Supplemental Material for: *Ab initio* electron dynamics in high electric fields: accurate predictions of velocity-field curves

Ivan Maliyov,[1] Jinsoo Park,[1] and Marco Bernardi[1,*]

[1] *Department of Applied Physics and Materials Science, California Institute of Technology, Pasadena, California 91125*

RT-BTE SOLVERS

We provide details for the rt-BTE solvers discussed in the manuscript and implemented in PERTURBO [1]. The rt-BTE reads:

$$\frac{\partial f_{n\mathbf{k}}(t)}{\partial t} = \frac{e\mathbf{E}}{\hbar} \cdot \nabla_{\mathbf{k}} f_{n\mathbf{k}}(t) + \mathcal{I}^{\mathrm{e-ph}}[f_{n\mathbf{k}}(t)]. \tag{1}$$

In the following, for the electron occupations $f_{n\mathbf{k}}(t)$, we will omit the band and momentum indices, n and $\mathbf{k}$.

For explicit and predictor-corrector solvers (groups (i) and (ii) in the main text), we consider the right-hand side of Eq. (1) as one term, without splitting the advection and e-ph collision parts:

$$\frac{\partial f(t)}{\partial t} = F\big[f(t)\big]. \tag{2}$$

I. EXPLICIT SOLVERS

In explicit methods, one finds the function at the next time step, $t+\Delta t$, using only the value of the function at the current time, t. The simplest and most commonly used explicit solver is forward Euler (**FE**), which requires only one evaluation of $F\big[f(t)\big]$:

$$f(t+\Delta t) = f(t) + F\big[f(t)\big]\Delta t. \tag{3}$$

Another widely used explicit solver is the fourth-order Runge-Kutta (**RK4**) method, which is more accurate, but also more computationally expensive since it requires four evaluations of $F\big[f(t)\big]$ at each step:

$$f(t+\Delta t) = f(t) + \frac{1}{6}\Delta t\big(k_1 + 2k_2 + 2k_3 + k_4\big), \tag{4}$$

$$k_1 = F\big[f(t)\big], \tag{5}$$

$$k_2 = F\big[f(t) + k_1\Delta t/2\big], \tag{6}$$

$$k_3 = F\big[f(t) + k_2\Delta t/2\big], \tag{7}$$

$$k_4 = F\big[f(t) + k_3\Delta t\big]. \tag{8}$$

II. PREDICTOR-CORRECTOR (PC) SOLVERS

In PC solvers, one first finds the function at $t+\Delta t$ with an explicit solver, and then refines the result with an implicit solver, which uses both the function at the current time t and at the next time step $t+\Delta t$. For the predictor step, we use the explicit solvers (FE or RK4) described in the previous section. For the corrector step, we use backward Euler (**BE**):

$$f(t+\Delta t) = f(t) + F\big[f(t+\Delta t)\big]\Delta t \tag{9}$$

or Crank-Nicolson (**CN**) [2]:

$$f(t+\Delta t) = f(t) + \frac{1}{2}\Big(F\big[f(t)\big] + F\big[f(t+\Delta t)\big]\Big)\Delta t. \tag{10}$$

For both the BE and CN solvers, the $f(t+\Delta t)$ value is obtained from the first predictor step.

III. STRANG SPLITTING SOLVERS

For splitting methods, we consider separately the advection and collision terms in Eq. (1), and denote them respectively as $F_{\rm adv}\big[f(t)\big]$ and $F_{\rm coll}\big[f(t)\big]$ as in the main text. The rt-BTE is written as:

$$\frac{\partial f_{n\mathbf{k}}(t)}{\partial t} = F_{\rm adv}\big[f(t)\big] + F_{\rm coll}\big[f(t)\big]. \tag{11}$$

The Strang 2$^{\rm nd}$ order splitting algorithm obtains the function at the next time step, $f(t+\Delta t)$, from the function at the current time, $f(t)$, advancing the equation to the next time step using a sequence of three operations [3]. For the collision-advection-collision (CAC) sequence described in the main text, this sequence of operations is:

1. $\dfrac{\partial f^{(1)}(t)}{\partial t} = F_{\rm coll}\big[f^{(1)}(t)\big]$,
   find $f^{(1)}(t+\Delta t/2)$ from $f^{(1)}(t) = f(t)$

2. $\dfrac{\partial f^{(2)}(t)}{\partial t} = F_{\rm adv}\big[f^{(2)}(t)\big]$,
   find $f^{(2)}(t+\Delta t)$ from $f^{(2)}(t) = f^{(1)}(t+\Delta t/2)$

3. $\dfrac{\partial f^{(3)}(t)}{\partial t} = F_{\rm coll}\big[f^{(3)}(t)\big]$,
   find $f^{(3)}(t+\Delta t)$ from $f^{(3)}(t+\Delta t/2) = f^{(2)}(t+\Delta t)$

4. $f(t+\Delta t) = f^{(3)}(t+\Delta t)$.

For steps 1,2, and 3, one can use any type of explicit or PC solver described above. If the advection part is evaluated first, steps 1 and 3 use $F_{\rm adv}\big[f(t)\big]$, and step 2 $F_{\rm coll}\big[f(t)\big]$. In this case, the resulting solver is called ACA.

* bmarco@caltech.edu

[1] J.-J. Zhou, J. Park, I.-T. Lu, I. Maliyov, X. Tong, and M. Bernardi, Comput. Phys. Commun. **264**, 107970 (2021).
[2] J. W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods* (Springer Science & Business Media, 2013).
[3] G. Strang, SIAM J. Numer. Anal. **5**, 506 (1968).