XTANT manual

XTANT: A simulation package for modelling **X**-ray-induced **T**hermal **A**nd **N**onthermal **T**ransitions

Current version: XTANT-3 Last modification: 20/07/2022

Nikita Medvedev[1]

Table of Contents

[1] nikita.medvedev@fzu.cz

Hamburg, CFEL at DESY, 2011-2016
Prague, Czech Academy of Sciences, 2016-2022

I. Disclaimer, how to cite

Although we endeavour to ensure that the code XTANT and results delivered are correct, no warranty is given as to its accuracy. We assume no responsibility for possible errors or omissions. We shall not be liable for any damage arising from the use of this code or its parts or any results produced with it, or from any action or decision taken as a result of using this code or any related material.

This code is distributed *as is* for non-commercial peaceful purposes only, such as research and education. It is explicitly **prohibited** to use the code, its parts, its results or any related material for military-related and other than peaceful purposes.

By using this code or its materials, you agree with these terms and conditions.

The use of the code is at your own risk. Should you chose to use it, an appropriate citation is mandatory:

N. Medvedev, V. Tkachenko, V. Lipp, Z. Li, B. Ziaja, Various damage mechanisms in carbon and silicon materials under femtosecond x-ray irradiation, 4open. 1 (2018) 3. https://doi.org/10.1051/fopen/2018003

Should you use electron-phonon coupling in the calculations, the following citation should be included in addition to the abovementioned one:

N. Medvedev, I. Milov, Electron-phonon coupling in metals at high electronic temperatures, Phys. Rev. B. 102 (2020) 064302. https://doi.org/10.1103/PhysRevB.102.064302

In a publication, we recommend that at least the following parameters should be mentioned for reproducibility of the results:

Material, its initial structure, the number of atoms in the supercell, the initial conditions (atomic and electronic temperatures), an ensemble used, a type of boundary conditions, a type of cross sections in Monte Carlo simulation, a type of tight binding parameterization, whether the electron emission was included or not and if yes, whether Coulomb potential for atoms was accounted for and what model for electron emission was used, whether an additional short-range repulsive potential was used, time step of MD simulation, parameters of the incoming laser pulse (its photon energy, deposited dose, duration).

Most of these parameters can be found in an output file !OUTPUT_[*Material*]_Parameters.txt described below.

II. Brief description of the model

The hybrid code uses developed scheme combining a few different models in one:

1) Transferrable tight binding (TB) method for obtaining the Hamiltonian of system of atoms
2) Molecular dynamics (MD) for tracing atomic motion on the potential energy surface, calculated with the tight binding method
3) Monte Carlo (MC) method for tracing high-energy electrons and core holes, excited by an incoming FEL pulse
4) Boltzmann collision integral describe nonadiabatic coupling of low-energy electrons (valence and bottom of the conduction band) to the ions (generalized electron-phonon coupling)

5) RPA model for calculation of the complex dielectric function and optical properties

Details on the physical background of the model and the numerics developed can be found in [1]. Various parts of the model are contained in different modules of the code, which are largely independent.

III. Limits of applicability of XTANT

The code has the following limitations due to approximations of the model:

- The photon energy is limited to ~30 eV (no nonlinear photoabsorption included, which becomes increasingly important at lower photon energies) up to ~20 keV (no radiative decays of core holes are included, which becomes important for heavy elements at higher photon energies; no relativistic effects for electrons are included).
- The deposited dose is limited to ~5 eV/atom, depending on the TB parameterization of the material. The code MAY be used at higher doses, if you understand the risks and shortcomings – such as, e.g., the results at ultrashort timescales may still be reliable, or ultrafast cooling or electron emission that may quickly reduce the dose, etc.
- At low deposited doses, the electron-phonon coupling model is unreliable, it is limited to electronic temperatures above some ~2-3 kK.
- The number of atoms in the simulation box must be larger than some 200-300, defined by the two factors: 1) the TB cut-off radius must be smaller than half of the simulation box with periodic boundaries; 2) the electron-phonon coupling, calculated at the gamma-point only, requires a few hundred atoms for convergence (must be checked for each material).
- Duration of simulation is typically limited to a few picoseconds, due to possible accumulating MD instabilities – they must be checked by the energy conservation in the simulation. Poor convergence (non-conserved energy) should be corrected by reducing MD time step.

IV. Compiling XTANT

If executable (e.g. XTANT.x, or XTANT.exe) does not exist, compile the source files. With help of the makefile, the compilation under Linux is as follows: go into the directory with the whole code (all necessary files that must be present are described below), and execute the command

make

in a terminal. It will compile the code and create an executable named XTANT.x in the same folder. It might take a few moments. It automatically compiles the code with OpenMP[2].

Then execute the corresponding shell-script, which specifies paths to MKL libraries on your computer/cluster. Examples of the shell scripts prepared for running XTANT.x on some clusters it was used before include:

./XTANT_DESY.sh - to run XTANT at the cluster in DESY (Hamburg)

[2] More on OpenMP with fortran: http://www.openmp.org/presentations/miguel/F95_OpenMPv1_v2.pdf

./XTANT_Metacentrum.sh – to run XTANT on Luna cluster as part of the Metacentrum (Prague)

Alternatively, if you are using Windows (with Intel Fortran compiler, OpenMP and MKL-library installed, and paths set in the environmental variables), rename file Make.bat.txt into Make.bat, and execute it. It will create an executable XTANT.exe.

Run this XTANT.exe in case you are using Windows.

In case if the program cannot find libraries required for OpenMP parallelisation, you might have to specify the paths to them manually by executing commands similar to these (insert your current paths accordingly):

```
export LD_LIBRARY_PATH=/opt/intel/2011/lib/intel64:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/opt/products/mkl/11.0/mkl/lib/em64t:$LD_LIBRARY_PATH
```

Some other libraries might be missing on your workstation. In this case, find the paths to them and export them analogously.

