

Time-Dependent DFT in real time

First-principles simulations of materials with



CECAM Flagship School, 28th June - 2nd July 2021

Time-Dependent Density Functional Theory (TD-DFT)

Background

- TD-DFT is an extension of DFT to time-dependent external potential

$$\hat{V}_{ext}(r) \iff n(r)$$

Hohenberg-Kohn (DFT)

$$\hat{V}_{ext}(r, t) \iff n(r, t)$$

Runge-Gross (TD-DFT)

- Perturbative treatment (Linear-Response)

$$V_{ext}(r, t) = V_{ext}(r) + \delta V_{ext}(r, t)$$

Gross-Kohn (TD-DFT + LR)

$$\delta V_{ext}(r, t) \ll V_{ext}(r)$$

Casida formalism

$$\delta\rho = \chi \delta V_{ext}$$

Polarizability

Time-Dependent Density Functional Theory (TD-DFT)

Background

Kohn-Sham (DFT)

$$\hat{V}_{ext}(r) \iff n(r)$$

$$n(r) = \sum_i^N |\psi_i^{KS}(r)|^2$$

$$\hat{H}^{KS} \psi_j^{KS}(r) = \epsilon_j^{KS} \psi_j^{KS}(r)$$

Runge-Gross (TD-DFT)

$$\hat{V}_{ext}(r, t) \iff n(r, t)$$

$$n(r, t) = \sum_i^N |\psi_i^{KS}(r, t)|^2$$

$$\hat{H}^{KS} \psi_j^{KS}(r, t) = i \frac{\partial}{\partial t} \psi_j^{KS}(r, t)$$

The Hamiltonian is determined as for KS... with the caveat that the xc comes from the functional derivative of an xc-Action (unknown). In practice: **adiabatic approximation** (i.e. take instantaneous xc-functionals from DFT)

Time-Dependent Density Functional Theory (TD-DFT)

Real-Time propagation

$$\hat{H}^{KS} \psi_j^{KS}(r, t) = i \frac{\partial}{\partial t} \psi_j^{KS}(r, t)$$

Yabana-Bertsch

$$\Psi(t + \Delta t) = U(t + \Delta t, t) \Psi(t) = T e^{-i \int_t^{t+\Delta t} H(t') dt'} \Psi(t)$$

LCAO formalism...

$$|\psi_i(r, t)\rangle = \sum_{\mu} c_{i,\mu}(t) |\mu\rangle \quad \hat{H}(t) \sum_{\mu} c_{j\mu}(t) |\mu\rangle = i \frac{\partial}{\partial t} \sum_{\mu} c_{j\mu}(t) |\mu\rangle$$

(When the basis is time-independent)

$$i \frac{\partial c_j(t)}{\partial t} = S^{-1} H(t) c_j(t)$$

First Implementation in SIESTA:

Optical properties for molecular systems

A. Tsolakidis, Daniel Sánchez-Portal & R.M. Martin, *Phys. Rev. B* **66**, 235416 (2002)

$$i \frac{\partial c_j(t)}{\partial t} = S^{-1} H(t) c_j(t) \quad U(t + \Delta t, t) = e^{-i S^{-1} H(t) \Delta t}$$

- Crank-Nicholson operator

$$c(t + \Delta t) = \frac{1 - i S^{-1} H(t) \Delta t / 2}{1 + i S^{-1} H(t) \Delta t / 2} c(t) + \mathcal{O}(\Delta t^2)$$

