbf{F}(\tau)\|_2\, d\tau \ll \pi, \tag{35}$$

where $\|\cdots\|_2$ denotes the matrix 2-norm.[49]

2. Interaction representation

In this section, we transform the LvN equation into the interaction representation, in an effort to reduce the magnitude of $\|\mathbf{F}(\tau, \mathbf{P})\|_2$ and thus enlarge the radius of convergence in Δt, according to Eq. (35). One way to implement the interaction representation is to consider $\mathbf{F}_N$ as a reference Fock matrix in the Nth propagation step; the value $\mathbf{F}_N$ thus becomes an initial condition for time propagation over the interval $t \in [t_N,\ t_{N+1}]$. For simplicity, we therefore denote $\mathbf{F}_N$ as $\mathbf{F}_0$ in what follows and note that this quantity is independent of time for propagation over the aforementioned interval. We define $\delta\mathbf{F}_t = \mathbf{F}(t) - \mathbf{F}_N$ and partition the total Fock matrix at time t according to

$$\mathbf{F}_t = \mathbf{F}_0 + \delta\mathbf{F}_t. \tag{36}$$

Transform $\delta\mathbf{F}_t$ and $\mathbf{P}$ into the interaction representation according to

$$\delta\mathbf{F}^{\mathcal{I}}_t = e^{i\mathbf{F}_0 t}\ \delta\mathbf{F}_t\ e^{-i\mathbf{F}_0 t} \tag{37}$$

and

$$\mathbf{P}^{\mathcal{I}}_t = e^{i\mathbf{F}_0 t}\ \mathbf{P}_t\ e^{-i\mathbf{F}_0 t}. \tag{38}$$

In this representation, the LvN equation becomes

$$i\frac{\partial}{\partial t}\mathbf{P}^{\mathcal{I}}_t = [\delta\mathbf{F}^{\mathcal{I}}_t,\ \mathbf{P}^{\mathcal{I}}_t], \tag{39}$$

therefore the propagator in the interaction representation is

$$\mathbf{U}^{\mathcal{I}}(t_{N+1}, t_N) = \hat{\mathcal{T}}\exp\left(-i\int_{N\Delta t}^{(N+1)\Delta t} \delta\mathbf{F}^{\mathcal{I}}_t\, dt\right). \tag{40}$$

The time integration proceeds according to

$$\mathbf{P}^{\mathcal{I}}_{N+1} = \mathbf{U}^{\mathcal{I}}(t_{N+1}, t_N)\,\mathbf{P}^{\mathcal{I}}_N\ [\mathbf{U}^{\mathcal{I}}(t_{N+1}, t_N)]^{\dagger}\ . \tag{41}$$

Since $\delta\mathbf{F}^{\mathcal{I}}_t$ is the time-varying part of the Fock matrix with respect to $\mathbf{F}_N$, it should be relatively small (especially when the external field is weak), and accordingly $\|\delta\mathbf{F}^{\mathcal{I}}_t\|_2$ should be much smaller than $\|\mathbf{F}\|_2$. This should enlarge the convergence radius in Δt.

Next, we demonstrate that the algorithm for each time step in the interaction picture yields the same final formalism as in the Heisenberg picture, within the same order of error in Δt, when the Magnus expansion is truncated at the same order in both representations. We explicitly consider the LFLP-PC case (i.e., midpoint approximation for the integrand), but the EP-PC case (trapezoidal approximation) can be proved in a similar way. Using midpoint approximation in the interaction representation, the propagator is

$$\mathbf{U}^{\mathcal{I}}(t_{N+1},t_N) = \exp(-i\Delta t\,\delta\mathbf{F}^{\mathcal{I}}_{N+1/2})\ , \tag{42}$$

which in light of Eq. (37) can be recasted as

$$\begin{aligned}\mathbf{U}^{\mathcal{I}}(t_{N+1},t_N) &= \exp\{-i\Delta t \exp\left[i(N+1/2)\Delta t\,\mathbf{F}_0\right] \\ &\qquad \times \delta\mathbf{F}_{N+1/2}\exp\left[-i(N+1/2)\Delta t\,\mathbf{F}_0\right]\} \\ &= \sum_{n=0}^{\infty}\frac{1}{n!}\{-i\Delta t \exp\left[i(N+1/2)\Delta t\,\mathbf{F}_0\right] \\ &\qquad \times \delta\mathbf{F}_{N+1/2}\exp\left[-i(N+1/2)\Delta t\,\mathbf{F}_0\right]\}^n\ .\end{aligned} \tag{43}$$

Multiplying out the power series and combining terms, we obtain

$$\begin{aligned}\mathbf{U}^{\mathcal{I}}(t_{N+1},t_N) &= e^{i(N+1/2)\Delta t\,\mathbf{F}_0} \\ &\qquad \times\left[\sum_{n=0}^{\infty}\frac{(-i\Delta t\,\delta\mathbf{F}_{N+1/2})^n}{n!}\right]e^{-i(N+1/2)\Delta t\,\mathbf{F}_0} \\ &= e^{i(N+1/2)\Delta t\,\mathbf{F}_0}e^{-i\Delta t\,\delta\mathbf{F}_{N+1/2}}e^{-i(N+1/2)\Delta t\,\mathbf{F}_0}\ .\end{aligned} \tag{44}$$

Substituting this expression for $\mathbf{U}^{\mathcal{I}}$ into Eq. (41) and then inverting Eq. (38) to solve for $\mathbf{P}_{N+1}$, one obtains

$$\begin{aligned}\mathbf{P}_{N+1} &= e^{-i(\Delta t/2)\mathbf{F}_0}\ e^{-i\Delta t\,\delta\mathbf{F}_{N+1/2}}\ e^{-i(\Delta t/2)\mathbf{F}_0} \\ &\qquad \times\mathbf{P}_N\ e^{i(\Delta t/2)\mathbf{F}_0}\ e^{i\Delta t\,\delta\mathbf{F}_{N+1/2}}\ e^{i(\Delta t/2)\mathbf{F}_0}\ .\end{aligned} \tag{45}$$

According to the second-order Trotter-Suzuki factorization,[50]

$$e^{(\hat{A}+\hat{B})\Delta t} = e^{\hat{A}\Delta t/2}\ e^{\hat{B}\Delta t}\ e^{\hat{A}\Delta t/2} + \mathcal{O}[(\Delta t)^3]\ , \tag{46}$$

therefore

$$\begin{aligned}&e^{-i(\Delta t/2)\mathbf{F}_0}\ e^{-i\Delta t\,\delta\mathbf{F}_{N+1/2}}\ e^{-i(\Delta t/2)\mathbf{F}_0} \\ &\qquad = e^{-i\Delta t(\mathbf{F}_0+\delta\mathbf{F}_{N+1/2})} + \mathcal{O}[(\Delta t)^3],\end{aligned} \tag{47}$$

with a local truncation error of $\mathcal{O}[(\Delta t)^3]$. Equation (45) can be further simplified by combining exponentials to obtain

$$\mathbf{P}_{N+1} = e^{-i\Delta t(\mathbf{F}_0+\delta\mathbf{F}_{N+1/2})}\ \mathbf{P}_N\ e^{i\Delta t(\mathbf{F}_0+\delta\mathbf{F}_{N+1/2})}. \tag{48}$$

This equation is identical with step 2 in the LFLP-PC algorithm within the Heisenberg picture,

$$\mathbf{P}_{N+1} = e^{-i\Delta t\,\mathbf{F}_{N+1/2}}\ \mathbf{P}_N\ e^{i\Delta t\,\mathbf{F}_{N+1/2}}. \tag{49}$$

This result, and the analogous one for the LFLP-PC scheme, demonstrates that PC propagation of the density matrix is identical within the Heisenberg and interaction representations, within a local truncation error of $\mathcal{O}[(\Delta t)^3]$. As such, the convergence criterion in Eq. (35) is actually determined by $\|\delta\mathbf{F}^{\mathcal{I}}_t\|_2$. The latter is bounded from above by its Frobenius norm[49] and can be simplified in the Heisenberg picture using trace invariance under cyclic permutations, i.e.,

$$\|\delta\mathbf{F}^{\mathcal{I}}\|_2 \le \mathrm{tr}[\delta\mathbf{F}^{\mathcal{I}}(\delta\mathbf{F}^{\mathcal{I}})^{\dagger}] = \mathrm{tr}(\delta\mathbf{F}\delta\mathbf{F}^{\dagger}) = \|\delta\mathbf{F}\|_{\mathrm{F}}. \tag{50}$$

Thus the quality of the truncated Magnus expansion can be monitored by checking $\|\delta\mathbf{F}\|_{\mathrm{F}}$ on-the-fly during the calculation. In principle, this could be used to create an adaptive step-size algorithm that reduces Δt when necessary, although we have not yet implemented such a procedure.

III. NUMERICAL RESULTS

The proposed algorithms were implemented in the Q-Chem program,[51] building upon the initial TDKS implementation in Q-Chem that was reported by Nguyen and Parkhill.[36] These algorithms are available starting from v. 5.0 of the code. Most calculations reported here use either the PBE[52] or the LRC-ωPBEh[53] density functional, and the 6-31G(d) basis set, although other functionals and basis sets are briefly considered to demonstrate that these results are indeed representative. The choice of an appropriate functional and basis set for applications is beyond the scope of this paper, though we note that basis sets much larger than 6-31G(d) are required for modeling strong-field effects,[40,54,55] and functionals such as LRC-ωPBEh should be tuned.[40,56] Description of Rydberg states, which are important in high harmonic generation,[55,57,58] certainly requires diffuse basis functions at least. Finally, studies of strong-field ionization require in addition the use of absorbing boundary conditions,[37,54–56] which are not considered here.

TDKS calculations start from the electronic ground state, subject to a weak perturbation during the first time step, whose magnitude is 10^{-5} a.u. (5 mV/nm) in each of the x, y, and z directions,[59] as appropriate for obtaining an unpolarized absorption spectrum. For TDKS calculations using the PC algorithms, the threshold ξ [Eq. (21)] is set to 10^{-7} unless stated otherwise. TDKS absorption spectra are computed from the trace of the cross section tensor,[35]

$$S(\omega) = \tfrac{1}{3}\left[\sigma_{xx}(\omega)+\sigma_{yy}(\omega)+\sigma_{zz}(\omega)\right]\ , \tag{51}$$

where

$$\sigma_{ij}(\omega) = \left(\frac{4\pi\omega}{c}\right)\mathfrak{Im}\left[\alpha_{ij}(\omega)\right] \tag{52}$$

depends on the imaginary part of the frequency-dependent polarizability tensor, whose matrix elements

$$\alpha_{ij}(\omega) = \frac{\partial\mu_i}{\partial E_j(\omega)} \tag{53}$$

(where $i,j\in\{x,y,z\}$) describe how the induced dipole moment μ_i responds to an applied electric field with frequency ω propagating in direction j.