In case of having problems with a lack of memory (which XTANT usually does), you must set it unlimited by executing

```
ulimit -s unlimited
limit stacksize unlimited
```

(For Windows, use the compiler option /F9999999999)

These four commands usually are not necessary since they are already inside of the XTANT.sh

If you wish to recompile the code without OpenMP, first you have to clean up all the old compiled files with the command

make clean

After that you can compile the code without OpenMP by specifying

make OMP=no

Again, new recompilation with OpenMP must be preceded by making clean, because there are pre-processing options included. So, each recompilation which changes the involvement of OpenMP must be done *only after cleaning up* the files. This option is mainly for code-developing, since executing is much faster with parallelization with OpenMP.

Another option for debugging during developing of the code is compiling with:

make DEBUG=yes

Under Windows a user may compile it in the following way: Make.bat X or make X, where X is one of the following options (no '*make clean*' needed for recompiling):

DEBUG - for debugging during developing of the code (such as checking arrays boundaries, undeclared variables, etc.), compiles XTANT_DEBUG.exe.

DEBUGOMP or db - to compile parallel version with all debug options on, compiles XTANT_DEBUG_OMP.exe.

FAST or slow – to compile parallelized code with no optimizations and no debug (fast compilation, not-so-the-fastest execution), compiles XTANT_OMP.exe.

For a compilation for release, simply use Make.bat or make, with no additional option, which compiles XTANT.exe.

V. Compiling additional post-processing programs

XTANT package has additional useful programs for data analysis, saved in the folder !XTANT_ANALYSIS_SUBROUTINES. A line for their compilation is written in the top part of each file, and provided here below:

1) A program for calculation of atomic pair correlation function from atomic coordinates:
   XTANT_atomic_data_analysis.f90

This program can be compiled as follows (example for Windows):

```
ifort.exe /F9999999999 /O3 /Qipo /Qvec-report1 /fpp /Qopenmp /heap-arrays XTANT_atomic_data_analysis.f90 -o XTANT_atomic_data_analysis.exe /link /stack:9999999999
```

2) A program for calculation of atomic velocity autocorrelation functions and phonon spectra:
   XTANT_autocorrelators.f90
   This program can be compiled as follows (example for Windows):

```
ifort.exe /F9999999999 /O3 /Qipo /Qvec-report1 /fpp /Qopenmp /heap-arrays
XTANT_autocorrelators.f90 -o XTANT_autocorrelators.exe /link /stack:9999999999
```

3) A program for extracting electron-ion coupling parameter and electron heat capacity and chemical potential as a function of electron temperature from a set of simulations runs:
   XTANT_coupling_parameter.f90

To compile (example for Windows):

```
ifort.exe /F9999999999 /O3 /Qipo /Qvec-report1 /fpp /Qopenmp /heap-arrays XTANT_coupling_parameter.f90 -o XTANT_coupling_parameter.exe /link /stack:9999999999
```

4) A program for extracting optical parameters for a given photon wavelength from the data on spectrum of the optical parameters:
   XTANT_dielectric_function_analysis.f90

To compile (example for Windows):

```
ifort.exe /F9999999999 /O3 /Qipo /Qvec-report1 /fpp /Qopenmp /heap-arrays XTANT_dielectric_function_analysis.f90 -o XTANT_dielectric_function_analysis.exe /link /stack:9999999999
```

VI. Running XTANT with additional options

To include additional options in the code, you can run it with some additional options as follows:
./XTANT.sh X (or XTANT.x X, or XTANT.exe X)

where X is an option available. At the moment, there are only a few options:

1) help -- If you need some short info on the XTANT, when starting the code, you can also call the following help-commands: help (so, run it as ./XTANT.sh help, or correspondingly ./XTANT.x help or XTANT.exe help). It will printout the numbers and meaning of possible errors in the error-file (see below), how to communicate with the program on-the-fly (see below), etc. This flag will only print some info and stop the execution of the code; no calculations will be performed.
2) info -- Which before running the code will print out some basic information about the XTANT. Running with this option will tell you some information about references, disclaimer, how to cite the code and similar info. This flag will only print some info and stop the execution of the code; no calculations will be performed.
3) size -- If you want to create file with cohesive energy of the material as a function of the nearest neighbour distance (to compare with other works), run XTANT with this option will create an output file named OUTPUT_Energy.dat with the energy as a function of the nearest neighbour distance, see a description below. Note: this file is overwritten every time you run the program with the flag "size".
4) allow_rotation -- By default, initializing MD removes total angular momentum of the system. If you want to start MD simulation without removing it, which will allow the whole system to rotate (e.g. might be useful for modelling individual molecules), use this flag.
5) verbose -- If you want to see a lot of information on the screen during the simulation run (such as, when which subroutine was called, etc.), which may be useful for debugging and testing, use this option.
6) For regular simulation run without additional options simply call:

./XTANT.sh or XTANT.exe for Windows.

Note that XTANT supports as many flags at a simulation run as you need, they can be combined.

VII. Files of the code

This code XTANT is contained in a number of files that have to be compiled together, plus additional files for post-processing of the data if needed. All the files listed below must be in the same directory for XTANT code compilation and execution:

- Makefile – this file uses standard linux program make[3] to create links between the compiler and source files, compile all the modules, and the final program: XTANT.x. Use it for Linux-based systems. For Windows, alternatively rename Make.bat.txt to Make.bat and execute the file Make.bat in the command line (assuming all the paths to needed libraries such as OpenMP are provided in your system, or the libraries themselves are in the same folder).

[3] Details on what is 'make': http://linux.about.com/library/cmd/blcmdl1_make.htm

By default, it uses Intel fortran-2013 compiler (ifort2013; or ifort.exe for Windows); this can be changed in the Makefile. Note that the code uses some of the intel-features, that would need to be corrected, should you want to compile the code with gnu-fortran (gfortran).