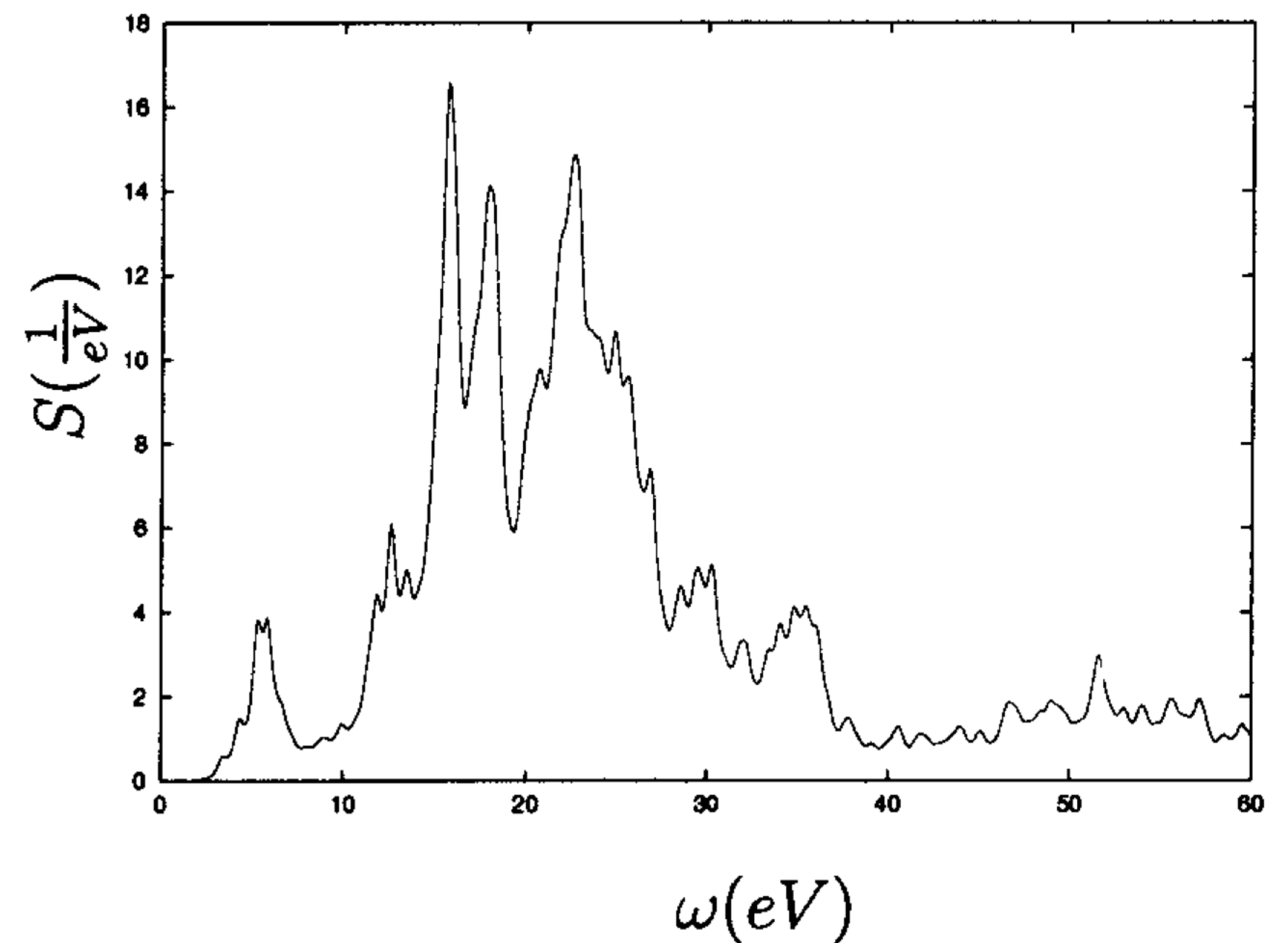
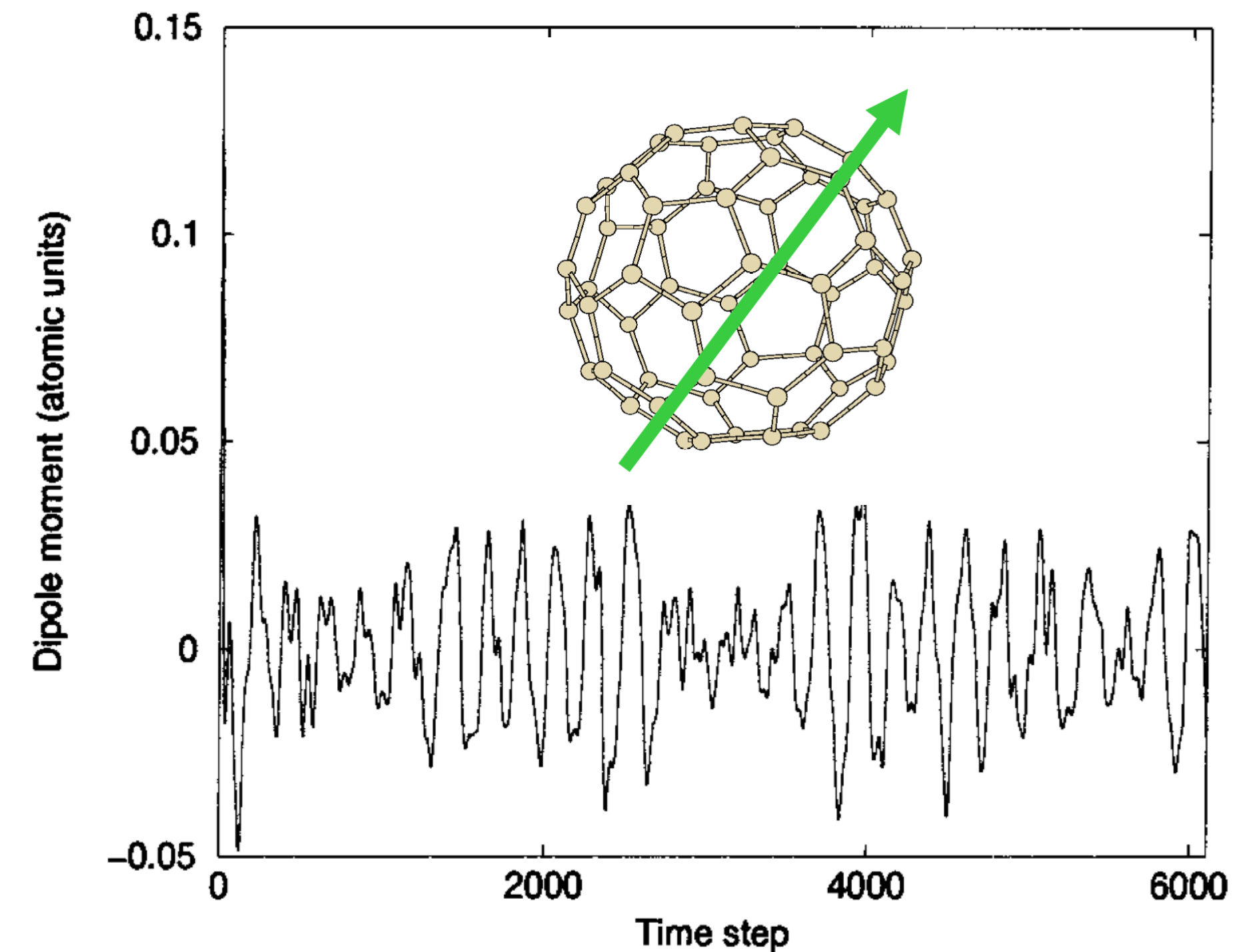
- ▶ Preserves orthonormality of states
- ▶ Explicitly time reversal invariant
- ▶ Conserves the total energy

- Other propagators are possible

- ▶ Leap-frog integration

$$c(t + \Delta t) = c(t - \Delta t) - 2i S^{-1} H(t) c(t) \Delta t + \mathcal{O}(\Delta t^2)$$

- ▶ Lanczos exponential approximation, different polynomial expansions...