A. Numerical stability

We find that the LFLP-PC3 algorithm, in which the initial predictor matrix $\mathbf{F}^{\mathrm{p}}_{N+1/2}$ is always averaged with the current corrector matrix $\mathbf{F}^{\mathrm{c}}_{N+1/2}$ to estimate the half-step Fock matrix, is unstable even when Δt is small; see Fig. S1 in the supplementary material. This is a general problem for predictor/corrector methods applied to ordinary differential equations, where the use of the average of the initial predictor value with the current corrector value as a starting point for the next iteration can lead to instability.[60] (Curiously, however, we observe no such instability for EP-PC3, where the initial $\mathbf{P}^{\mathrm{p}}_{N+1}$ is always averaged with the current $\mathbf{P}^{\mathrm{c}}_{N+1}$.) For this reason, we do not consider the LFLP-PC3 algorithm in what follows, although we do consider EP-PC3.

To compare the stability of the MMUT and PC algorithms for large time steps, we simulated one water molecule for 90 fs following perturbation by an external electric field. For demonstrative and comparative purposes, these tests

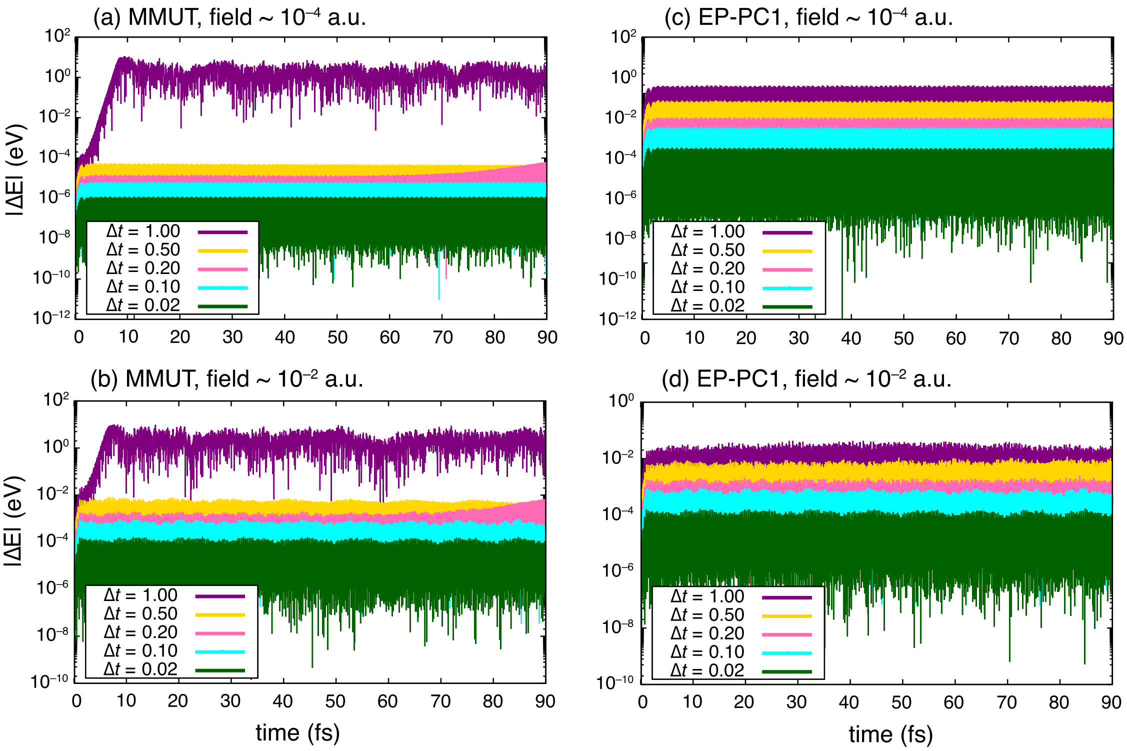

FIG. 4. Fluctuations in the total energy for TDKS simulations of H_2O at the PBE/6-31G(d) level using two different propagators (MMUT on the left and EP-PC1 on the right), for time steps ranging from 0.02 to 1.0 a.u. As the time step is increased, the MMUT algorithm becomes unstable, whereas energy fluctuations are stable using EP-PC1. Two different external field amplitudes are considered; note that the vertical scales are different in the various panels.

intentionally use time steps >0.2 a.u., significantly larger than the ~0.05 a.u. time steps that are typically used in MMUT simulations.[33,34] The impulsive, time-dependent external electric field is selected to have a frequency that is resonant with the bright state of H_2O (corresponding to 12.0 eV for PBE and 12.6 eV for LRC-ωPBEh), with a Gaussian pulse

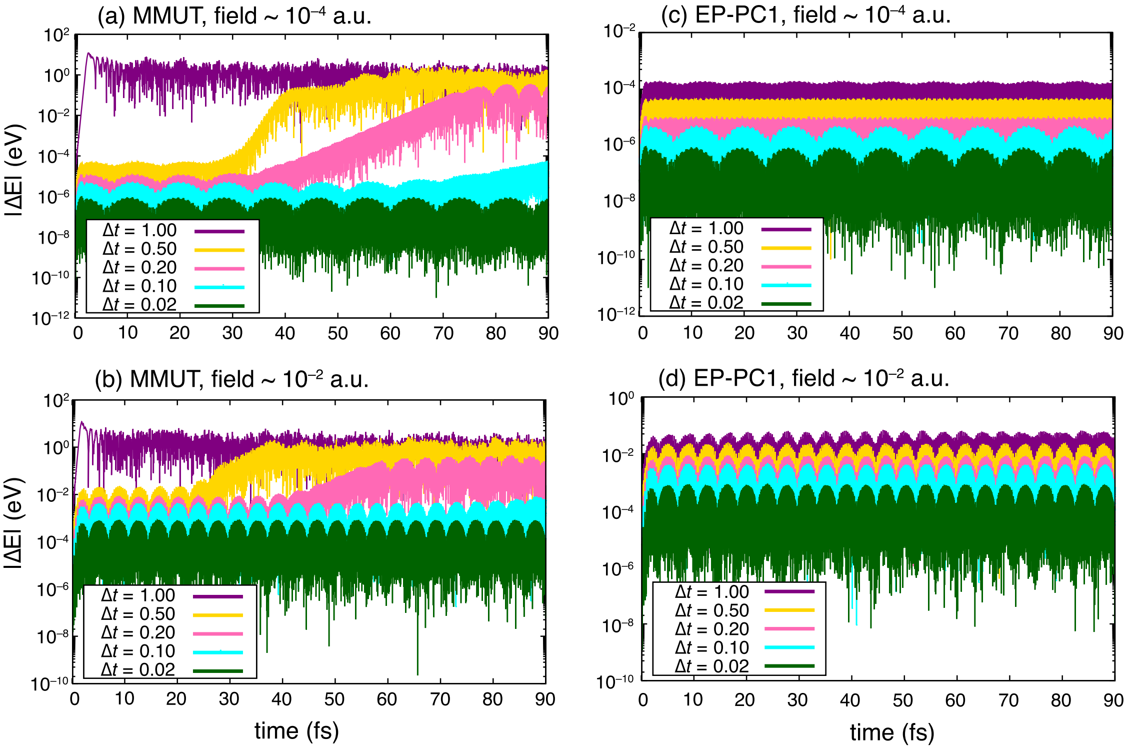

FIG. 5. Fluctuations in the total energy for TDKS simulations of H_2O at the LRC-ωPBEh/6-31G(d) level using two different propagators (MMUT on the left and EP-PC1 on the right), for time steps ranging from 0.02 to 1.0 a.u. As the time step is increased, the MMUT algorithm becomes unstable, whereas energy fluctuations are stable using EP-PC1. Two different external field amplitudes are considered; note that the vertical scales are different in the various panels.

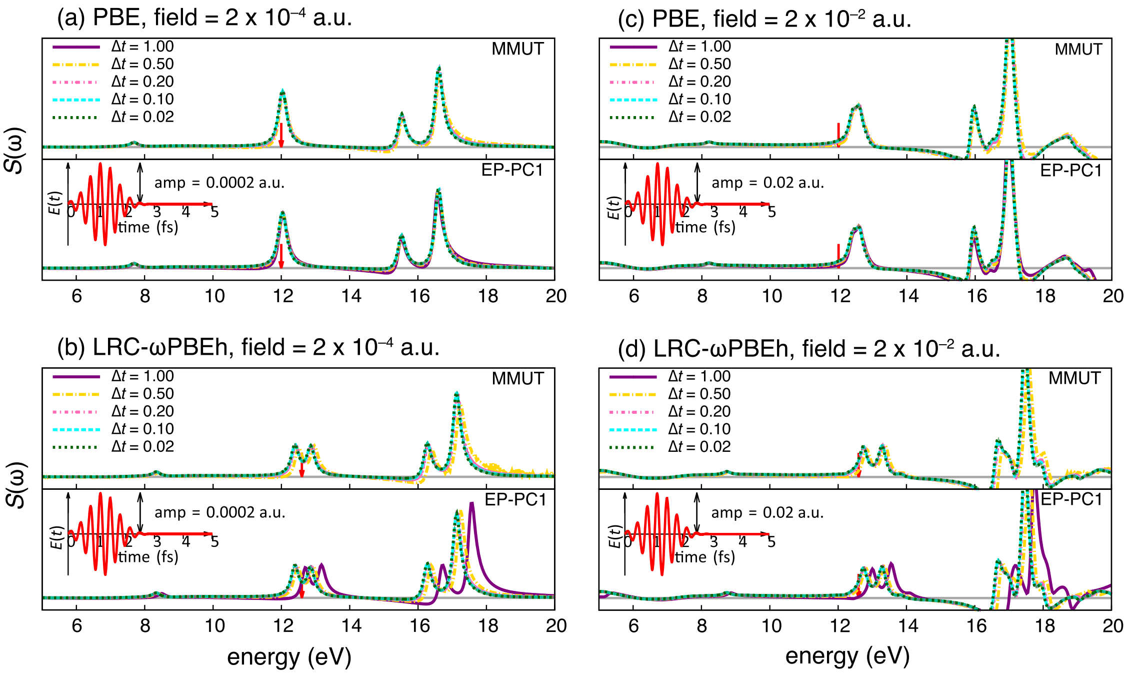

FIG. 6. Absorption cross sections $S(\omega)$, in arbitrary units, for H_2O computed using different propagators (MMUT and EP-PC1) and time steps (0.02–1.0 a.u.). The gray horizontal line in the background is a reference line indicating where $S(\omega) \equiv 0$. The impulsive, time-dependent external field has a Gaussian envelope (shown in red) and a resonant frequency whose value, in energy units, is indicated by the red arrow around 12 eV.

envelope whose maximum amplitude ranges from 10^{-4} -10^{-1} a.u. Following the pulse, the molecule evolves from the ground state into a superposition of states, and if the external field is weak, then this superposition will be dominated by the ground eigenstate, with an excited-state contribution that is dominated by states around 12 eV.