- XTANT.sh – this is the shell script to be executed for running the XTANT (executable XTANT.x) under Linux. Alternatively, under Windows run the created XTANT.exe.
- XTANT_MAIN_FILE.f90 – this file contains the main part of the XTANT code. It assembles the program into one peace, performs the dynamics of atoms and electrons by calling all necessary subroutines. Also, all initialization of variables, reading input files, creating output files are called from here.
- Algebra_tools.f90 – this file contains linear algebra necessary subroutines (also with references to LAPACK library[4]).
- Atomic_tools.f90 – this file contains subroutines used for the atomic subsystem.
- BS_Basis_sets.f90 – this file contains subroutines to deal with Gaussian basis sets that are required for xTB calculations (*unfinished*)
- BS_Cartesian_Gaussians.f90 – this file contains subroutines to deal with Cartesian Gaussian basis sets that are required for xTB calculations (*unfinished*)
- BS_Spherical_Gaussians.f90– this file contains subroutines to deal with Spherical Gaussian basis sets that are required for xTB calculations (*unfinished*)
- Coulomb.f90 – this file contains Coulomb potential and forces for modeling Coulomb explosion of finite-size systems.
- Dealing_with_3TB.f90 – this module contains subroutines for reading and interpreting files in 3TB format [2]. Note that currently XTANT only supports 2-body part of the parameterization, the 3-body part is unfinished.
- Dealing_with_BOP.f90 – this module contains subroutines for reading and interpreting files in BOP format [3]. Note that this parameterization only supports dimer molecules, not solids.
- Dealing_with_DFTB.f90 – this module contains subroutines for reading and interpreting files in the Slater-Koster format[5], as provided by DFTB [4].
- Dealing_with_EADL.f90 – this file contains subroutine to read from the EPICS2017 (former EADL and EPDL) databases[6], needed for naming atomic shells, extracting information on Auger-decay rates, ionization potential, photoabsorption cross sections (used in one of the options of MC, see below).
- Dealing_with_files.f90 – this file contains useful subroutines to deal with files, such as counting lines and columns, reading, checking for errors etc.
- Dealing_with_output_files.f90 – this file contains all subroutines to create and prepare output directories and files, to communicate with the program and interpret user's commands.

[4] http://www.netlib.org/lapack/
[5] http://www.dftb.org/parameters/introduction/
[6] See details of this database here: https://www-nds.iaea.org/epdl97/libsall.htm, physical details and references for the database are here: https://www-nds.iaea.org/epdl97/document/epdl97.pdf

- Dealing_with_xTB.f90 – this module contains subroutines for reading and interpreting files in the extended tight binding, xTB, format [5] (*unfinished*)
- Electron_tools.f90 – this file contains subroutines to deal with Fermi-function of low-energy electrons.
- Exponential_wall.f90 – this module contains short-range exponential repulsive potential and forces needed in case TB parameterization provides too low barrier for atoms at short distances.
- Gnuplotting.f90 – this module contains subroutines to create gnuplot shell scripts.
- Initial_configuration.f90 – this file sets up the initial conditions, such as constructing initial atomic positions and velocities and so on.
- Little_subroutines.f90 – this file contains useful subroutines, such as for approximations, search in arrays, resizing arrays, etc.
- MC_cross_sections.f90 – this file contains subroutines for calculation of electron cross sections and mean free path used in the Monte Carlo part. It uses complex dielectric function formalism [6,7], or BEB cross sections [8,9].
- Monte_Carlo.f90 – this file contains all Monte Carlo model subroutines for photons, high-energy electrons, and core holes Auger decays.
- Nonadiabatic.f90 – this file contains subroutines for Boltzmann collision integrals and nonadiabatic electron-ion energy exchange [10,11].
- Objects.f90 – this file contains all the introduced objects in the framework of the object-oriented programming[7], and some subroutines to deal with these objects.
- Optical_parameters.f90 – this file contains subroutines for calculation of the optical part of the complex dielectric function within tight binding and RPA [12], or within Drude model [13].
- Periodic_table.f90 – this file contains subroutines to extract information about each elements from the periodic table (must be attached as one of the input files, see below).
- Read_input_data.f90 – this file contains subroutines to read all necessary input files (see below).
- TB.f90 – this file contains general subroutines to deal with tight binding (TB) formalism. *Eventually, calling a DFT package, or HF, can be placed here too.*
- TB_3TB.f90 – contains subroutines to calculate TB Hamiltonian within one of the following basis sets: s, sp^3, sp^3d^5, and corresponding forces, according to 3TB model [2].
- TB_BOP.f90 – contains subroutines to calculate TB Hamiltonian within one of the following basis sets: s, sp^3, sp^3d^5, and corresponding forces, according to BOP method [3]. *Note that this parameterization only supports dimer molecules, not solids (unfinished).*
- TB_DFTB.f90 – contains subroutines to calculate TB Hamiltonian and repulsive term within one of the following basis sets: s, sp^3, sp^3d^5, and corresponding forces, according to DFTB method [4].
- TB_Fu.f90 – contains subroutines to calculate TB Hamiltonian within the sp^3-basis set and repulsive energy, and corresponding forces, as a combination of Pettifor's parameters, according

[7] Quick introduction into object oriented programming in FORTRAN: http://www.pgroup.com/lit/articles/insider/v3n1a3.htm and http://www.pgroup.com/lit/articles/insider/v3n2a2.htm

to Fu *et al.*[14]. Note that the tests showed unstable systems, even though with correct band structure; *not recommended for using until solved*.

- TB_Koster_Slater.f90 – contains some subroutines for the Koster-Slater angular parameterizations up to *d*-orbital [15].
- TB_Molteni.f90 – contains subroutines to calculate TB Hamiltonian within the sp^3s^*-basis set and repulsive energy, and corresponding forces, according to Molteni *et al.*[16].
- TB_NRL.f90 – contains subroutines to calculate TB Hamiltonian within the sp^3d^5-basis set and corresponding forces, according to NRL format [17].
- TB_Pettifor.f90 – contains subroutines to calculate TB Hamiltonian within the sp^3-basis set and repulsive energy, and corresponding forces, according to Pettifor *et al.*[18].
- Transport.f90 – contains simple rate equations mimicking heat transport out of the system using Berendsen thermostat [19].
- Universal_constants.f90 – this file contains all universal constants.
- Use_statements.f90 – contains all 'use' statements included in the main file.
- Van_der_Waals.f90 – this file contains all subroutines to deal with van der Waals potential.
- Variables.f90 – this file contains all global variables (mainly as defined objects) used throughout the code.
- ZBL_potential.f90 – ZBL potential [20] for core-core repulsion, only used for construction of dimer repulsive potentials for BOP tight binding parameterization.