Optical properties for molecular systems

Basic idea

Yabana-Bertsch, *Phys. Rev. B* **54**, 4484 (1996)

- Apply an external electric field

$$V_{ext}(r, t) = eE(t) \cdot z$$

- Compute the induced polarisation

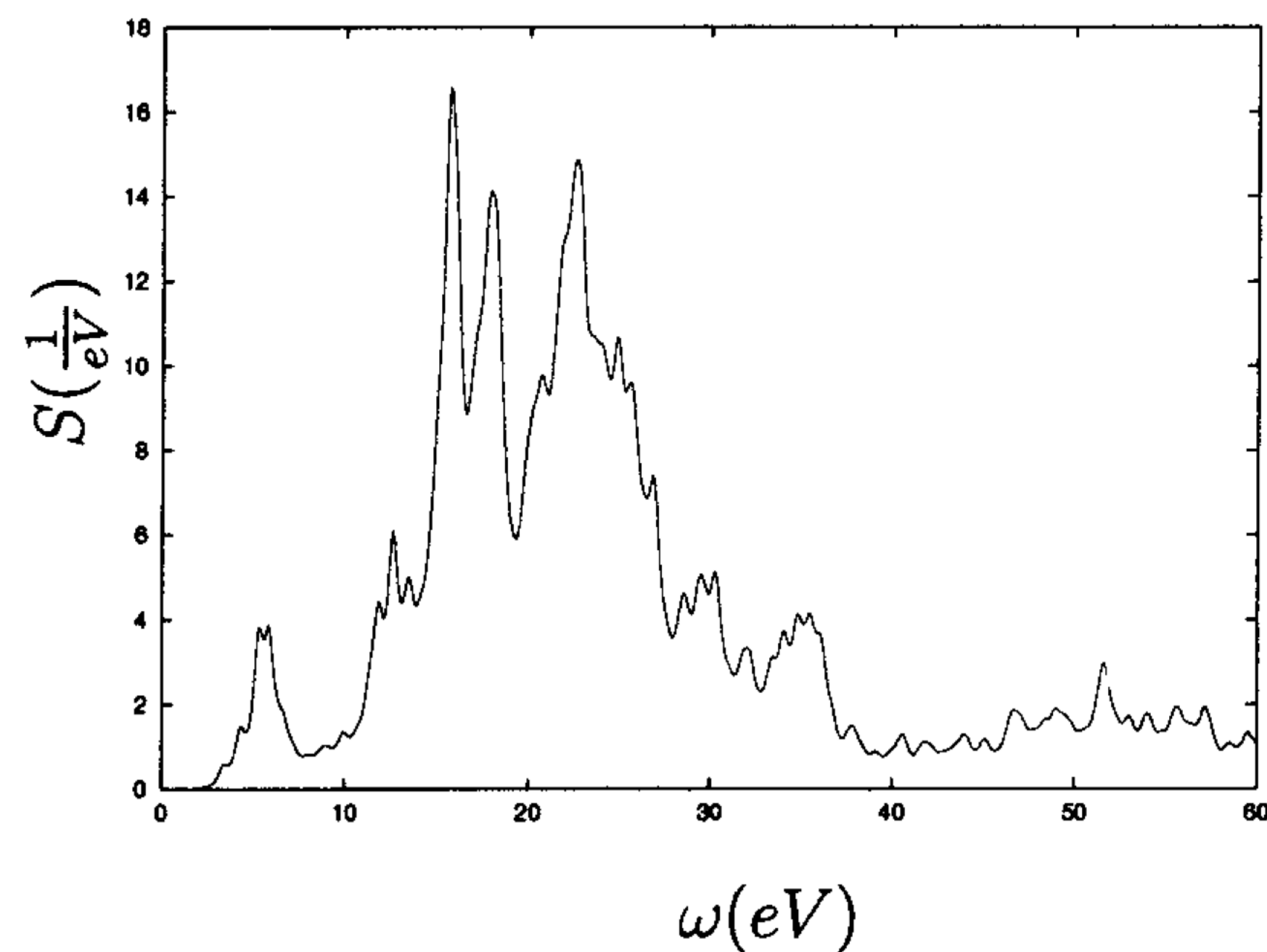
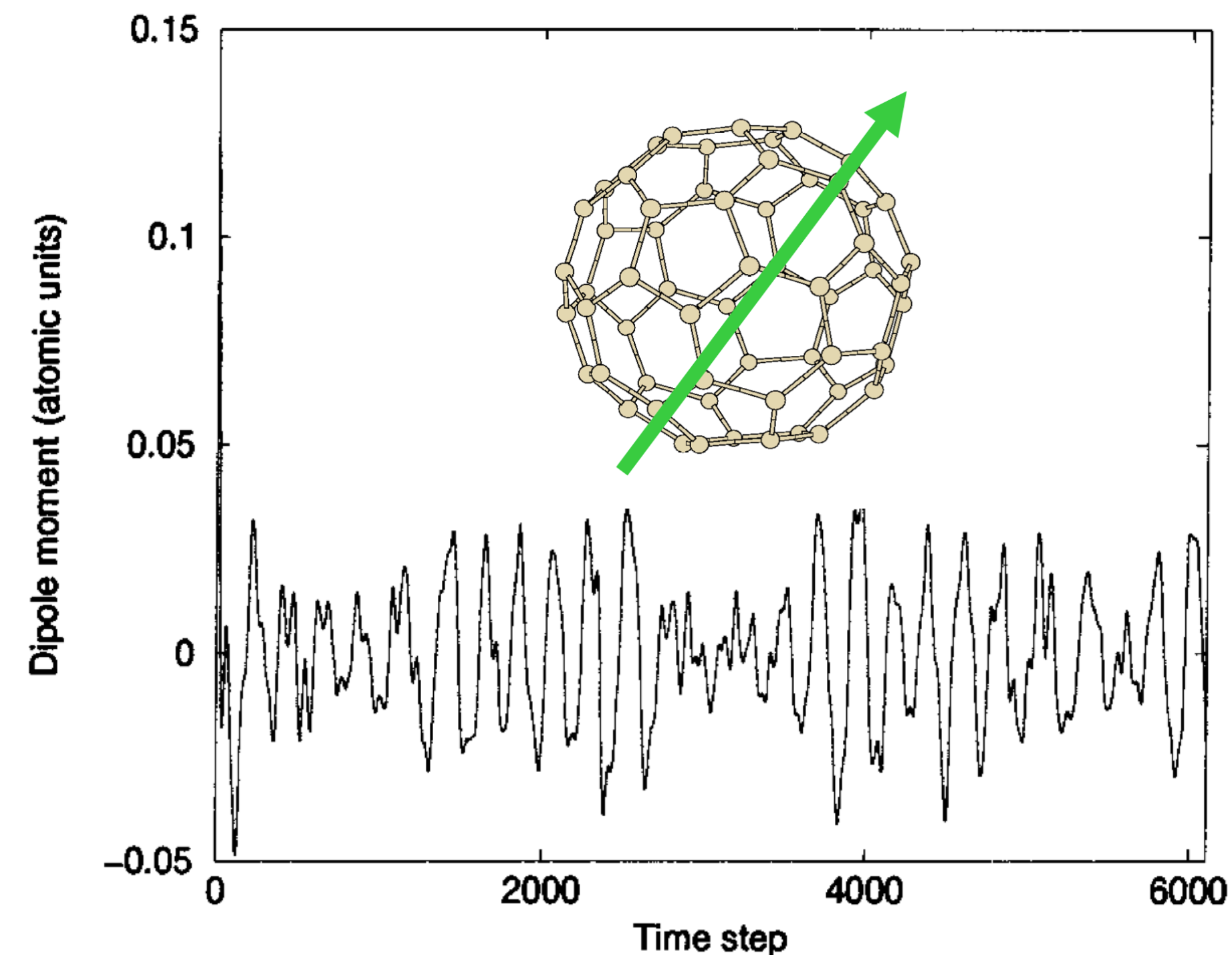
$$p(t) = \int d^3r z \cdot n(r, t) = \int dt' \alpha(t - t') E(t')$$