Figures 4 and 5, and Fig. S2 in the supplementary material, depict fluctuations in the total energy for TDKS simulations of H_2O at the PBE/ and LRC-ωPBEh/6-31G(d) levels. Both the MMUT and EP-PC1 propagator algorithms are examined, using time steps ranging up to $\Delta t = 1.0$ a.u. (=0.024 fs). Because of the resonant nature of the pulse, the molecule gains energy from the external field, causing an increase in the total energy immediately following the pulse (Fig. S5 of the supplementary material). Energy fluctuations are computed as the total energy change between successive steps, $|E_N - E_{N-1}|$. The maximum field amplitude is 2×10^{-4} a.u. for the weak field and 0.2 a.u. for the strong field.

In practical applications, the MMUT algorithm requires that much smaller time steps be used,[35–41] and for good reason: our results demonstrate that this algorithm is no longer stable for $\Delta t \geq 0.2$ a.u. at the PBE/6-31G(d) level (Fig. 4) or for $\Delta t \geq 0.1$ a.u. at the LRC-ωPBEh/6-31G(d) level (Fig. 5). The non-local exchange term evidently makes the system more sensitive to errors, likely because the Hartree-Fock exchange contribution to the Fock matrix oscillates more rapidly in time as compared to other components of the exchange-correlation functional.[61,62] Meanwhile, the induced dipole diverges to unrealistically large values as errors accumulate; see Fig. S3 in the supplementary material. The EP-PC1 propagator, however, is stable across the entire range of time steps in Figs. 4 and 5, up to 1.0 a.u., and the induced dipole moment exhibits stable oscillations with reasonable amplitude (Fig. S4 of the supplementary material). Finally, we note that fluctuations in tr(**P**) are $<10^{-7}$ in all simulations. Since matrix exponents are computed exactly in this work, via diagonalization, and our PC methods are designed such that averages are taken in the exponent, any deviations in tr(**P**) are due to round-off errors that are controllable by tightening numerical drop tolerances.

TABLE I. Maximum values of $\Delta t \|\delta \mathbf{F}\|_F$, rounded to the nearest power of ten, for TDKS simulations of H_2O, using the EP-PC1 propagator for various time steps and field amplitudes. The total simulation time is 90 fs in each case.

	Δt (a.u.)									
	PBE					LRC-ωPBEh				
Field strength (a.u.)	0.02	0.10	0.20	0.50	1.00	0.02	0.10	0.20	0.50	1.00
2×10^{-4}	10^{-7}	10^{-5}	10^{-5}	10^{-4}	10^{-3}	10^{-6}	10^{-5}	10^{-4}	10^{-4}	10^{-3}
2×10^{-3}	10^{-6}	10^{-4}	10^{-4}	10^{-3}	10^{-2}	10^{-5}	10^{-4}	10^{-3}	10^{-3}	10^{-2}
2×10^{-2}	10^{-5}	10^{-3}	10^{-3}	10^{-2}	10^{-1}	10^{-4}	10^{-3}	10^{-2}	10^{-2}	10^{-1}
2×10^{-1}	10^{-3}	10^{-1}	10^{-1}	10^{0}	10^{0}	10^{-3}	10^{-1}	10^{-1}	10^{0}	10^{0}

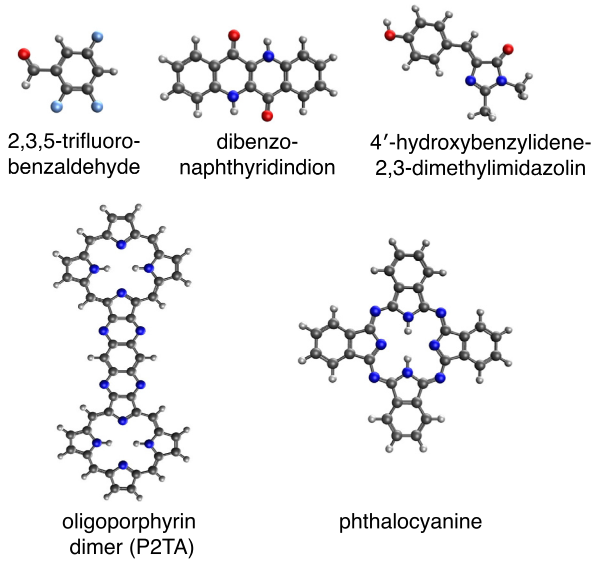
2,3,5-trifluoro-benzaldehyde
dibenzo-naphthyridindion
4′-hydroxybenzylidene-2,3-dimethylimidazolin
oligoporphyrin dimer (P2TA)
phthalocyanine

FIG. 7. Larger molecules used to check the convergence of the PC iterations.

Absorption cross sections $S(\omega)$ for this same set of simulations (Figs. 4 and 5, and Fig. S2 of the supplementary material) are plotted in Fig. 6 and Fig. S6. As the pulse amplitude increases, the spectra begin to deviate from the weak-field result. Using the MMUT algorithm, the spectrum can be resolved for $\Delta t \leq 0.5$ a.u., with the divergence that appears in the late-stage propagation (Figs. 4 and 5) manifesting as noise in the spectrum. Presumably, however, these spectra will eventually be unresolvable if the divergent dynamics is propagated further forward in time. The EP-PC algorithms, on the other hand, have no problem resolving the spectrum all the way up to $\Delta t = 1.0$ a.u., although the peaks start to shift as Δt increases. This shifting is discussed in Sec. III B.

To verify convergence of Magnus expansion, we monitored $\|\delta\mathbf{F}\|_{\mathrm{F}}$ on-the-fly during these simulations. Table I lists the maximum value of the product $\Delta t\|\delta\mathbf{F}\|_{\mathrm{F}}$ for each simulation, a quantity that should be bounded for a convergent simulation, according to Eq. (35). For weak fields, the values in Table I are quite small in comparison to the region of convergence indicated by Eq. (35), even for large values of Δt, which demonstrates that first-order truncation of the Magnus expansion is not a critical limitation for simulation of weak-field electron dynamics. For stronger fields, however, the maximum value of $\Delta t\|\delta\mathbf{F}\|_{\mathrm{F}}$ becomes comparable to the radius of convergence, and as such $\|\delta\mathbf{F}\|_{\mathrm{F}}$ is a much more important quantity that should be monitored in simulations of strong-field dynamics.

To check the convergence of the iterative step in the PC methods (step 5 in Figs. 1 and 2), we ran tests on a set of

TABLE II. Average number of Fock build per time step for different molecules, calculated at the PBE/6-31G(d) level with a threshold $\xi = 10^{-7}$. The notation "n.c." indicates that the simulation did not converge, either because the predicted and corrected matrices diverged from one another or else because more than 200 Fock builds were required in a single time step.

	Number of Fock builds					Number of Fock builds				
	LFLP-		EP-			LFLP-		EP-		
Δt (a.u.)	PC1	PC2	PC1	PC2	PC3	PC1	PC2	PC1	PC2	PC3
	Water					Benzene				
0.5	2.94	2.94	2.06	2.13	2.13	2.13	2.13	2.00	2.00	2.00
1.0	4.06	4.19	3.19	3.13	3.06	2.75	2.81	2.06	2.13	2.13
1.5	9.19	9.25	5.06	5.25	5.63	3.50	3.56	2.75	2.69	2.69
2.0	66.06	n.c.	n.c.	n.c.	n.c.	4.81	4.88	3.69	3.56	3.69
	Adenine					2,3,5-trifluorobenzaldehyde				
0.5	2.13	2.13	2.00	2.00	2.00	2.13	2.13	2.00	2.00	2.00
1.0	2.81	2.81	2.00	2.00	2.00	3.38	3.38	2.13	2.06	2.06
1.5	4.00	3.88	2.81	2.56	2.56	11.44	11.13	3.50	2.69	3.13
2.0	14.69	13.56	7.00	6.81	6.69	n.c.	n.c.	n.c.	n.c.	n.c.
	Dibenzonaphthyridindione					4′-hydroxybenzylidene-2,3-dimethylimidazoline				
0.5	2.13	2.13	2.00	2.00	2.00	2.13	2.13	2.00	2.00	2.00
1.0	2.69	2.63	2.00	2.00	2.00	2.63	2.56	2.00	2.00	2.00
1.5	4.25	3.25	2.25	2.25	2.25	4.75	3.88	2.56	2.38	2.38
2.0	n.c.	n.c.	n.c.	n.c.	n.c.	n.c.	n.c.	n.c.	n.c.	n.c.
	Oligoporphyrin dimer					Phthalocyanine				
0.5	2.13	2.13	2.00	2.00	2.00	2.13	2.13	2.00	2.00	2.00
1.0	2.38	2.31	2.00	2.00	2.00	2.38	2.38	2.00	2.00	2.00
1.5	3.00	2.75	2.00	2.00	2.00	2.81	2.63	2.00	2.00	2.00
2.0	6.56	3.44	2.13	2.19	2.19	6.06	3.75	2.63	2.25	2.25

molecules ranging in size from H_2O up to an 84-atom oligoporphyrin dimer. The molecules in the test set are water, benzene, adenine, and five larger molecules depicted in Fig. 7, which were taken from Ref. 63. In Table II (for PBE) and Table III (for LRC-ωPBEh), we list the average number of Fock builds required per time step, for each molecule in the test set, using different propagator algorithms and values of Δt in the range 0.5–2.0 a.u. (For Δt = 2.0 a.u., convergence usually requires more than 10 Fock builds per time step, and for larger time steps, most simulations fail to converge within 200 iterations.) Note that we use a threshold $\xi = 10^{-7}$ for the PBE functional but $\xi = 10^{-8}$ for LRC-ωPBEh, in an effort to obtain spectra of similar quality with both functionals. This difference is discussed in Sec. III B.

The convergence properties with respect to Δt are similar for the LFLP-PC and EP-PC methods, and stable convergence is obtained for each molecule and algorithm out to at least Δt = 1.5 a.u. The two LFLP-PC variants perform similarly, and likewise the three EP-PC variants afford similar performance to one another. Because the convergence properties depend upon the choice of functional and basis set, we performed additional tests on the adenine molecule using the LFLP-PC1 and EP-PC1 algorithms in conjunction with the 6-31G, 6-31G(d), and 6-31+G basis sets and 10 different exchange-correlation functionals, including generalized gradient approximations (GGAs), global hybrid functionals, and long-range corrected (i.e., range-separated hybrid) functionals. The performance of each method is characterized in terms of the average number of Fock builds per time step, for Δt in the range 0.1–2.0 a.u., with results shown in Table IV. The optimal choice of Δt is not necessarily the choice that minimizes the number of Fock builds per time step but rather that which minimizes the number of Fock builds *per unit of simulated time*. Although this optimal choice depends somewhat on the choice of functional and basis set, in each case it lies in the range Δt = 1.0–1.5 a.u. This conclusion, reached for a single molecule using a collection of functionals and basis sets, is consistent with the conclusion drawn from Tables II and III based on a selection of molecules using a single functional and basis set.