Additionally, the following modules for post-processing of the data can be compiled (stored in the directory !XTANT_ANALYSIS_SUBROUTINES):

5) XTANT_atomic_data_analysis.f90
6) XTANT_autocorrelators.f90
7) XTANT_coupling_parameter.f90
8) XTANT_dielectric_function_analysis.f90

VIII. INPUT FILES

The code requires input files stored in the directory: INPUT_DATA. This name cannot be changed. The directory contains the following files and directories:

- INPUT_MATERIAL.txt – input file with all the parameters of the material and laser pulse.
- NUMERICAL_PARAMETERS.txt – input file with all the numerical parameters of the calculations.

These files cannot be renamed, the program has to be able to find them by these exact names.
The directories with the following names must also be present:

- Atomic_parameters must contain the following databases:
  - EADL2017.all – Electronic atomic database (ionization potentials, Auger- and radiative decay rates, kinetic energies of atomic electrons etc.)
  - EPDL2017.all – Photoionization cross sections database for all elements.

- INPUT_atomic_data.dat – periodic table of elements.
- INPUT_Hubbard_U.dat – table with Hubbard U parameters for (chemical hardness) for selected elements, according to ThreeBodyTB model[8] [2].

- DFTB containing directories with Slater-Koster files within DFTB format, e.g. matsci-0-3, and others. Inside of the directories .skl files must be present, named [*El*]-[*El*].skf, where [*El*] stands for the element which overlap parameters with the second listed element this file contains[9].
- 3TB_PARAMETERS – files containing ThreeBodyTB parameterizations for elemental solids and binary compounds in xml format
- BASIS_SETS – files with Gaussian basis sets in the gbs format[10] (*currently unused, since xTB or ab-initio model is unfinished*)
- BOP_data – all the dimer parameters in BOP format in the file models.bx [3]

A few other folders with the names of the materials must be there. The material name given in the INPUT_MATERIAL.txt (see below) must exactly coincide with the name of the folder, such as e.g.:

- Diamond, Silicon, Gold, etc. Each of the folders contains a few files describing the material properties, used in the code as described below.

1. File INPUT_MATERIAL.txt

File INPUT_MATERIAL.txt contains the following lines, which must be exactly in this order, with exactly as many numbers inside in each line, as described below:

```
1  SiAl            ! material name
2  SiAl            ! chemical formula of the compound
3  300.0d0         ! initial electron temperature [K]
4  300.0d0         ! initial atomic temperature [K]
5  -100.0          ! start of simulation [fs]
6  1000.0          ! end of simulation [fs]
7  1               ! number of FEL-pulses
8  2.0d0           ! absorbed dose per this pulse [eV/atom]
9  20.0d0          ! hw, photon energy [eV]
10 10.0d0          ! pulse FWHM-duration [fs]
11 1               ! type of pulse to be analysed: 0 = rectangular, 1 = Gaussian, 2 = SASE
12 0.0d0           ! position of the maximum of the laser pulse [fs]
13 0   1   1.0d-1  5.0d0   1.0d-1  ! optical coef: 0=no, 1=Drude, 2=Trani-k, 3=Trani gamma; spectrum (1) using KK (2), or no (0)
14 2   800.0   -70.0   ! how many rays (0=exclude, 1=1st ray, (>1)=sum all); probe-pulse wavelength [nm]; FWHM probe pulse [fs]
15 89.0    50.0d0      ! angle of prob-pulse with respect to normal [degrees]; material thickness [nm]
```

Figure VIII.1 INPUT_ MATERIAL.txt example.

- Line 1: material name, must exactly coincide with the name of the folder, where the material parameters are stored (mentioned above, described below).
- Line 2: chemical composition or element, which material consists of. Each element **must** start with a capital letter, followed by the small letters and/or number corresponding to the contribution of this element to the compound – that is how the program parses the names into separate chemical elements to be used for the periodic table reading. For example: diamond or graphite must be set as C; silicon – Si; gallium arsenide – GaAs; quartz – SiO2, etc.
- Line 3: initial electron temperature in [K].
- Line 4: initial atomic temperature in [K].

[8] The ThreeBodyTB code and its parameters can be found here: https://github.com/usnistgov/ThreeBodyTB.jl
[9] Detailed description of the files format is provided here: http://www.dftb.org/parameters/introduction/ The skl files can also be downloaded from there with the format that can be read by XTANT, no alterations needed.
[10] Basis set files can be downloaded from: https://www.basissetexchange.org/

- Line 5: starting time of simulation in [fs]. The starting time in the simulation will be chosen as the minimum between the user-provided value here, and [-50+FWHM*2.35], where FWHM is the full width at half maximum of the laser pulse (see line 12 below).
- Line 6: total duration of the simulation in [fs]. *Can be later changed during the simulation, see below description of the Communication file.*
- Line 7: number of FEL-pulses to be simulated (multiple-pulses allowed). Number of next lines depends on this. In this example of Fig.1, there is only 1 pulse to be modelled. In case you want to model two pulses, set here 2.
- Next lines specify the parameters of each FEL-pulse:
    - Line 8: absorbed dose in [eV/atom], used for energy deposition from this pulse. Setting it equal to 0 gives NO laser pulse, dynamics of unirradiated system (for example, for electron-ion thermalization, if nonadiabatic coupling is included).
    - Line 9: photon energy of the incoming FEL in [eV].
    - Line 10: duration of the pulse, τ, in [fs] (FWHM for Gaussian pulse, total duration for flat-top or SASE [21]).
    - Line 11: type of the pulse to be used: 0 means flat-top pulse, 1 gives Gaussian, 2 mimics SASE-like spiky pulse [21].
    - Line 12: position of the center, t_0 (Gaussian maximum) of the laser pulse [fs]. The simulation will start at ($t_0 - \tau - 50$) fs, if this value is smaller than the starting time set by the user (line 5).
    - If you want to set a second pulse, repeat the same lines 8-12 (with different pulse parameters) in the same order.
- Line 13: this line contains 5 numbers:
  first one sets whether you want to calculate evolution of the optical properties (set 0 if not), and within which model:
  a) 1 for Drude model.
  b) 2 for Trani model at many k-points, distributed according to user-defined grid or to Monkhorst-Pack grid [22] (this option requires many diagonalizations of TH Hamiltonian which are currently not parallelized, and thus is very slow).
  c) 3 for Trani model [12] at the Gamma-point only.