- Polarizability:

$$\alpha(\omega) = \int dt e^{i\omega t} \alpha(t) \sim \int dt e^{i\omega t} p(t)$$

- Dipole strength function:

$$S(\omega) = \frac{2m}{\pi e^2 \hbar} \omega \operatorname{Im} \alpha(\omega) \sim \sigma(\omega) \quad \text{photoabsorption cross section}$$



Optical properties for molecular systems

Additional notes

- Time step \sim attosecond (10^{-16} fs)

- $\Delta T \sim 2\pi\hbar/\Delta E$ 1 eV  $\Delta T \approx 4$ fs

- Non-linear optical properties also accessible

(... for example work by J.J Rehr & collaborators)

$$p_i(t) = p_i^0 + p_{ij}^{(1)}(t) \cdot E_j + p_{ijk}^{(2)}(t) E_j E_k + p_{ijkl}^{(3)}(t) E_j E_k E_l + \dots$$

$$\alpha(\omega) \longrightarrow \chi_{ij}^{(1)}(\omega) \quad \chi_{ijk}^{(2)}(\omega_1, \omega_2), \chi_{ijkl}^{(3)}(\omega_1, \omega_2, \omega_3), \dots$$

- Solids...? Replace electric potential (non-periodic) by vector potential (not implemented)

$$\vec{E} = -\vec{\nabla}\phi - \frac{\partial \vec{A}}{\partial t} \quad i\hbar \frac{\partial}{\partial t} \psi(t) = \left\{ \frac{1}{2m_e} \left[\vec{p} - \frac{e}{c} \vec{A}(t) \right]^2 - e\phi \right\} \psi(t)$$

Ehrenfest dynamics

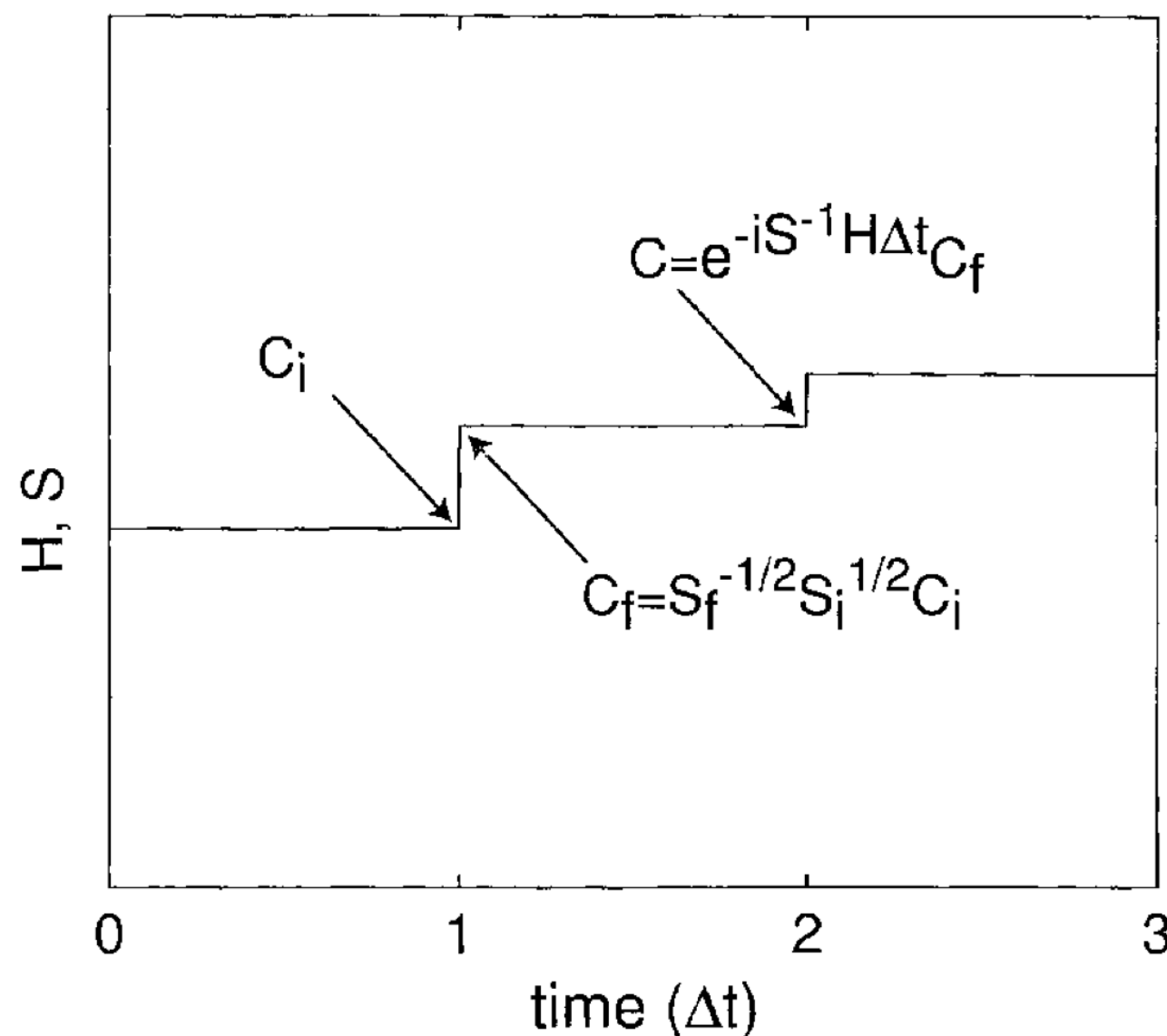
Moving orbitals...