Whereas the MMUT algorithm requires exactly one Fock build per time step, the PC methods require a minimum of two and are thus more efficient only insofar as a time step twice as large can be used. A typical MMUT time step is $\Delta t \lesssim 0.1$ a.u., and the data in Table IV show that the PC algorithms with Δt = 0.2 a.u. require just slightly more than two Fock builds per step, meaning that the efficiency of the new algorithms is comparable or superior to that of MMUT in most cases. (Even when the efficiency is about the same, the PC methods facilitate on-the-fly checks of the stability of the time propagation that

TABLE III. Average number of Fock build per time step for different molecules, calculated at the LRC-ωPBEh/6-31G(d) level with a threshold $\xi = 10^{-8}$. The notation "n.c." indicates that the simulation did not converge, either because the predicted and corrected matrices diverged from one another or else because more than 200 Fock builds were required in a single time step.

	Number of Fock builds					Number of Fock builds				
	LFLP-		EP-			LFLP-		EP-		
Δt (a.u.)	PC1	PC2	PC1	PC2	PC3	PC1	PC2	PC1	PC2	PC3
	Water					Benzene				
0.5	2.81	2.94	3.63	3.63	3.56	2.38	2.38	2.94	2.94	2.94
1.0	4.50	4.50	6.19	6.13	5.56	3.25	3.19	4.44	4.44	4.44
1.5	7.50	8.00	10.75	10.69	10.56	4.50	4.56	6.56	6.56	6.63
2.0	11.13	12.75	20.50	20.50	19.63	6.56	6.69	9.56	9.56	10.00
	Adenine					2,3,5-trifluorobenzaldehyde				
0.5	2.38	2.38	3.06	3.06	3.06	2.44	2.63	3.38	3.38	3.31
1.0	3.88	3.94	5.38	5.38	5.44	5.00	4.81	6.81	6.75	6.63
1.5	6.75	6.81	10.13	10.13	10.38	11.56	11.31	13.88	13.69	14.06
2.0	14.44	14.50	21.50	21.38	21.75	17.81	14.56	n.c.	n.c.	n.c.
	Dibenzonaphthyridindione					4′-hydroxybenzylidene- 2,3-dimethylimidazoline				
0.5	2.31	2.31	3.13	3.13	3.13	2.31	2.31	3.13	3.13	3.19
1.0	3.81	3.81	5.75	5.75	5.56	4.13	4.13	6.06	6.06	6.06
1.5	7.69	7.94	12.00	12.00	12.25	8.94	9.19	13.50	13.50	13.88
2.0	20.75	20.50	n.c.	n.c.	n.c.	28.31	28.50	n.c.	n.c.	n.c.
	Oligoporphyrin dimer					Phthalocyanine				
0.5	2.38	2.38	3.15	3.15	3.25	2.44	2.50	3.13	3.13	3.13
1.0	3.64	3.58	5.25	5.33	5.29	3.38	3.38	5.06	5.06	5.19
1.5	6.60	6.29	8.75	8.75	8.50	5.88	5.75	9.08	9.08	9.25
2.0	10.50	11.50	12.50	14.67	12.00	13.50	13.13	19.60	19.80	21.00

TABLE IV. Average number of Fock builds per time step for TDKS simulations of adenine, computed using the LFLP-PC1 or EP-PC1 propagator algorithm in conjunction with various exchange-correlation functionals, basis sets, and time steps. The notation "n.c." indicates that the simulation did not converge, either because the predicted and corrected matrices diverged from one another or else because more than 200 Fock builds were required in a single time step. In each case, the optimal choice of Δt, which minimizes the number of Fock builds per unit of simulated time, is highlighted in boldface type.

	6-31G		6-31G(d)		6-31+G		6-31G		6-31G(d)		6-31+G	
Δt (a.u.)	LFLP	EP	LFLP	EP	LFLP	EP	LFLP	EP	LFLP	EP	LFLP	EP
	PBE						LRC-ωPBEh					
0.1	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00
0.2	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.13	2.13	2.00	2.13	2.13
0.5	2.25	2.94	2.13	2.00	2.25	2.94	2.31	3.06	2.13	2.00	2.31	3.06
1.0	**3.38**	**4.63**	**2.13**	**2.06**	**3.56**	**4.75**	**3.75**	**5.31**	2.13	2.25	**4.06**	**5.38**
1.5	6.56	9.88	4.00	3.88	6.19	9.50	7.31	23.06	**3.00**	**3.25**	7.00	10.44
2.0	28.81	n.c.	14.69	13.56	22.44	n.c.	15.50	n.c.	5.83	6.64	13.75	21.75
	PBE0						M06-L					
0.1	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00
0.2	2.13	2.00	2.11	2.00	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00
0.5	2.25	2.94	2.13	2.00	2.25	2.94	2.25	2.94	2.13	2.00	2.25	2.94
1.0	**3.56**	**4.94**	2.13	2.13	**3.75**	**5.19**	**3.63**	**4.69**	2.13	2.13	**3.69**	**4.81**
1.5	7.00	10.38	**2.44**	**3.19**	6.94	10.00	6.50	9.25	**3.00**	**3.13**	6.06	9.13
2.0	20.75	30.25	7.25	7.31	16.06	26.25	18.56	28.13	4.91	6.87	15.63	25.13
	B3LYP						M06					
0.1	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00
0.2	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00
0.5	2.25	2.94	2.13	2.00	2.25	2.94	2.25	2.94	2.13	2.00	2.25	2.94
1.0	**3.56**	**4.81**	**2.13**	**3.19**	**3.75**	**5.13**	**3.63**	**5.00**	**2.13**	**2.13**	**3.69**	**5.19**
1.5	7.00	10.44	2.38	8.56	6.81	10.31	6.94	10.44	2.50	3.38	6.88	10.44
2.0	23.81	n.c.	8.25	n.c.	18.94	24.33	22.88	n.c.	5.82	8.50	18.94	25.33
	CAM-B3LYP						M06-2X					
0.1	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00
0.2	2.13	2.06	2.13	2.00	2.13	2.06	2.13	2.13	2.13	2.00	2.13	2.13
0.5	2.25	3.06	2.13	2.00	2.25	3.06	2.25	3.06	2.13	2.00	2.25	3.06
1.0	**3.69**	**5.25**	2.13	**2.19**	**4.06**	**5.31**	**3.63**	**5.31**	2.13	2.19	**4.00**	**5.38**
1.5	7.19	10.50	**2.94**	3.31	6.94	10.88	7.25	10.25	**2.94**	**3.31**	7.00	10.44
2.0	18.19	27.50	5.63	6.88	14.69	24.50	16.19	25.13	5.00	7.09	13.50	22.44
	LRC-ωPBE						M11-L					
0.1	2.13	2.00	2.13	2.00	2.13	2.00	2.13	2.00	2.09	2.00	2.13	2.00
0.2	2.13	2.13	2.13	2.00	2.13	2.13	2.13	2.13	2.13	2.00	2.13	2.13
0.5	2.31	3.06	2.13	2.00	2.31	3.06	2.31	3.06	2.13	2.00	2.31	3.06
1.0	**4.00**	**5.44**	**2.13**	**2.25**	**4.06**	**5.44**	**3.81**	**5.56**	**2.13**	2.44	**4.13**	**5.63**
1.5	7.81	11.63	2.88	3.56	7.81	11.56	6.94	10.63	3.31	**3.30**	7.25	10.56
2.0	25.56	n.c.	8.13	8.72	21.19	n.c.	14.19	20.69	5.94	5.36	12.69	19.50

are not possible with MMUT.) For a more direct comparison to the efficiency of the MMUT algorithm, we took the PBE0 data from Table IV and converted them into units of "equivalent MMUT time steps," defined as the (average) time step that can be taken for the cost of a single Fock build, or in other words the time step that would be needed in the MMUT algorithm in order to match the efficiency of the PC algorithm. Numerically, the equivalent MMUT time steps are simply the reciprocals of the average Fock builds listed in Table IV, and the former values are given in Table V. For $\Delta t < 0.2$ a.u., this equivalent MMUT time step is no larger than a typical, stable MMUT time step as dictated by energy conservation (~0.1 a.u.), which is another way of stating that the PC and MMUT algorithms are approximately equally efficient for time steps this small. For actual time steps in the range 0.5–1.5 a.u., however, the efficiency of the PC algorithms is in some cases comparable to a MMUT algorithm with a time step of 0.5–0.6 a.u. (assuming such a simulation were even stable using MMUT), meaning that the PC algorithm is 2–3 times more efficient.

Referring to Eq. (24), the convergence rate can be adjusted using Δt. As such, a feasible way to find an appropriate value of Δt is to run tests using different values in the beginning, prior to committing to large-scale simulations, in order to estimate the range of convergence with respect to the time step. This allows the user to estimate the most efficient choice of Δt *a priori* since this choice is obviously a compromise between

TABLE V. Equivalent MMUT time steps for TDKS simulations of adenine using the PBE0 functional and the LFLP-PC1 and EP-PC1 algorithms. The equivalent MMUT Δt is defined as the average time step that can be taken for the cost of a single Fock build, i.e., at the cost of the MMUT algorithm. Optimal (largest) values are highlighted in boldface type. Numerically, these values are the reciprocals of the PBE0 data in Table IV.

	Equivalent MMUT Δt (a.u.)					
Actual Δt	6-31G		6-31G(d)		6-31+G	
(a.u.)	LFLP	EP	LFLP	EP	LFLP	EP
0.1	0.05	0.05	0.05	0.05	0.05	0.05
0.2	0.09	0.10	0.09	0.10	0.09	0.10
0.5	0.22	0.17	0.24	0.25	0.22	0.17
1.0	**0.28**	**0.20**	0.47	0.47	**0.27**	**0.19**
1.5	0.21	0.14	**0.62**	**0.47**	0.22	0.15
2.0	0.10	0.07	0.28	0.27	0.12	0.08

reducing the number of steps at the expense of steps that require more than one Fock build. For non-self-consistent approaches such as the MMUT propagator, the validity of a simulation cannot be guaranteed or known even after the fact, except in the trivial case of a divergent simulation that fails to conserve energy and thus indicates a much too aggressive choice for Δt. For stable simulations, the convergence of computed properties (e.g., spectra) should in principle be checked by running several full-length simulations using different values of Δt. A significant advantage of PC methods is that convergence of the Magnus expansion can be monitored and Δt adjusted on-the-fly to ensure convergence.