  Note also that if you want to calculate DOS of the material (see option in the numerical parameters file below) but *not* the optical properties, you can set here a negative value: -2 will calculated DOS on multiple k-points specified; setting -3 will provide DOS at gamma point.

  The second number in this line indicates whether you want to calculate the complex dielectric function only for a given (probe-photon) energy (set 0), or for the whole spectrum (set 1). Default choice is 0.

  The next three numbers define the interval of the spectrum you'd like to calculate, in case the previous number is set to 1:

  The third number is the starting point in [eV], the fourth is the ending point in [eV], and the fifth is the energy step in [eV] to make a uniform grid. If any of these three numbers is set negative, then default values for the interval are used, which are: from 0 to 50 eV with the step of 0.05 eV.

  The numbers in the raw must be separated by TAB, not SPACE.

- Line 13: contains three numbers:
  - First one sets for how many rays propagation you want to calculate the optical parameters (transmission, reflection, and absorption of the probe-pulse): set 1 for the first ray, or a value larger than 1 for summing up all rays. For femtosecond probe pulse, the default choice is 1 (however, for very thin samples, thinner than ~50 nm, sum up all).
  - The second number sets the wavelength of the probe pulse in [nm].
  - The third one is the duration of the probe pulse in [fs]. If the number if set to a positive value, the output files will be additionally convolved with the Gaussian probe pulse of the given duration. A set of additional convolved output data will be created with the tag 'CONVOLVED' (see below). To exclude this option, set the duration to zero or a negative value.

  The numbers in a raw must be separated by TAB, not SPACE.
- Line 14: contains two numbers:
  - First one sets the angle of incidence of the probe-pulse in degrees to the normal.
  - Second one sets the thickness of the material layer through which the probe pulse absorption and reflection are calculated in [nm]. Must be equal to the experimental target thickness, if it is thinner than the FEL photon attenuation length; or may be to the FEL-photon attenuation length otherwise. The numbers in a raw must be separated by TAB, not SPACE.

2. File NUMERICAL_PARAMETERS.txt

File NUMERICAL_PARAMETERS.txt contains the following lines, which must be exactly in this order, with exactly as many numbers inside in each line, as described below:

```
2   3   4        ! number of unit-cells in X,Y,Z
1   1   1        ! periodicity along X,Y,Z directions (1=yes, 0=no)
EADL             ! where to take material parameters from (EADL, CDF, XATOM...)
-1.0d0      ! [g/cm^3] density of the material (used in MC in case of EADL parameters), if <0 uses data from MD supercell
1           ! number of MC ITERATIONS
1           ! number of threads for OPENMP
2           ! MD algorithm: 0=Verlet (2d order); 1=Yoshida (4th order, slow); 2=Martyna (4th order, fast)
1           ! frozen atoms (=0), or moving normally (=1)
25.5d0      ! Parinello-Rahman super-cell mass coefficient
0.2d0   dt_grid.txt ! time step for MD [fs]
1.0d0       ! printout data into files every 'dt_save_time' [fs]
0           ! it's = 1 if P=const, or = 0 if V=const
0.0d0       ! external pressure [Pa]
T   0   0.35 ! include SCC (True, False), model of gamma (default 0); mixing factor (default 0.35)
0   1d-3    ! scheme (0=decoupled electrons; 1=enforced energy conservation; 2=T=const; 3=BO);  when to start coupling
0           ! -1=nonperturbative (default), 0=no coupling, 1=dynamical coupling, 2=Fermi golden rule
-1.0d+3    4.0d0    ! [fs] when to switch on the nonadiabatic coupling; scaling factor (if needed)
5.0d0   0.001d0     ! [eV] acceptance window for nonadiabatic coupling; [eV] tollerance for quasidegenerate levels
1   5.0 2.0     ! quenching (0=no, 1=yes); starting from when [fs]; how often [fs]
0   300.0   100.0   ! Berendsen thermostat for atoms (0=no, 1=yes); bath temperature [K]; cooling time [fs]
0   300.0   10.0    ! Berendsen thermostat for electrons (0=no, 1=yes); bath temperature [K]; cooling time [fs]
15.0d0      ! [eV] cut-off energy in MC (<0 means Ecut=top of CB)
1.0d30      ! [eV] work function, for electron emission (>0 eV; <0 - number of collisions)
0           ! printout electron energy levels (1) or not (0)
0   0.1 1   ! printout DOS (1) or not (0); smearing of gaussian; print PDOS yes (1) or no (0)
1           ! printout Mulliken charges for types of atoms
0       ! printout electron distribution (1) or not (0)
0       ! printout atomic pair correlation function (1) or not (0)
1       ! printout atomic positions in XYZ (1) or not (0)
0       ! printout atomic positions in CIF (1) or not (0)
0       ! printout raw data file OUTPUT_coordinates_and_velosities.dat (1) or not (0)
1       ! power of mean displacement to print out (set integer N: <u^N>-<u0^N>)
-2.0e0   ! printout numbers of nearest neighbors within the given radius (<=0 NO, >0 = radius in [A])
png     ! which format to use to plot figures: eps, jpeg, gif, png, pdf
9   9   9        ! number of k-points in each direction (used only for Trani-k!)
7.760d0    0.426d0    ! initial n and k of unexcited material (used for DRUDE model only!)
0.35d0    0.35d0      ! effective electron and hole masses [free-electron mass] (used for DRUDE model only!)
1.d0    1.d0          ! [fs] mean scattering times of electrons and holes (used for DRUDE model only!)
```

Figure VIII.2. NUMERICAL_PARAMETERS.txt example.