- Moving support basis adds new terms in the equations, similar to Pulay-like corrections

$$\psi_i(r, t) = \sum_{\mu} c_{i,\mu}(t) \phi_{\mu}(r - R_{\mu}(t)) \quad \longrightarrow \quad \frac{\partial \psi_i}{\partial t} = \sum_{\mu} \left[\frac{\partial c_{i,\mu}}{\partial t} \phi_{\mu}(r - R_{\mu}(t)) - c_{i,\mu}(t) \vec{v}_{\mu} \cdot \vec{\nabla} \phi_{\mu} \right]$$

$$i \frac{\partial c_j(t)}{\partial t} = S^{-1} H(t) c_j(t) \quad \longrightarrow \quad i (\partial_t + D) c_j(t) = S^{-1} H(t) c_j(t)$$

$$D_{\mu\nu} = \langle \phi_{\mu} | \partial_t \phi_{\nu} \rangle$$



- When the basis changes, a Löwdin orthonormalisation is performed

...related to the basis set evolution:

- looks very much like a Berry connection
- Christoffel symbol for covariant time-derivative

E. Artacho and D. O'Regan *Phys. Rev. B* **95**, 115155 (2017)

$$c_{new}(t + \Delta t) = S^{-1/2}(t + \Delta t) S^{1/2}(t) c(t + \Delta t)$$

Ehrenfest dynamics

Moving orbitals...

- Moving support basis adds new terms in the equations, similar to Pulay-like corrections
- The Löwdin step is the most expensive operation (and it is not completely consistent with the D matrix connection).
- Relevant for very fast atoms (work in progress):
 - Replacement of Löwdin by other algorithms
 - Corrections to the forces T. N. Todorov, *J. Phys. Condens. Matter* **13**, 10125 (2001)
- Improved propagation schemes (extrapolate the Hamiltonian)

$$c(t + \Delta t) = \frac{S - iH_{ext}\Delta t/2}{S + iH_{ext}\Delta t/2}c(t) \quad H_{ext} = H(t) + \frac{1}{2}\Delta H \quad \Delta H = H(t) - H(t - \Delta t)$$

Improved propagation schemes

- Extrapolate the Hamiltonian

$$c(t + \Delta t) = \frac{S - iH_{ext}\Delta t/2}{S + iH_{ext}\Delta t/2}c(t) \quad H_{ext} = H(t) + \frac{1}{2}\Delta H \quad \Delta H = H(t) - H(t - \Delta t)$$

- Intermediate step propagation

$$1) \quad c(t + \Delta t/2) = [S + iH\Delta t/4]^{-1} [S - iH\Delta t/4] c(t)$$

$$2) \quad c(t + \Delta t/2) \longrightarrow H(t + \Delta t/2)$$

$$3) \quad c(t + \Delta t) = [S + iH(t + \Delta t/2)\Delta t/2]^{-1} [S - iH(t + \Delta t/2)\Delta t/2] c(t)$$

- Divide Δt in n sub-steps

$$c(t + \Delta t) = \prod_{j=1}^n \left[S + iH_{ext}^j \frac{\Delta t}{n} \right]^{-1} \left[S - iH_{ext}^j \frac{\Delta t}{n} \right] c(t) \quad H_{ext}^j = H(t) + \frac{1}{n} \left(j - \frac{1}{2} \right) \Delta H$$