B. Convergence of calculated spectra

Finally, we examine how absorption spectra computed from TDKS simulations converge with respect to the threshold ξ and the time step Δt. Simulations in this section were propagated for a total simulation time $T = 5,000$ a.u. (≈121 fs). The resulting spectral line shapes are broadened by damping the induced dipole moment as $\boldsymbol{\mu}(t)\ e^{-t/\tau}$ with $\tau = 200$ a.u. Absolute values of the spectra are taken to avoid spurious negative parts due to numerical artifacts introduced by finite time-sampling.[43]

We first examine the dependence of the computed spectra on the threshold ξ. We calculate TDKS spectra of single H_2O molecule using $\Delta t = 0.5$ a.u. with ξ ranging from 10^{-5} to 10^{-10}. Figure 8 shows TDKS spectra computed using the EP-PC3 algorithm in comparison to stick spectra from a LR-TDDFT calculation. (TDKS results using other propagators are similar and can be found in Figs. S9–S12 of the supplementary material.) As ξ is decreased, the TDKS spectra converge to the LR result. For the loosest thresholds ($\xi = 10^{-5}$ for PBE and $\xi = 10^{-6}$ for LRC-ωPBEh), the TDKS spectra are noisy and

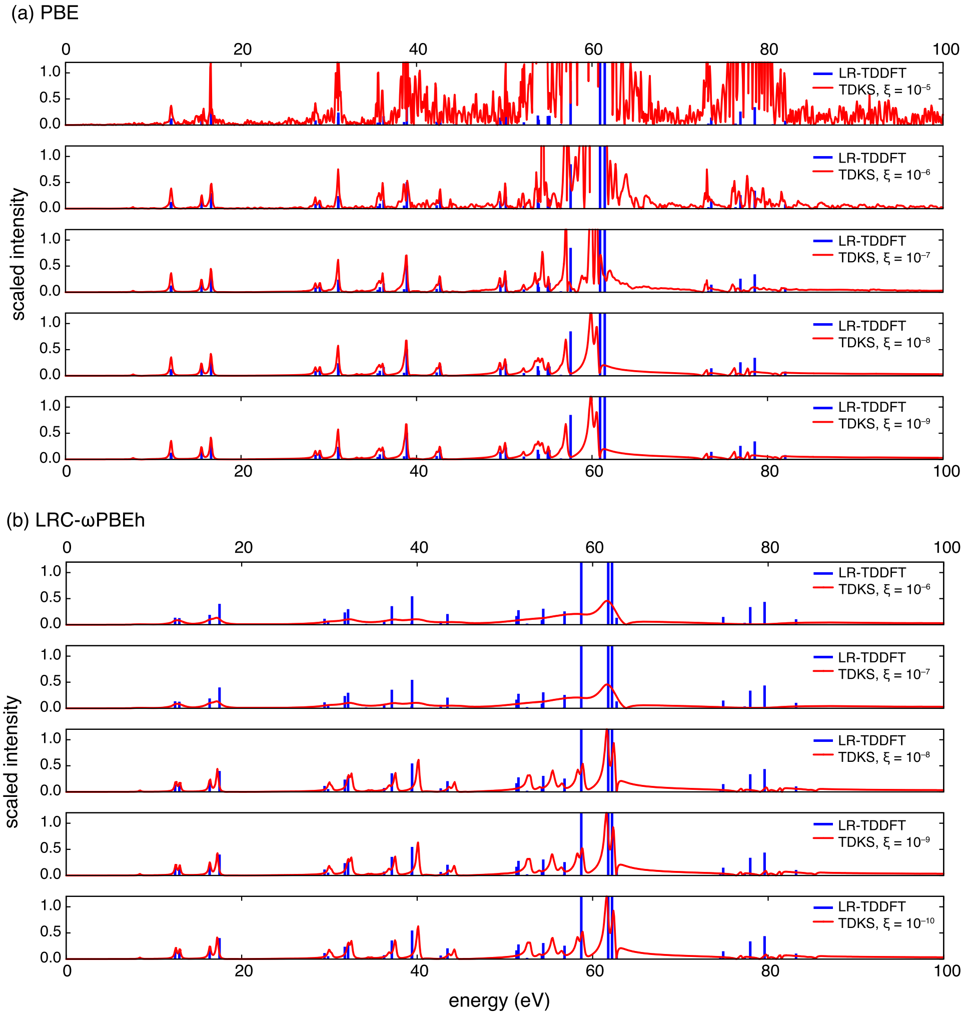

FIG. 8. Comparison of LR-TDDFT stick spectra (in blue) to broadband TDKS spectra (in red) for H_2O molecule. The TDKS spectra were computed using the EP-PC3 algorithm with $\Delta t = 0.5$ a.u., propagated for a total of 5000 a.u., using various thresholds ξ. All TDKS spectra have been scaled by a common, empirical factor in order to match the intensities from the LR-TDDFT calculation.

TABLE VI. Average number of Fock builds per time step as a function of the threshold ξ, for TDKS simulations of H_2O at the PBE/6-31G(d) level using various PC algorithms. Each simulation is propagated in time for a total of 5000 a.u. using Δt = 0.5 a.u.

	Number of Fock builds				
	LFLP-		EP-		
ξ	PC1	PC2	PC1	PC2	PC3
10^{-5}	2.89	2.80	2.00	2.83	2.83
10^{-6}	2.89	2.83	2.00	2.85	2.85
10^{-7}	2.91	2.88	2.00	2.87	2.87
10^{-8}	4.01	3.91	3.62	2.93	2.93
10^{-9}	4.85	4.85	4.14	4.47	3.36

exhibit spurious peaks but converge to clearly resolved spectra for $\xi = 10^{-9}$ (PBE) or $\xi = 10^{-10}$ (LRC-ωPBEh). These converged spectra are in good agreement with LR-TDDFT results for energies $\lesssim$40 eV (PBE) or $\lesssim$30 eV (LRC-ωPBEh), and while there is some peak shifting with respect to LR results at higher energies, this shifting is largely independent of ξ. This observation suggests that it is the finite time step, rather than the threshold, that is the major source of the peak shifting. The controlled convergence with respect to ξ indicates that this parameter can effectively be used to reduce the noise in time-domain dynamics, for GGA functionals, while for LRC functionals, the parameter ξ eliminates spurious damping in the time-domain dynamics. This is explored in Figs. S7 and S8 of the supplementary material, which show how various values of ξ affect the fluctuations in the time-dependent dipole moment.

Table VI shows the average number of Fock builds per time step for the H_2O simulations in Fig. 8(a). Unsurprisingly, this number increases as the threshold is loosened, so it is wise not to set too small a value for ξ. If only the low-energy part of the spectrum is of interest ($\lesssim$30–40 eV), then $\xi = 10^{-7}$ seems to be sufficient for GGA functionals or $\xi = 10^{-8}$ for range-separated hybrid functionals. Smaller values are necessary to eliminate noise at higher energies.

We next study the convergence of spectra with respect to Δt, using the spectrum of a single adenine molecule as a test case. Spectra are plotted in Fig. 9 for time steps ranging from 0.2 to 2.0 a.u., with the Δt = 0.2 a.u. result taken as a reference spectrum. (In Figs. S13 and S14 of the supplementary material, we use time steps as small as 0.02 a.u. to demonstrate that the Δt = 0.2 a.u. spectrum can indeed serve as a reference, for the PC propagator methods. This, again, stands in contrast to MMUT results.) Spectra in Fig. 9 were obtained using the EP-PC3 algorithm but results for other PC algorithms are similar; see Figs. S15–S18 of the supplementary material. Because the TDKS spectrum is obtained from the Fourier transform of the discrete time series representing the induced dipole moment [Eq. (52)], its range of validity is limited to frequencies smaller than the Nyquist frequency, $f_{\rm Ny} = \pi/\Delta t$.[64] Above this, a higher sampling rate is necessary for the time series, or in other words Δt must be reduced. For convenience, Table VII provides Nyquist frequencies (in units of eV) corresponding to values of Δt examined here.

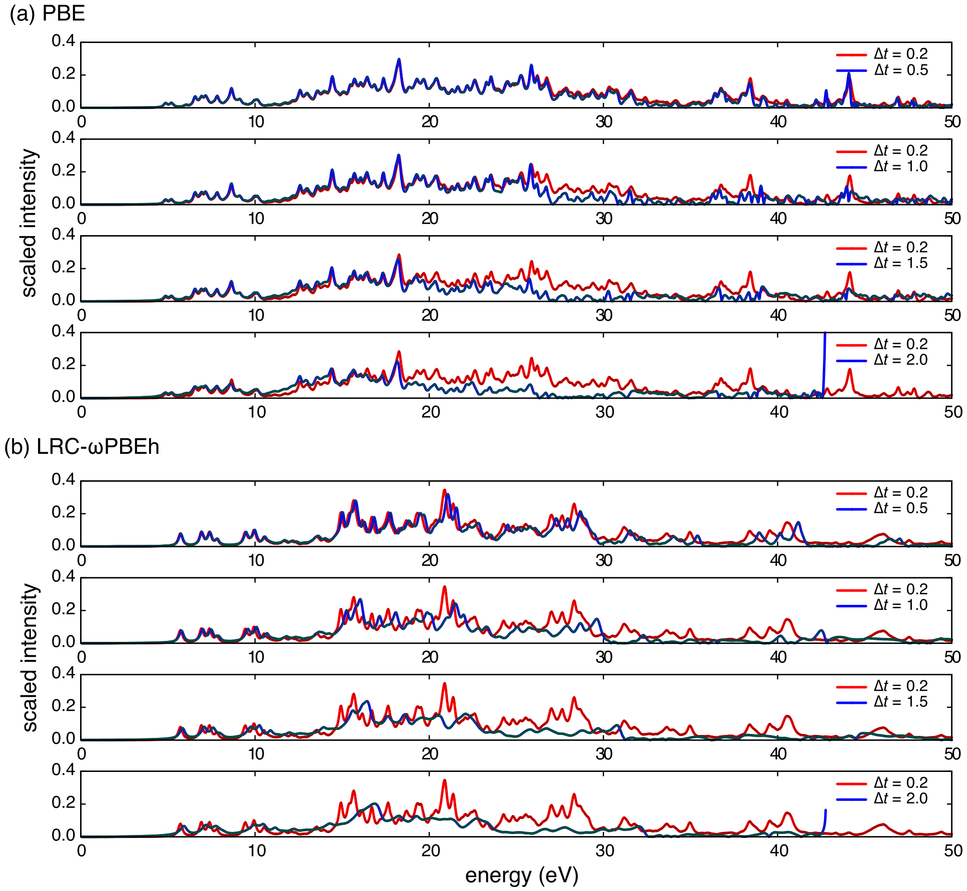

FIG. 9. Absorption spectrum of adenine computed via TDKS simulation using the EP-PC3 algorithm. A reference spectrum (in red), computed using Δt = 0.2 a.u., is compared to spectra obtained using several larger values of Δt. The latter are mostly consistent with the reference spectrum up to frequencies (energies) $\sim f_{\rm Ny}/5$ for PBE or $\sim f_{\rm Ny}/8$ for LRC-ωPBEh. For Δt = 2.0 a.u., we plot the spectrum only out to $f_{\rm Ny}$ ($\approx$43 eV).