Line 1: number of unit-cells used in the code along each direction X, Y, Z. If one unit cell contains Nat atoms, the total number of atoms in the supercell will be **Ntot**=Nat*Nx*Ny*Nz.

a. The numbers in a raw must be separated by TAB, not SPACE.

b. *Setting here 0 0 0 should in principle exclude the atomic dynamics and run only electronic MC simulations (analogous to XCASCADE code*[9]*); this option, however, has not yet been tested!*

Line 2: three numbers specify conditions at surfaces along X, Y, Z axes: setting here 0 creates an open surface along the axis (by adding empty space around the sample), whereas setting here 1 means periodic boundaries. For example: 1 1 0 means periodic boundaries along X and Y, but free boundary along Z (thin layer of material).

Note: Non-periodic simulation uses periodic boundary in a supercell, in which the sample is surrounded by empty space. The code increases the size of the simulation box by 50 times of its given value, and places the atoms in the middle. That means, all the values that include normalization to the supercell volume (i.e. electron-ion coupling parameter, electron heat capacity, pressure) will include the empty space volume, and must be rescaled manually for interpretation of the results.

Line 3: which cross sections to use in the MC module: set here CDF (a cdf-file must be provided, see below) to use cross sections based on the complex dielectric function formalism [23]; or EADL to use atomic BEB-cross-sections (default choice) [8,9].

Line 4: density of the material in [g/cm^3] to be used in the MC simulations. This value overwrites the default value given in the cdf-file (see below), if set positive. If you wish to use the default value (defined by the number of atoms and size of the supercell), set here any negative number (this must be the default choice).

Line 5: number of iterations to be performed within the Monte Carlo module. Small number of iterations gives not smooth curves. Too large numbers give too long computation times. Optimal value empirically determined is ~2,000,000/(Dose * Ntot). *Note that large values here result in large arrays taking a lot of memory. In case you don't need MC simulations but only TBMD, set here 1.*

Line 6: number of threads used for parallel calculation via OpenMP. Set 1 for nonparallel calculations. Set the number equal to the number of cores in your machine for optimal fast calculations.

Line 7: which MD integrator to use: 0 = velocity Verlet algorithm (2^{d} order) [24], 1=Yoshida algorithm (4th order; *it is 4 times slower than Verlet*) [25], 2 = Martyna predictor-corrector algorithm (4th order, *as fast as Verlet*) [26]. The default option is 2. *Note that Martyna algorithm is included only for atomic coordinates, while for the supercell vectors Verlet is used (2d order).*

Line 8: exclude MD module, freezing the atoms in their equilibrium positions (if set 0), or allow the atoms to move (set 1). The default option is 1.

Line 9: effective mass of the super-cell in [atomic mass] used in the framework of the Parrinello-Rahman MD [27] (only used in case of constant-pressure simulation, see below).

Line 10: time-step for the MD calculations.

a. It can be set in two ways:

Constant time-step: write any real(8) number of dt in [fs]. Default value is 0.1 fs (or smaller for P=const simulations, see below), but in some simulations can be as large as 1 fs or even larger (especially for V=const simulations). Larger steps can lead to instabilities, smaller steps conserve energy better but run the program slower.

Variable time-step: write here a name of a file where the array of timesteps is provided. The file must be present in the same folder INPUT_DATA. The file must contain two columns:

b. First column: time instant in [fs], when to change the timestep to the one given in the second column. E.g., if the file contains the following lines:
c. -1.0e10 1.0
d. -50 0.2
e. 100 0.5
f. At the first time step of the simulation, the timestep is set 1.0 fs. At the time instant of -50 fs, the timestep is changed to 0.2 fs. At the time instant of 100 fs, it is changed to 0.5 fs.

Line 11: time step how often the output data files must be saved in [fs] of the simulation time. Does not have to be equal to the MD time step, can be larger to sparse the output data (but cannot be smaller). Default choice is 1 fs.

Line 12: here 1 means constant pressure simulations (P=const; Parrinello-Rahman scheme of the super-cell motion, NPH ensemble); 0 means constant volume (V=const, NVE ensemble) [28]. For femtosecond dynamics, V=const (0) is the default choice.

Line 13: external pressure applied (set 0 to use for normal atmospheric pressure). Used only in case of P=const simulation.

Line 14: contains 3 numbers to set self-consistent-charge (SCC) calculations parameters:

a. Column 1: to use or not the SCC (*T* or *F* - for True or False)
this option works <u>only</u> with the Born-Oppenheimer approximation (option 3 in the next line), but not with any other simulation scheme, so cannot be combined with irradiation!
b. Column 2: which model for gamma to use: -1 is bare Coulomb (*do not use, only for testing*)
0 = Wolf's method of softly truncated Coulomb (this is the default choice) [29]
1 = Klopman-Ohno [30]
2 = Mataga-Nishimoto [30]
c. Column 3: mixing factor for self-consistent calculations: weight of the new charge in the next iteration. Recommended values are between 0.2 and 0.7. Smaller values lead to too slow convergence, whereas higher values may not converge at all. The empirically found optimal value is 0.35, which should be the default choice.

Note that the default option <u>must be</u> *F* (no SCC). It should only be used with TB parameterizations that are specifically fitted to account for SCC (such as 3TB, and some of DFTB parameterizations) and with BO simulation *only*, while in most cases, the parameterizations cannot account for the SCC effects, and including this option may lead to qualitatively incorrect potentials.

Line 15: contains two numbers:

a. First one describes which scheme of simulation to use:

0 sets a scheme of decoupled electrons and ions, with instant electron thermalisation (something like Two-Temperature Model). <u>This option must be the default choice.</u>

1 sets enforced total energy conservation. *Obsolete, do not use!*

2 sets here fixed electron temperature instead of total energy (does not work well with a pulse on or with electron-ion coupling on, only for tests of unirradiated material).

3 uses the true Born-Oppenheimer scheme – constant electron populations (does not work well with a pulse on or with electron-ion coupling on, only for tests of unirradiated material).

b. Second number tells when the scheme should start to be used. The default choice is 1.0d-3, meaning immediately after the beginning of simulation. *Strongly recommended not to change!*

The numbers in a raw must be separated by TAB, not SPACE.

Line 16: which electron-phonon coupling to use:

a. 0 means no coupling included;

-1 generalization of dynamical coupling as described in Ref.[31] (this must be the default choice);

1 means first-order dynamical coupling as described in Ref.[31];

2 uses Fermi's Golden Rule, which might overestimate the coupling rate (*do not use*).