Improved propagation schemes

Total energy conservation

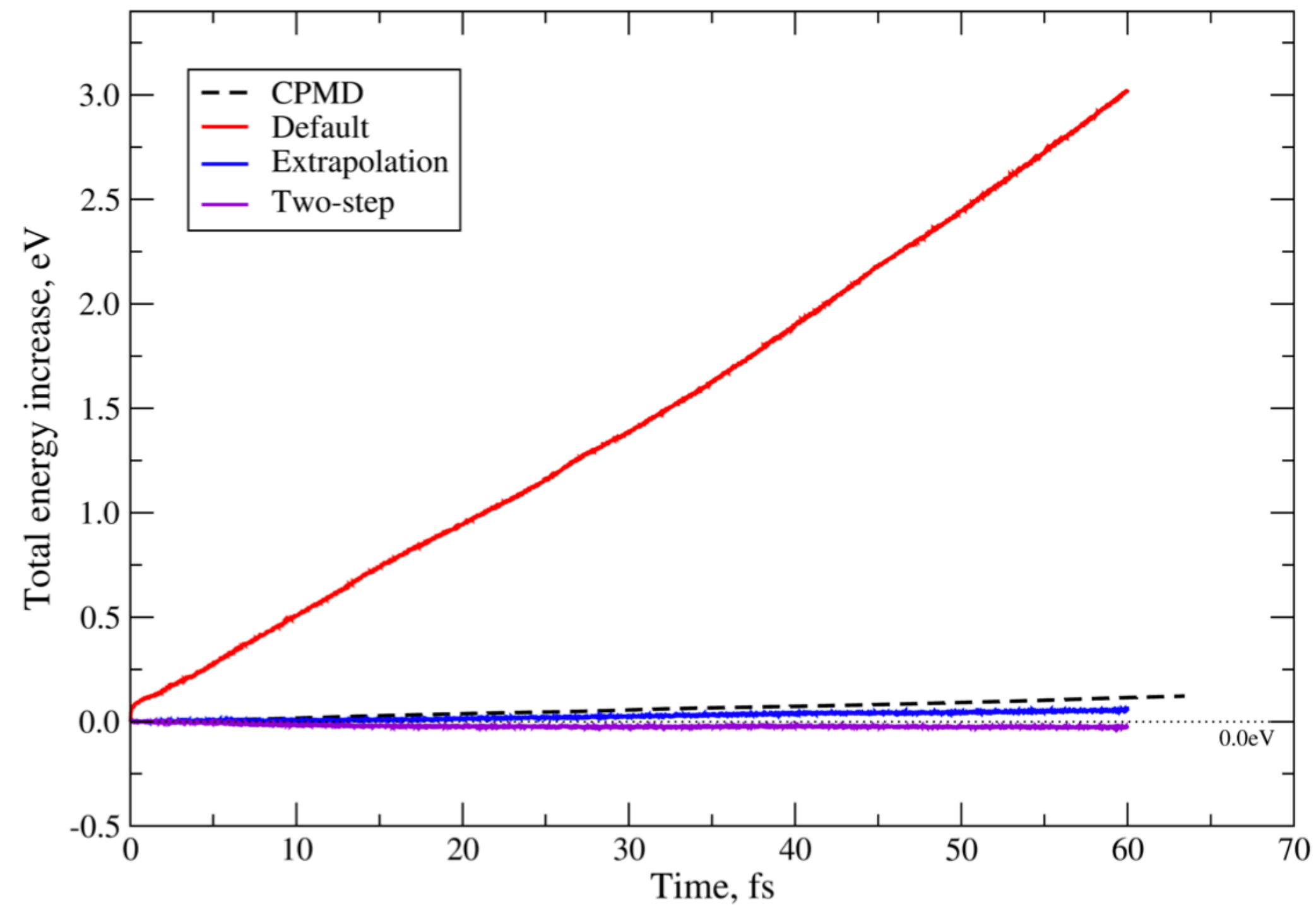


FIG. 9. Energy drift in an energy conserving TDDFT simulation of ionized-core uracil for the three propagation methods described here, as compared with the CPMD implementation, for a time step of $\Delta t = 0.24$ attoseconds (0.01 atomic units).