TABLE VII. Nyquist frequencies (expressed in energy units) for various values of Δt.

	Energy (eV)		
Δt (a.u.)	f_{Ny}	$f_{\text{Ny}}/5$	$f_{\text{Ny}}/8$
0.2	427	85	53
0.5	171	34	21
1.0	85	17	11
1.5	57	11	7
2.0	43	8	5

Spectra in Fig. 9 obtained using $\Delta t > 0.2$ a.u. are converged to the reference spectrum only at low energies. The larger the time step, the more quickly this agreement deteriorates as energy increases, both in terms of the peak positions and (especially) oscillator strengths. Empirically, we observe that the spectrum is in reasonable agreement with the reference up to energies $E_{\max} \approx f_{\text{Ny}}/5$ for the PBE functional or $E_{\max} \approx f_{\text{Ny}}/8$ for LRC-ωPBEh. This suggests that the time step should be selected such that $\Delta t < \pi/5E_{\max}$ for GGA functionals or $\Delta t < \pi/8E_{\max}$ for global or range-separated hybrid functionals, where $E_{\max}$ is the largest excitation energy of interest. Examining Table VII, we see that a rather large time step of $\Delta t = 1.0$–2.0 a.u. (depending on whether the functional contains Hartree-Fock exchange) suffices for computing spectra up to about 9 eV, which covers the optical and much of the UV region of the electromagnetic spectrum.

IV. CONCLUSIONS

We have introduced several predictor/corrector methods for efficient propagation of the TDKS equation in real time. The first type of algorithm, which we call LFLP-PC, is based on linear extrapolation and interpolation of the Fock matrix using a midpoint rule and may be regarded as a self-consistent version of the MMUT approach that is already in widespread use.[33,34] A second type of algorithm, EP-PC, uses an exponential extrapolation of the density matrix with trapezoidal integration of the propagator. Both methods use a one-term truncation of the Magnus expansion and all matrix exponentials are computed exactly, via diagonalization, which is appropriate in Gaussian basis sets or other representations where the dimension of the Hamiltonian is small. (This aspect of our algorithms would need to be modified for plane-wave or real-space grid implementations of DFT, where the dimension of the Hamiltonian matrix precludes exact diagonalization.)

The self-consistent PC approach improves the stability and computational efficiency of the density matrix propagation, and for weak-field dynamics, we are able to use time steps as large as $\Delta t = 1.0$–1.5 a.u. while retaining numerical stability and energy-conserving dynamics. Although the MMUT algorithm requires only a single Fock matrix construction per time step, whereas the self-consistent approaches require more than one, the MMUT approach is restricted to the use of time steps no larger than 0.2–0.5 a.u.,[35,39–41] and sometimes as small as 0.05 a.u.,[36–38] and overall the PC algorithms prove to be a net "win," reducing the average number of Fock builds per unit of simulation time. For most applications, the cost of TDKS calculations in Gaussian basis sets is overwhelmingly dominated by the cost of Fock matrix construction, so this reduction in the number of Fock builds translates directly into speedup of the simulations. The very small time interval between the predictor and corrector steps (a fraction of Δt) also means that the additional Fock builds can make good use of "incremental" techniques for Fock matrix construction,[34,65,66] leading to additional speedup. Consequently, the PC algorithms are no worse than equally efficient as compared to MMUT, and often several times more efficient, while at the same time providing tools that allow one to check on-the-fly for any divergence of the propagated dynamics in a way that is not possible using MMUT. The latter point is critical but is often overlooked. Although catastrophic divergence of a TDKS simulation can be detected simply by monitoring (lack of) energy conservation, it is possible for a simulation to conserve energy yet afford an absorption spectrum that differs from the LR-TDDFT result, even for a weakly perturbing external field. Assuming that the matrix exponentials are evaluated exactly, this behavior—if and when it occurs—is a manifestation of error accumulation due to finite truncation of the Magnus expansion combined with a time step that is too large.

We have discussed three factors that influence convergence of the iterative PC procedure. First, Eq. (24) affords a rigorous upper bound for how large Δt can be in a convergent simulation, although this formal result is difficult to evaluate in practice. For weak-field perturbations where the electron density oscillates around the ground-state density, however, the optimal time step can be determined at the beginning of the calculation using short test runs with different values of Δt. For strong-field perturbations, the electron density will deviate far from that of the ground state, but self-consistent iteration in the PC approach acts as an on-the-fly self-examination that can flag an overly large time step. This can be used to halt the simulation immediately, or else to reduce Δt, whenever the predicted and corrected Fock or density matrices fail to converge to one another.

A second quantity that influences convergence is $\|\delta\mathbf{F}\|_{\text{F}}$, which measures the magnitude of the fluctuations in the Fock matrix and provides as an indicator of the convergence (or lack thereof) of the Magnus expansion; see Eq. (35). This quantity is easily monitored and the fluctuations in $\mathbf{F}(t)$ over a given time step determine whether one has a valid approximation to the time-ordered integral that appears in the time evolution operator. Strong-field dynamics are characterized by large values of $\|\delta\mathbf{F}\|_{\text{F}}$ and Δt must be reduced accordingly.

Finally, the discrete nature of the time series that represents the electron dynamics places an upper bound on the absorption spectrum that is governed by the time step and the associated Nyquist frequency (f_{Ny}) arising from the discrete Fourier transform of the dipole fluctuations. In practice, we find that the spectrum is reasonably well converged for energies $E_{\max} \lesssim f_{\text{Ny}}/5$ (in the case of GGA functionals) or $E_{\max} \lesssim f_{\text{Ny}}/8$ (for hybrid functionals).

SUPPLEMENTARY MATERIAL

See supplementary material for additional tests and data for other parameter sets.

ACKNOWLEDGMENTS

This work was supported by National Science Foundation Grant No. CHE-1300603, with some preliminary work supported by the U.S. Department of Energy, Office of Basic Energy Sciences, Division of Chemical Sciences, Geosciences, and Biosciences under Award No. DE-SC0008550. Calculations were performed at the Ohio Supercomputer Center under Project No. PAA-0003.[67] J.M.H. is a fellow of the Alexander von Humboldt Foundation and serves on the Board of Directors of Q-Chem, Inc.

APPENDIX: CONVERGENCE OF THE PC STEP

This appendix presents a derivation of Eq. (24), which demonstrates that the PC iterations (step 5 in Figs. 1 and 2) necessarily converge if Δt is sufficiently small, and furthermore provides an explicit upper bound on how large Δt can be in order to obtain convergence. For definiteness, we consider the EP-PC algorithm but the proof for LFLP-PC is similar and the upper bound [Eq. (24)] is the same in either case.

Denote the construction of the Fock matrix as a mapping $\mathcal{F}$: $\mathbf{P} \mapsto \mathbf{F}$. Denote by $\mathcal{U} : \mathbf{F} \mapsto \mathbf{U}(t_{N+1}, t_N)$, the mapping between the Fock matrix and the propagator, for $t \in [t_N, t_{N+1}]$. Strictly speaking, $\mathcal{U}$ is a functional mapping because its involves integration of $\mathbf{F}(t)$ over the interval $[t_N, t_{N+1}]$. Within the adiabatic approximation to TDDFT, however, the exchange-correlation functional lacks memory and this simplifies to a mapping between complex-valued $n \times n$ matrices. Consistency between predicted and corrected matrices requires the existence of a solution to a fixed-point problem via self-iteration. To be specific, the problem of finding self-consistent values of $\mathbf{P}^{\text{p}}_{N+1}$ and $\mathbf{P}^{\text{c}}_{N+1}$ requires $\mathbf{P}_{N+1}$ to reach a solution $\mathbf{P}^{\star}_{N+1}$ that satisfies the equality

$$\mathbf{P}^{\star}_{N+1} = [\mathcal{U}\circ\mathcal{F}(\mathbf{P}^{\star}_{N+1})]\mathbf{P}_N[\mathcal{U}\circ\mathcal{F}(\mathbf{P}^{\star}_{N+1})]^{\dagger} . \tag{A1}$$

The notation $\mathcal{U}\circ\mathcal{F}$ indicates a compound mapping. In the first instance, for example, it means using the mapping $\mathcal{F}$ to build $\mathbf{F}_{N+1}$ based on $\mathbf{P}^{\star}_{N+1}$ then using $\mathcal{U}$ to build $\mathbf{U}_N$ based on $\mathbf{F}_{N+1}$.

Since the right side of Eq. (A1) can be regarded as a mapping of $\mathbf{P}^{\star}_{N+1}$ as a whole, and that mapping equals the variable $\mathbf{P}^{\star}_{N+1}$ itself, Eq. (A1) represents a fixed-point problem, i.e., a problem of the general form $x = f(x)$. Denote such a mapping $\mathcal{Y}$: $\mathbf{P}_{N+1} \mapsto \mathbf{P}_{N+1}$ as

$$\begin{aligned}\mathcal{Y}(\mathbf{P}_{N+1}) &= \exp\{-i(\Delta t/2)[\mathcal{F}(\mathbf{P}_{N+1})+\mathbf{F}_N]\}\\ &\quad\times\mathbf{P}_N\exp\{i(\Delta t/2)[\mathcal{F}(\mathbf{P}_{N+1})+\mathbf{F}_N]\},\end{aligned} \tag{A2}$$

where $\mathbf{F}_N$ and $\mathbf{P}_N$ are known values from the previous time step. The solution $\mathbf{P}^{\star}_{N+1}$ to the fixed-point problem in Eq. (A1) is approached by self-mapping (self-iteration) such that a new $\mathbf{P}_{N+1}$ is updated using the value of $\mathcal{Y}(\mathbf{P}_{N+1})$, until $\mathbf{P}_{N+1}$ equals $\mathcal{Y}(\mathbf{P}_{N+1})$.