Line 17: Coupling model; has two numbers:

a. Sets the time, when the nonadiabatic coupling switches on (for test purposes, one can first thermalize the system, and only later let it exchange energy). Default choice for real simulations: 1d-3.

b. Set the scaling factor for coupling calculations. For numerical reasons, must be equal to 4.0 for producing correct results with dynamical coupling. Can be smaller to artificially reduce coupling in the calculations of the coupling parameter (as described in section Calculation of electron-ion coupling parameter $g(T_e)$, $C_e(T_e)$ and $\mu(T_e)$), but the results must then be rescaled back manually!

Line 18: Coupling model (continuation); contains two numbers:

a. acceptance window for nonadiabatic coupling in [eV]. It excludes electron transitions between the levels separated by more than this specified value. E.g. set 5 eV by default to separate over-band-gap nonadiabatic transitions in diamond.

b. tolerance for quasi-degenerate levels in [eV]. It excludes transitions between too close levels, separated by smaller energy than this given number, to exclude degenerate states. The default value is 0.001 eV.

Line 19: Quenching; three numbers here specify:

The first number defines whether to include artificial quenching, (0=no, 1=yes), the 'yes'-option must be used *only* for construction of amorphous materials. Any 'real' simulation must have 0 here. 'Yes' here means that once in a time-step specified by the next numbers of the line, atomic velocities will be set to zero. Similar to the method known as "zero-temperature molecular dynamics".

The second number in this line is defining when to start cooling from in [fs].

The third number means how often set the atomic velocities to zero (in [fs]).

Line 20: Berendsen thermostat for atoms; Three numbers here define a simple model (rate equation) for artificial cooling mimicking transport effects [32], using Berendsen thermostat [19]:

First number: include electron heat transport out of the atomic system (1), or not (0).

Second number: in case if there is transport, sets the atomic bath temperature towards which the cooling will be made until temperatures equilibration in [K].

The third one is the characteristic time of cooling of atoms in [fs].

Line 21: Berendsen thermostat for electrons; Three numbers here define a simple model (rate equation) for artificial cooling mimicking transport effects [32], using Berendsen thermostat [19]:

First number: include electron heat transport out of the electronic system (1), or not (0). *This option is recommended always to be set to 0 (no transport!).*

Second number: in case if there is transport, sets the electronic bath temperature towards which the cooling will be made until temperatures equilibration in [K].

The third one is the characteristic time of electronic cooling in [fs].

Line 22: energy cut-off in [eV] that separates the low-energy and high-energy subspaces for electrons within MC and Boltzmann-equation [33]. The default value is 10 eV. If set negative, it uses dynamical evolution of the cut-off, adjusting it to the transient top-most CB level at each time-step (*only meaningful for small basis sets that do not include a lot of CB orbitals*).

Line 23: work function, setting whether we want to allow for electron emission and build-up of an unbalanced charge in the system, which may lead to Coulomb explosion. If the work function here is set higher than 1.0d25 [eV], no emission will take place. This must be the default choice.

a. If the work function is set smaller, an electron with energy above the set number will be considered emitted from the sample, and will disappear from the calculations (will forever stay in the high-energy domain, making no collisions).
b. If the work function is set to a negative value, another model for electron emission is used: an electron is considered to be emitted after a certain number of collisions that is specified by the absolute value of the number set in this line. E.g. if -2.0 is set here, an electron will be emitted after performing 2 collisions (unless is falls below the cut-off energy and joins the low-energy fraction).
c. If electrons are emitted, it builds up an uncompensated charge for the atomic system, inducing additional Coulomb repulsion of atoms (ions), if a file with Coulomb parameterization is present (see below) [34].

Line 24: to print out electron energy levels (eigenvalues of the TB Hamiltonian) as output (set 1) or not (set 0) at each saving time step. *Produces large files.*

Line 25: the three numbers here specify:

to calculate total density of states, DOS, (set 1), or not (set 0) at each saving time step. If 1 is specified here, the model of k-points calculations specified in the INPUT_MATERIAL.txt file is used (gamma-point of multiple k-points). The default value is 0.

which spreading to use for constructing the DOS out of the discrete energy levels in [eV].

to calculate partial DOS (PDOS) for the atomic shells of each element in the compound (1), or not (0). Note that if the first parameter (responsible for calculating total DOS) is set to 0, the PDOS will also not be calculated, independently of what is set here.

Line 26: to save Mulliken charges for types of atoms (set 1), or not (set 0)

Line 27: to save electron distribution function (set 1), or not (set 0) at each saving timestep. Produces large files. Most of the time, it is not necessary, thus use the default value 0.

Line 28: to save atomic pair correlation function (set 1), or not (set 0) at each saving timestep. Produces large files. Most of the time, it is not necessary, thus the default value is 0.

Line 29: save atomic positions additionally in xyz-format[11] (set 1), or not (0). This format is used for plotting and making movies of the atomic positions e.g. with VMD[12] or similar program.

Line 30: save atomic positions additionally in CIF-format[13] (set 1), or not (0). This format is used for powder diffraction patterns calculations e.g. with Mercury[14] software.

[11] See description here: http://en.wikipedia.org/wiki/XYZ_file_format
[12] VMD: http://www.ks.uiuc.edu/Research/vmd/
[13] See description here: http://www.iucr.org/resources/cif
[14] Mercury: https://www.ccdc.cam.ac.uk/solutions/csd-system/components/mercury/

Line 31: to save raw data file OUTPUT_coordinates_and_velosities.dat (1) or not (0). This file is necessary for post-analysis calculations of atomic velocity autocorrelations and phonon spectra (see below). If not needed, do not save it, as it produces very large files.

Line 32: sets *N*, the power of mean displacement to print out (set integer *N*: <u^*N*>-<u0^*N*>). For example, for mean square displacement, set 2.

Line 33: save numbers of nearest neighbours within the given radius: to exclude optional set a number <0, a number >0 means the radius within which the atoms considered to be neighbours in the units of [A].

Line 34: which format to use to plot output figures: eps, jpeg, gif, png, pdf

Line 35: contains three numbers: numbers of k-points in each direction x, y, z; used only with the 'Trani-k' option (number 2 in the line 12 in the file INPUT_MATERIAL.txt), and ignored with other options.