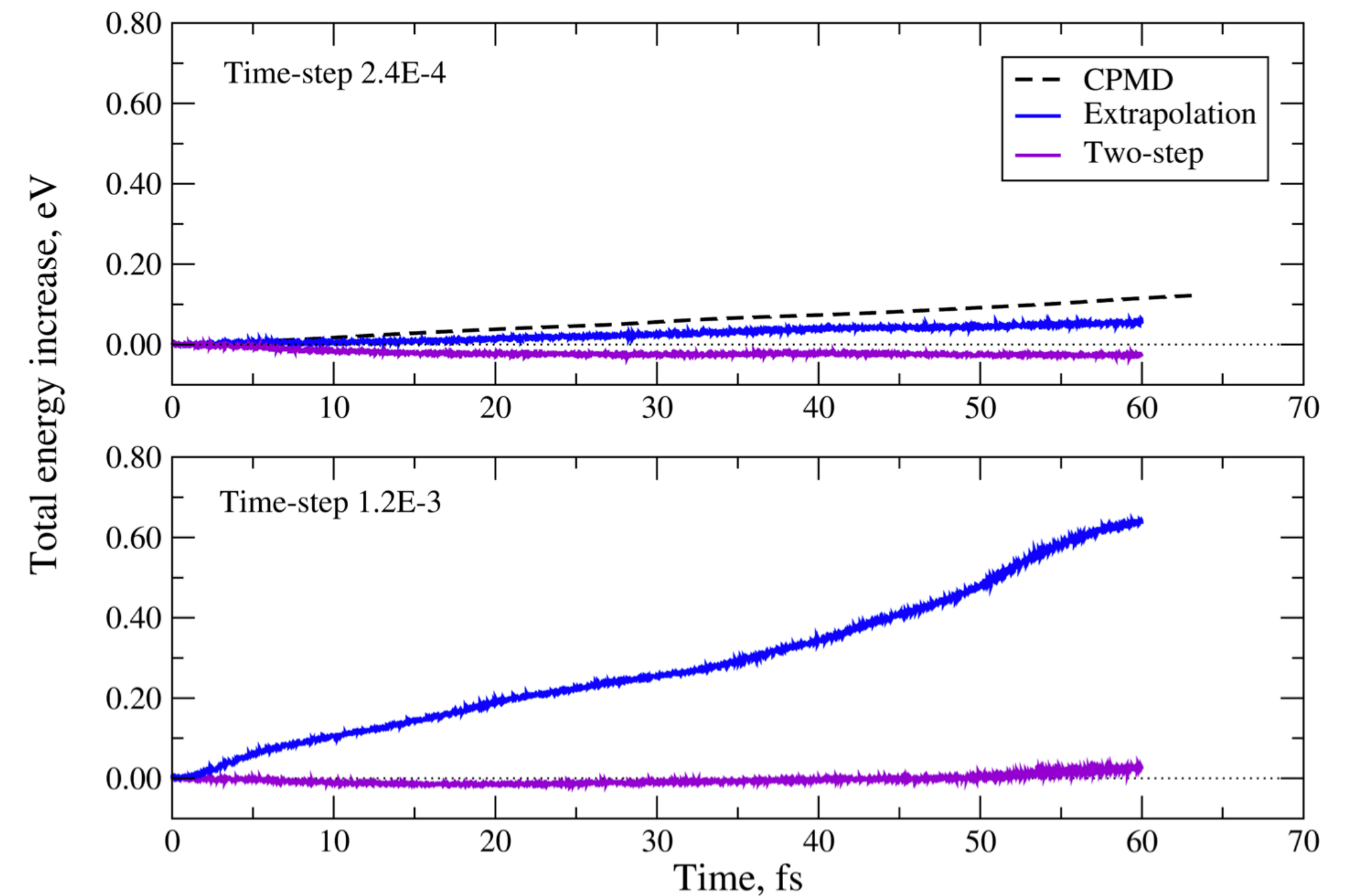


FIG. 10. Energy drift comparison as in Fig. 9, for time steps $\Delta t = 0.24$ and 1.2 attoseconds.

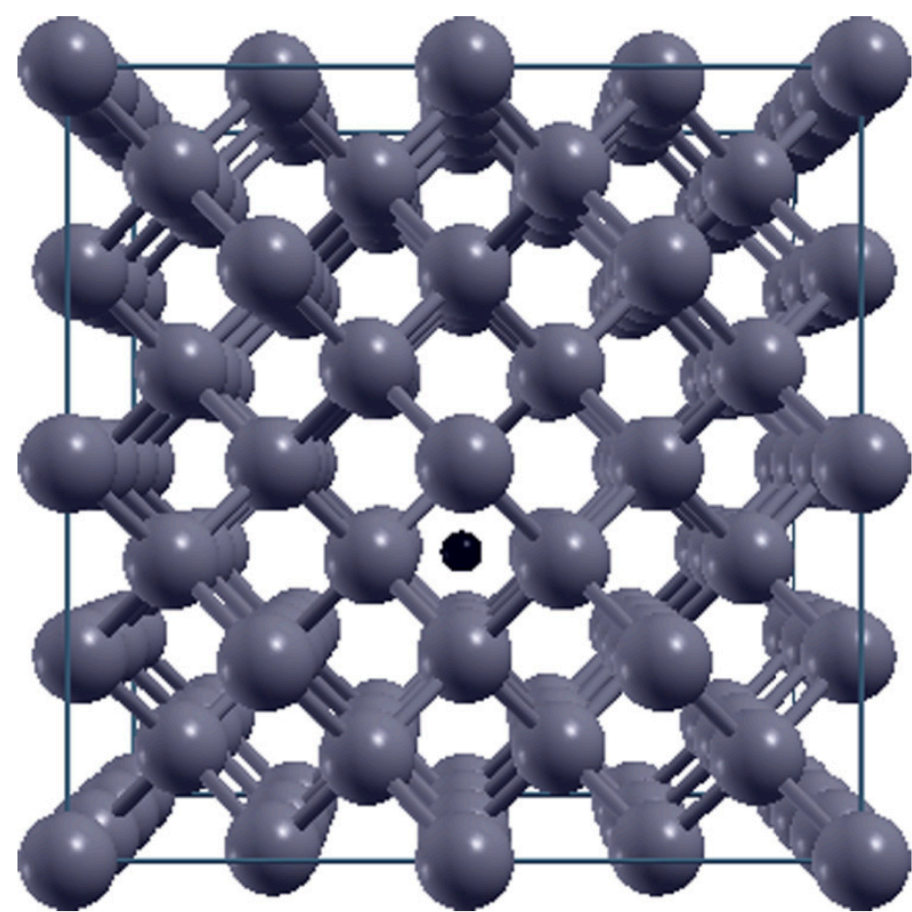
Applications for Radiation Damage

Fast ions moving in a host material

- Electronic stopping power

$$E(t) = \sum_i \int d\mathbf{r} \psi_i^*(\mathbf{r}, t) \left\{ \frac{\hbar^2 \nabla^2}{2m_e} + V_{\text{ext}}(\mathbf{r}) \right\} \psi_i(\mathbf{r}, t) + E_{\text{HXC}}[n] \\ + V_{\text{ion-ion}}[\{\mathbf{R}_J\}_J] + \sum_J \frac{1}{2} M_J \ddot{\mathbf{R}}_J^2$$