For self-mapping procedures such as these, the existence of a convergent solution can be estimated by using the Banach fixed-point theorem.[68] Consider a first-order expansion of Eq. (A2),

$$\begin{aligned}\mathcal{Y}(\mathbf{P}_{N+1}) = &\left\{\mathbf{1}-\frac{i\Delta t}{2}[\mathcal{F}(\mathbf{P}_{N+1})+\mathbf{F}_N]+\mathcal{O}[(\Delta t)^2]\right\}\mathbf{P}_N\\ &\times\left\{\mathbf{1}+\frac{i\Delta t}{2}[\mathcal{F}(\mathbf{P}_{N+1})+\mathbf{F}_N]+\mathcal{O}[(\Delta t)^2]\right\}.\end{aligned} \tag{A3}$$

Define the distance between two matrices $\mathbf{B}_1$ and $\mathbf{B}_2$ as

$$d(\mathbf{B}_1,\mathbf{B}_2) = \|\mathbf{B}_1-\mathbf{B}_2\|_{\text{F}} . \tag{A4}$$

According to the Banach fixed-point theorem, convergence of the self-mapping problem requires the existence of $q \in [0, 1)$ such that

$$d\left(\mathcal{Y}(\mathbf{P}^{[k]}_{N+1}),\mathcal{Y}(\mathbf{P}^{[k+1]}_{N+1})\right) \le q\,d\left(\mathbf{P}^{[k]}_{N+1},\mathbf{P}^{[k+1]}_{N+1}\right), \tag{A5}$$

where the superscript in $\mathbf{P}^{[k]}_{N+1}$ refers to $\mathbf{P}_{N+1}$ obtained after k iterations. Equation (A5) requires that for each iteration, the mapping is a contraction that shrinks the distance between the points $\mathbf{P}^{[k]}_{N+1}$ and $\mathbf{P}^{[k+1]}_{N+1}$ by a uniform fraction $q < 1$, such that eventually the sequence $\{\mathbf{P}^{[0]}_{N+I}, P^{[1]}_{N+I}, P^{[2]}_{N+I}, \cdots\}$ converges to the solution $\mathbf{P}^{\star}_{N+1}$. The smaller q is, the faster the rate of convergence.

Next substitute Eq. (A3) into Eq. (A5) and denote

$$\delta\mathbf{F}^{[k]}_{N+1} = \mathcal{F}(\mathbf{P}^{[k]}_{N+1}) - \mathcal{F}(\mathbf{P}^{[k+1]}_{N+1}) \tag{A6}$$

and

$$\delta\mathbf{P}^{[k]}_{N+1} = \mathbf{P}^{[k]}_{N+1} - \mathbf{P}^{[k+1]}_{N+1} . \tag{A7}$$

For the first order terms, the Banach theorem affords a condition that

$$\frac{\Delta t}{2}\left\|\mathbf{P}_N\,\delta\mathbf{F}^{[k]}_{N+1} - \delta\mathbf{F}^{[k]}_{N+1}\,\mathbf{P}_N\right\|_{\text{F}} \le q\left\|\delta\mathbf{P}^{[k]}_{N+1}\right\|_{\text{F}}. \tag{A8}$$

The Frobenius norm is a scalar, so we can move $\|\delta\mathbf{P}^{[k]}_{N+1}\|_{\text{F}}$ from the right to the left side in Eq. (A8) and thus rewrite this inequality, using a commutator, as

$$\frac{\Delta t}{2}\left\|\left[\mathbf{P}_N,\ \frac{\delta\mathbf{F}^{[k]}_{N+1}}{\|\delta\mathbf{P}^{[k]}_{N+1}\|_{\text{F}}}\right]\right\|_{\text{F}} \le q\ . \tag{A9}$$

As $\mathbf{P}^{[k]}_{N+1}$ approaches the solution $\mathbf{P}^{\star}_{N+1}$, the quantity $\delta\mathbf{P}^{[k]}_{N+1}$ becomes small. By considering the limit $\delta\mathbf{P}^{[k]}_{N+1} \to 0$, the second term in the commutator of Eq. (A9) can be simplified into a form reminiscent of a directional derivative,

$$\begin{aligned}\left(\frac{\delta\mathbf{F}^{[k]}_{N+1}}{\|\delta\mathbf{P}^{[k]}_{N+1}\|_{\mathrm{F}}}\right)_{ij} \approx \frac{\delta F_{ij}}{\|\delta\mathbf{P}\|_{\mathrm{F}}}\bigg|_{\mathbf{P}\to\mathbf{P}^{\star}_{N+1}} &= \lim_{\mathbf{P}^{[k]}_{N+1}\to\mathbf{P}^{[k+1]}_{N+1}} \frac{\left[\mathcal{F}(\mathbf{P}^{[k]}_{N+1}) - \mathcal{F}(\mathbf{P}^{[k+1]}_{N+1})\right]_{ij}}{\|\delta\mathbf{P}^{[k]}_{N+1}\|} \\ &= \lim_{\mathbf{P}^{[k]}_{N+1}\to\mathbf{P}^{[k+1]}_{N+1}} \frac{\sum_{pq}\left(\frac{\partial\mathcal{F}_{ij}}{\partial P_{pq}}\right)[(\mathbf{P}^{[k]}_{N+1})_{pq} - (\mathbf{P}^{[k+1]}_{N+1})_{pq}]}{\|\delta\mathbf{P}^{[k]}_{N+1}\|} \\ &= \lim_{\mathbf{P}^{[k]}_{N+1}\to\mathbf{P}^{[k+1]}_{N+1}} \frac{\sum_{pq}\left(\frac{\partial\mathcal{F}_{ij}}{\partial P_{pq}}\right)(\delta\mathbf{P}^{[k]}_{N+1})_{pq}}{\|\delta\mathbf{P}^{[k]}_{N+1}\|} \\ &= \sum_{pq}\frac{(\delta\mathbf{P})_{pq}}{\|\delta\mathbf{P}\|}\frac{\partial F_{ij}}{\partial P_{pq}}\bigg|_{\mathbf{P}\to\mathbf{P}^{\star}_{N+1}} = \left(\hat{\boldsymbol{\nabla}}_{\mathbf{P}}\,\mathbf{F}\Big|_{\mathbf{P}\to\mathbf{P}^{\star}_{N+1}}\right)_{ij}\ . \qquad \text{(A10)}\end{aligned}$$

In obtaining the second equality in Eq. (A10), we have used a first-order Taylor expansion around $\mathbf{P}^{\star}_{N+1}$. The penultimate line in this expression, which defines the directional derivative in the final line, can be viewed as a vector dot product between a direction vector $(\delta\mathbf{P})/\|\mathbf{P}\|$ and a gradient $\partial\mathbf{F}/\partial\mathbf{P}$.

Subject to the approximation at the start of Eq. (A10), Eq. (A9) now can be expressed as

$$\frac{\Delta t}{2}\left\|\left[\mathbf{P}_N,\ \hat{\boldsymbol{\nabla}}_{\mathbf{P}}\,\mathbf{F}\Big|_{\mathbf{P}\to\mathbf{P}^{\star}_{N+1}}\right]\right\|_{\mathrm{F}} \lesssim q\ . \qquad \text{(A11)}$$

The terms inside of $\|\cdots\|_{\mathrm{F}}$ in Eq. (A11) are determined by the properties of the system, insofar as $\mathbf{P}_N$ depends on the electronic dynamics, which is driven by the external field, and $\hat{\boldsymbol{\nabla}}_{\mathbf{P}}\mathbf{F}$ depends on the derivative of the Fock matrix with respect to the future density matrix $\mathbf{P}_{N+1}$. The latter is numerically approached in the direction $\delta\mathbf{P}_{N+1}$. Considering that $q < 1$, the existence of a convergence solution requires that the upper-bound condition quoted in Eq. (24) be satisfied. The maximum value of Δt is therefore determined by the system and its dynamical properties. By adjusting Δt to be larger (within the convergence region), the rate of convergence is decreased which means more iterations per time step are needed to reach consistency within the predictor/corrector scheme and vice versa.

[1] L. Blancafort and A. Migani, J. Am. Chem. Soc. **129**, 14540 (2007).

[2] B. E. Hardin, E. T. Hoke, P. B. Armstrong, J.-H. Yum, P. Comte, T. Torres, J. M. J. Frechet, M. K. Nazeeruddin, M. Gratzel, and M. D. McGehee, Nat. Photonics **3**, 406 (2009).

[3] A. Hagfeldt, G. Boschloo, L. Sun, L. Kloo, and H. Pettersson, Chem. Rev. **110**, 6595 (2010).

[4] P. Salières, L. L. Déroff, T. Auguste, P. Monot, P. d'Oliveira, D. Campo, J. F. Hergott, H. Merdji, and B. Carré, Phys. Rev. Lett. **83**, 5483 (1999).

[5] H. Sakai, J. J. Larsen, I. Wendt-Larsen, J. Olesen, P. B. Corkum, and H. Stapelfeldt, Phys. Rev. A **67**, 063404 (2003).

[6] K. J. Schafer, B. Yang, L. F. DiMauro, and K. C. Kulander, Phys. Rev. Lett. **70**, 1599 (1993).

[7] P. M. Paul, E. S. Toma, P. Breger, G. Mullot, F. Augé, P. Balcou, H. G. Muller, and P. Agostini, Science **292**, 1689 (2001).

[8] K. Ramasesha, S. R. Leone, and D. M. Neumark, Annu. Rev. Phys. Chem. **67**, 41 (2016).

[9] A. R. Attar, A. Bhattacherjee, C. D. Pemmaraju, K. Schnorr, K. D. Closser, D. Prendergast, and S. R. Leone, Science **356**, 54 (2017).

[10] S. Zigo, A.-T. Le, P. Timilsina, and C. A. Trallero-Herrero, Sci. Rep. **7**, 42149 (2017).

[11] M. Sharifi, F. Kong, S. L. Chin, H. Mineo, Y. Dyakov, A. M. Mebel, S. D. Chao, M. Hayashi, and S. H. Lin, J. Phys. Chem. A **111**, 9405 (2007).

[12] B. Winter, Nucl. Instrum. Methods Phys. Res., Sect. A **601**, 139 (2009).

[13] M. Faubel, K. R. Siefermann, Y. Liu, and B. Abel, Acc. Chem. Res. **45**, 120 (2012).

[14] E. Gross and W. Kohn, Adv. Quantum Chem. **21**, 255 (1990).

[15] M. A. L. Marques and E. K. U. Gross, Annu. Rev. Phys. Chem. **55**, 427 (2004).

[16] D. Rappoport and F. Furche, "Excited states and photochemistry," in *Time-Dependent Density Functional Theory*, Volume 706 of Lecture Notes in Physics, edited by M. A. L. Marques, C. A. Ullrich, F. Nogueira, A. Rubio, K. Burke, and E. K. U. Gross (Springer, 2006), Chap. 23, pp. 337–357.

[17] M. E. Casida, "Time-dependent density functional response theory for molecules," in *Recent Advances in Density Functional Methods, Part I*, Volume I of Recent Advances in Computational Chemistry, edited by D. P. Chong (World Scientific, River Edge, NJ, 1995), Chap. 5, pp. 155–192.

