Manual for TCSFA Package

by Timy

1 Form of the Light Field

Dipole approximation is employed. The light field is defined by its vector potential with the $\sin^2$-envolope:

$$\boldsymbol{A}\cdot\hat{\boldsymbol{z}} = -\frac{E_0}{\omega\sqrt{1+\xi^2}}\sin^2\left(\frac{\omega t}{2N_c}\right)\sin(\omega t+\varphi)$$

$$\boldsymbol{A}\cdot\hat{\boldsymbol{x}} = -\frac{E_0\,\xi}{\omega\sqrt{1+\xi^2}}\sin^2\left(\frac{\omega t-\pi/2}{2N_c}\right)\cos(\omega t+\varphi),$$

where E_0 is the electric field amplitude, ω is carrier frequency, N_c is the number of optical cycles, φ is the phase, ξ is the ellipticity. When $\xi=0$, it describes the linearly polarized field with the polarization direction along the z-axis.

1 Organization of the TCSFA Package

- `src`
  - `ccsfa`: The main directory for the source of the TCSFA package
    - `core`: the core part of the TCSFA
    - `include`: stores head files generated by the Python script “config.py”.
    - `p_entry`: entry point of the parallel version.
    - `s_entry`: entry point of the standalone version.
      - `main.f90`: is the entry point of standalone computation. One can modify this file to calculate a single trajectory or multiple trajectories and related quantities, or whatever.
      - `data_proc.f90`: is used to reproduce a batch of trajectories selected from the raw data with certain conditional filter. Given a subset of the raw data `select.dat` and filter, it generates the index of the data satisfying the critera `filter.dat` and the corresponding transition amplitude `traj_m.dat`.
    - `config.py`: translates parameter list into individual head file for each module.
  - `mmff`: MPI Framework for Fortran (⟨hlink|https://github.com/timy/MMFF|MMFF⟩)
- ana
  - `proc`: post-process programs for analysis of the generated raw data
  - `plot`: some easy-to-use python scripts for quick access to visualization of results
  - `data`: data results for visualization

W_{sub} without Coulomb correction can be integrated numerically with the function `action_W_im_num` with the integrand `v2_integrand` from t_s to t_0; or it can be obtained with the analytical expression with the function `action_W_im`. W_{mix} (S-representation) or $W_{\text{mix}}+W_{\text{kep}}$ (W-representation) can be directly obtained as an argument from function `rk4_re` by numerical integration.