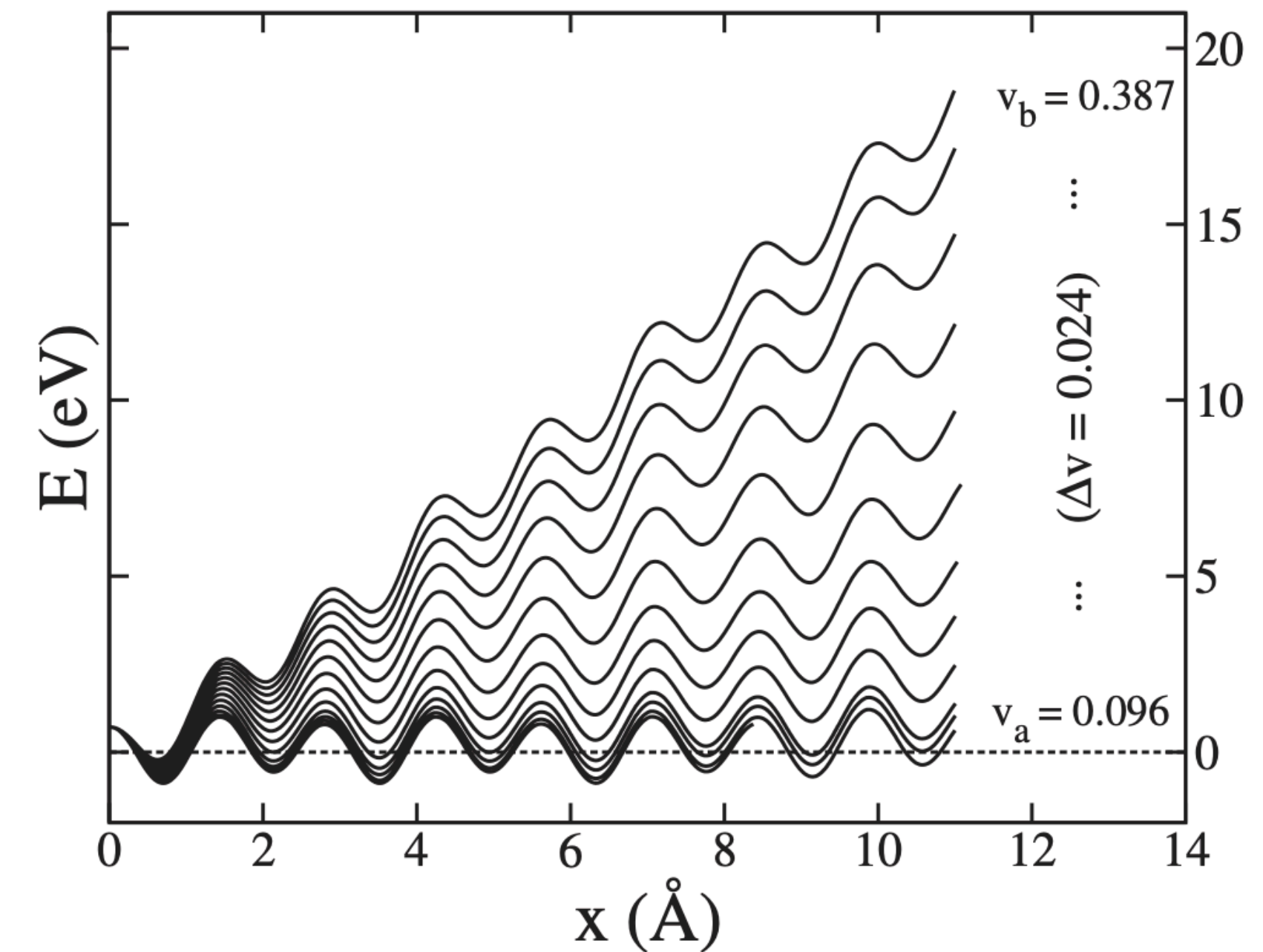
- Insulators, semiconductors, and metals



- ▶ Threshold velocities
- ▶ Barkas effect
- ▶ Periodic boundary conditions
- ▶ Transients

Pruneda et al, *Phys. Rev. Lett.* **99**, 235501 (2007)

Total *electronic* energy as a function of displacement for different velocities



Check other works from Emilio Artacho & collaborators

FIG. 2. (Color online) Ge supercell in the [001] direction with H in a channel.

How to use it?

Variables to be defined in the input file

MD.TypeOfRun **TDED**

TDED.WF.Initialize `false` *(logical)*

If set to `true` in a standard self-consistent SIESTA calculation, it makes the program save the KSOs after reaching self-consistency. This constitutes the first run.

TDED.Nsteps `1` *(integer)*

Number of electronic time steps between each atomic movement. It can not be less than 1.

TDED.TimeStep `0.001 fs` *(time)*

Length of time for each electronic step. The default value is only suggestive. Users must determine an appropriate value for the electronic time step.

How to use it?

Variables to be defined in the input file

`MD.TypeOfRun` `TDED`

- A first run is needed to “initialise”. For optical response, you have to define an electric field in the first run, and then switch it off.
- After the initialisation, SIESTA uses `SystemLabel.TDWF` and `SystemLabel.TDXV`
- In each SCF step, the occupied electronic states are evolved in time.
- If you want electron dynamics-only, set `MD.FinalTimeStep` to 1
- `MD.LengthTimeStep` is ignored (SIESTA uses instead `TDED.TimeStep` multiplied by `TDED.Nsteps`).

How to use it?

Variables for the solver

TDED.Extrapolate false *(logical)*

An extrapolated Hamiltonian is applied to evolve KSOs for **TDED.Extrapolate.Substeps** number of substeps within a single electronic step without re-evaluating the Hamiltonian.

TDED.Extrapolate.Substeps 3 *(integer)*

Number of electronic substeps when an extrapolated Hamiltonian is applied to propagate the KSOs. Effective only when **TDED.Extrapolate** set to be true.

TDED.Inverse.Linear true *(logical)*

If **true** the inverse of matrix

$$\mathbf{S} + i\mathbf{H}(t)\frac{dt}{2} \tag{19}$$

is calculated by solving a system of linear equations which implicitly multiplies the inverted matrix to the right hand side matrix. The alternative is explicit inversion and multiplication. The two options may differ in performance.

How to use it?

I/O information

TDED.Write.Etot true *(logical)*

If **true** the total energy for every time step is stored in the file `SystemLabel.TDETOT`.

TDED.Write.Dipole false *(logical)*

If **true** a file `SystemLabel.TDDIPOL` is created that can be further processed to calculate polarizability.

TDED.Write.Eig false *(logical)*

If **true** the quantities $\langle \phi(t) | H(t) | \phi(t) \rangle$ in every time step are calculated and stored in the file `SystemLabel.TDEIG`. This is not trivial, hence can increase computational time.

How to use it?

I/O information

TDED.WF.Save `false` *(logical)*

Option to save wavefunctions at the end of a simulation for a possible restart or analysis. Wavefunctions are saved in file `SystemLabel.TDWF`. A TDED restart requires `SystemLabel.TDWF`, `SystemLabel.TDXV`, and `SystemLabel.VERLET_RESTART` from the previous run. The first step of the restart is same as the last of the previous run.

TDED.Saverho `false` *(logical)*

If **true** the instantaneous time-dependent density is saved to `<istep>.TDRho` after every **TDED.Nsaverho** number of steps.

TDED.Nsaverho `100` *(integer)*

Fixes the number of steps of ion-electron dynamics after which the instantaneous time-dependent density is saved. May require a lot of disk space.