[18] C. A. Ullrich and A. D. Bandrauk, "Atoms and molecules in strong laser fields," in *Time-Dependent Density Functional Theory*, Volume 706 of Lecture Notes in Physics, edited by M. A. L. Marques, C. A. Ullrich, F. Nogueira, A. Rubio, K. Burke, and E. K. U. Gross (Springer-Verlag, Berlin, 2006), Chap. 24, pp. 357–375.

[19] C. A. Ullrich and A. D. Bandrauk, "Atoms and molecules in strong laser fields," in *Fundamentals of Time-Dependent Density Functional Theory*, Volume 837 of Lecture Notes in Physics, edited by M. A. L. Marques, N. T. Maitra, F. M. S. Nogueira, E. K. U. Gross, and A. Rubio (Springer-Verlag, Berlin, 2012), Chap. 18, pp. 351–371.

[20] Y. Takimoto, F. D. Vila, and J. J. Rehr, J. Chem. Phys. **127**, 154114 (2007).

[21] X. Andrade, S. Botti, M. A. L. Marques, and A. Rubio, J. Chem. Phys. **126**, 184106 (2007).

[22] F. Ding, B. E. Van Kuiken, B. E. Eichinger, and X. Li, J. Chem. Phys. **138**, 064104 (2013).

[23] R. E. Stratmann, G. E. Scuseria, and M. J. Frisch, J. Chem. Phys. **109**, 8218 (1998).

[24] M. Gao, S. Paul, C. D. Schwieters, Z.-Q. You, H. Shao, J. M. Herbert, J. R. Parquette, and C. P. Jaroniec, J. Phys. Chem. C **119**, 13948 (2015).

[25] A. Bruner, D. LaMaster, and K. Lopata, J. Chem. Theory Comput. **12**, 3741–3750 (2016).

[26] J. Theilhaber, Phys. Rev. B **46**, 12990 (1992).

[27] K. Yabana and G. F. Bertsch, Phys. Rev. B **54**, 4484 (1996).

[28] K. Yabana and G. F. Bertsch, Int. J. Quantum Chem. **75**, 55 (1999).

[29] M. R. Provorse and C. M. Isborn, Int. J. Quantum Chem. **116**, 739 (2016).

[30] J. J. Goings, P. J. Lestrange, and X. Li, Wiley Interdiscip. Rev.: Comput. Mol. Sci. **8**, e1341 (2018).

[31] X. Andrade, D. Strubbe, U. De Giovannini, A. H. Larsen, M. J. T. Oliveira, J. Alberdi-Rodriguez, A. Varas, I. Theophilou, N. Helbig, M. J. Verstraete, L. Stella, F. Nogueira, A. Aspuru-Guzik, A. Castro, M. A. L. Marques, and A. Rubio, Phys. Chem. Chem. Phys. **17**, 31371 (2015).

[32] A. Schleife, E. W. Draeger, Y. Kanai, and A. A. Correa, J. Chem. Phys. **137**, 22A546 (2012).

[33] X. Li, S. M. Smith, A. N. Markevitch, D. A. Romanov, R. J. Levis, and H. B. Schlegel, Phys. Chem. Chem. Phys. **7**, 233 (2005).

[34] W. Liang, C. T. Chapman, and X. Li, J. Chem. Phys. **134**, 184102 (2011).

[35] K. Lopata and N. Govind, J. Chem. Theory Comput. **7**, 1344 (2011).

[36] T. S. Nguyen and J. Parkhill, J. Chem. Theory Comput. **11**, 2918 (2015).

[37] R. G. Fernando, M. C. Balhoff, and K. Lopata, J. Chem. Theory Comput. **11**, 646 (2015).

[38] J. J. Goings and X. Li, J. Chem. Phys. **144**, 234102 (2016).

[39] R. Nadler and J. F. Sanz, Theor. Chem. Acc. **132**, 1342 (2013).

[40] A. Sissay, P. Abanador, F. Mauger, M. Gaarde, K. J. Schafer, and K. Lopata, J. Chem. Phys. **145**, 094105 (2016).

[41] S. A. Fischer, C. J. Cramer, and N. Govind, J. Phys. Chem. Lett. **7**, 1387 (2016).

[42] R. Shankar, *Principles of Quantum Mechanics*, 2nd ed. (Springer Science & Business Media, 2012).

[43] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, 2007).

[44] S. Blanes, F. Casas, J. A. Oteo, and J. Ros, Phys. Rep. **470**, 151 (2009).

[45] N. O. Foglia, U. N. Morzan, D. A. Estrin, D. A. Scherlis, and M. C. G. Lebrero, J. Chem. Theory Comput. **13**, 77 (2017).

[46] A. Castro, M. A. L. Marques, and A. Rubio, J. Chem. Phys. **121**, 3425 (2004).

[47] F. Casas and A. Iserles, J. Phys. A: Math. Gen. **39**, 5445 (2006).

[48] C.-L. Cheng, J. S. Evans, and T. Van Voorhis, Phys. Rev. B **74**, 155112 (2006).

[49] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd ed. (Cambridge University Press, New York, NY, USA, 2012).

[50] M. Suzuki, Proc. Jpn. Acad., Ser. B **69**, 161 (1993).

[51] Y. Shao, Z. Gan, E. Epifanovsky, A. T. B. Gilbert, M. Wormit, J. Kussmann, A. W. Lange, A. Behn, J. Deng, X. Feng, D. Ghosh, M. Goldey, P. R. Horn, L. D. Jacobson, I. Kaliman, R. Z. Khaliullin, T. Kús, A. Landau, J. Liu, E. I. Proynov, Y. M. Rhee, R. M. Richard, M. A. Rohrdanz, R. P. Steele, E. J. Sundstrom, H. L. Woodcock, III, P. M. Zimmerman, D. Zuev, B. Albrecht, E. Alguire, B. Austin, G. J. O. Beran, Y. A. Bernard, E. Berquist, K. Brandhorst, K. B. Bravaya, S. T. Brown, D. Casanova, C.-M. Chang, Y. Chen, S. H. Chien, K. D. Closser, D. L. Crittenden, M. Diedenhofen, R. A. DiStasio, Jr., H. Do, A. D. Dutoi, R. G. Edgar, S. Fatehi, L. Fusti-Molnar, A. Ghysels, A. Golubeva-Zadorozhnaya, J. Gomes, M. W. D. Hanson-Heine, P. H. P. Harbach, A. W. Hauser, E. G. Hohenstein, Z. C. Holden, T.-C. Jagau, H. Ji, B. Kaduk, K. Khistyaev, J. Kim, J. Kim, R. A. King, P. Klunzinger, D. Kosenkov, T. Kowalczyk, C. M. Krauter, K. U. Lao, A. Laurent, K. V. Lawler, S. V. Levchenko, C. Y. Lin, F. Liu, E. Livshits, R. C. Lochan, A. Luenser, P. Manohar, S. F. Manzer, S.-P. Mao, N. Mardirossian, A. V. Marenich, S. A. Maurer, N. J. Mayhall, C. M. Oana, R. Olivares-Amaya, D. P. O'Neill, J. A. Parkhill, T. M. Perrine, R. Peverati, P. A. Pieniazek, A. Prociuk, D. R. Rehn, E. Rosta, N. J. Russ, N. Sergueev, S. M. Sharada, S. Sharma, D. W. Small, A. Sodt, T. Stein, D. Stück, Y.-C. Su, A. J. W. Thom, T. Tsuchimochi, L. Vogt, O. Vydrov, T. Wang, M. A. Watson, J. Wenzel, A. White, C. F. Williams, V. Vanovschi, S. Yeganeh, S. R. Yost, Z.-Q. You, I. Y. Zhang, X. Zhang, Y. Zhao, B. R. Brooks, G. K. L. Chan, D. M. Chipman, C. J. Cramer, W. A. Goddard, III, M. S. Gordon, W. J. Hehre, A. Klamt, H. F. Schaefer, III, M. W. Schmidt, C. D. Sherrill, D. G. Truhlar, A. Warshel, X. Xu, A. Aspuru-Guzik, R. Baer, A. T. Bell, N. A. Besley, J.-D. Chai, A. Dreuw, B. D. Dunietz, T. R. Furlani, S. R. Gwaltney, C.-P. Hsu, Y. Jung, J. Kong, D. S. Lambrecht, W. Liang, C. Ochsenfeld, V. A. Rassolov, L. V. Slipchenko, J. E. Subotnik, T. Van Voorhis, J. M. Herbert, A. I. Krylov, P. M. W. Gill, and M. Head-Gordon, Mol. Phys. **113**, 184 (2015).

[52] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[53] M. A. Rohrdanz, K. M. Martins, and J. M. Herbert, J. Chem. Phys. **130**, 054112 (2009).

[54] P. Krause and H. B. Schlegel, J. Phys. Chem. Lett. **6**, 2140 (2015).

[55] A. F. White, C. J. Heide, P. Saalfrank, M. Head-Gordon, and E. Luppi, Mol. Phys. **114**, 947 (2016).

[56] K. Lopata and N. Govind, J. Chem. Theory Comput. **9**, 4939 (2013).

[57] E. Luppi and M. Head-Gordon, Mol. Phys. **110**, 909 (2012).

[58] E. Luppi and M. Head-Gordon, J. Chem. Phys. **139**, 164121 (2013).

[59] Because the perturbation is weak, the response is in the linear regime and the induced dipole moment is determined by the first-order term in an expansion with respect to field strength: $\delta\mu_j(\omega) = \alpha_{jk}E_k + \mathcal{O}(E_k^2)$. The spectrum is actually computed using only a single perturbation, in all three direction at once, which is equivalent to the summation in Eq. (51).

[60] R. W. Hamming, *Numerical Methods for Scientists and Engineers*, 2nd ed. (Dover Publications, Inc., New York, NY, USA, 1986).

[61] M. Guidon, F. Schiffmann, J. Hutter, and J. VandeVondele, J. Chem. Phys. **128**, 021404 (2008).

[62] N. Luehr, T. E. Markland, and T. J. Martínez, J. Chem. Phys. **140**, 084116 (2014).

[63] S. Tussupbayev, N. Govind, K. Lopata, and C. J. Cramer, J. Chem. Theory Comput. **11**, 1102 (2015).

[64] A. V. Oppenheim, *Discrete-Time Signal Processing*, 3rd ed. (Pearson Education, Upper Saddle River, New Jersey, 1999).

[65] E. Schwegler, M. Challacombe, and M. Head-Gordon, J. Chem. Phys. **106**, 9708 (1997).

[66] S. T. Brown and J. Kong, Chem. Phys. Lett. **408**, 395 (2005).

[67] Ohio Supercomputer Center, http://osc.edu/ark:/19495/f5s1ph73.

[68] M. A. Khamsi and W. A. Kirk, *An Introduction to Metric Spaces and Fixed Point Theory* (John Wiley & Sons, 2011), Vol. 53.