Next lines are used for the Drude model only (ignored for other optical models, but the lines with some numbers in the correct order must still be present in the file!):

Line 36: initial values on unexcited materials optical coefficients *n* and *k*.

Line 37: effective conduction band electrons and valence band holes masses in units of the free-electron mass.

Line 38: mean scattering times of electrons and holes in [fs]

Line 39: This and next lines are optional, each block may be written in arbitrary order, or skipped altogether (recommended). The optional blocks allow the user to overwrite the default atomic data, such as the atomic mass, Auger decay times, kinetic energies of shell electrons, electronic populations, and the name of the element, if needed. The default values are taken from the EPICS-2017 database, and recommended for use, so only replace them if you are absolutely sure in what you are doing.

- To replace an atomic mass, the following block of data must be used:

Line Op1: MASS – the keyword, indicating that the following line defines the atomic mass

Line Op2: must contain two numbers: *atom* (integer), *mass* (real).
The *atom* must correspond to the number of the element in the used compound, as defined by the chemical formula in the line 2 of the file 'INPUT_MATERIAL.txt'. For elemental targets, the number 1 still must be present.
The *mass* sets the mass of the element in the atomic mass units. See example of a compound SiAu, setting the mass of Au (element #2) to "infinity" (1.0d30), in **Figure VIII.3**. Note that such a mass will essentially freeze all the atoms of the chosen element (which will not speed up the calculations, since the force is still calculated for them, just they are too heavy to move).

```
MASS
2   1.0d30

NO_AUGER

AUGER
1       1       1d25
```

Figure VIII.3. Example of optional lines in the file NUMERICAL_PARAMETERS.txt

- To replace a name of the element, the following block of data must be used:

Line Op1: NAME – the keyword, indicating that the following line defines the name of element

Line Op2: must contain two numbers: *atom* (integer), *name* (character(3)).
The *atom* is analogous to the one from the block MASS above.
The *name* sets the new name of the element.

- To switch off all Auger decays in MC module, use the one-line option NO_AUGER, see an example in **Figure VIII.3**.
- To replace Auger decay times of selected shells of selected elements, use the optional block (see an example in **Figure VIII.3**):

Line Op3: AUGER – the keyword indicating overwriting of Auger decay times in this block.

Line Op4: three numbers: atom (integer), shell (integer), time (real).

The number *atom* sets the number of the element in the compound, analogously to the block MASS above, following the numbers set in the chemical formula in the input file.
The number *shell* sets the atomic shell, for which the Auger decay time must be replaced. The shells numbers are printed out in the output file !OUTPUT_[*material*]_Parameters.txt, see below. The number of the shell must coincide with the number printed out in this file.
The number *time* sets the Auger decay time for the given shell of the given element in [fs].

- To replace ionization potentials of selected shells of selected elements, use the optional block:

Line Op5: Ip – the keyword indicating overwriting of ionization potentials in this block.

Line Op6: three numbers: *atom* (integer), *shell* (integer), *ionization_potential* (real).

The numbers *atom* and *shell* are analogous to those in the block Auger (see above).

The *ionization_potential* sets the ionization potential in the shell in [eV].

- To replace kinetic energy of electrons in selected shells of selected elements (this value is only used in the BEB cross sections in MC module), use the optional block:

Line Op7: Ek – the keyword indicating overwriting of kinetic energies in this block.

Line Op8: three numbers: *atom* (integer), *shell* (integer), *kinetic_energy* (real).

The numbers *atom* and *shell* are analogous to those in the block Auger (see above).

The *kinetic_energy* sets the kinetic energy of electrons in the shell in [eV].

- To replace the number of electrons in selected shells of selected elements (electron population of the atomic shell), use the optional block:

Line Op9: Ne – the keyword indicating overwriting of electronic populations in this block.

Line Op10: three numbers: *atom* (integer), *shell* (integer), *population* (real).

The numbers *atom* and *shell* are analogous to those in the block Auger (see above).

The *population* sets the number of electrons in the shell of a given element.

3. Executing consecutive runs of the program automatically

If you want to run XTANT program several times in a raw with different parameters (useful, e.g., for finding damage threshold by varying only the pulse fluence while keeping all others parameters the same, or for calculations of electron-phonon coupling parameter vs. electronic temperature), you can create several input files in the following manner:

First simulation run will use as input files

INPUT_MATERIAL.txt and NUMERICAL_PARAMETERS.txt

After the end of simulation, the program will check presence of the next input files in the same folder named with consecutive integer number at the end of file names:

INPUT_MATERIAL_1.txt and NUMERICAL_PARAMETERS_1.txt

If they are present, XTANT will read the data from these files and start simulation over from the beginning automatically. Next automatic simulation run must have the next integer number at the end of the files (…_2), and so on.

This is a very useful option for calculations of the coupling parameter dependence on the electronic temperature that requires averaging over many simulation runs.

4. Folder [*material name*]

Folders with material parameters, named exactly as the material name given above in the file INPUT_MATERIAL.txt, contain a number of files describing necessary material parameters for the simulation.

Do not change these files, unless you want to change your properties of the material!

Folder with the files for already created materials are typically already set, and the user does not need to worry about them. If you want to create a new material, create it by example of already existing folders.

a) [*A*]_[*A*]_TB_Hamiltonian_parameters.txt and [*A*]_[*A*]_TB_Repulsive_parameters.txt

These files contain all the parameters used in the tight binding Hamiltonian for each pair-wise interaction of atoms [*A*]. These files contain attractive and repulsive parts, correspondingly. For instance, in case of diamond we only have carbon atoms, thus only files C_C_TB_Hamiltonian_parameters.txt and C_C_TB_Repulsive_parameters.txt should be in the folder. In case of GaAs, combination of each interaction should be present in the files: Ga-Ga, Ga-As, and As-As.

In case identical parameters are used for multiple combinations of elements (this is possible within DFTB parameterization files as they only contain links to databases), one can use files: TB_Hamiltonian_parameters.txt and TB_Repulsive_parameters.txt. In a case XTANT cannot find files

[*A*]_[*A*]_TB_Hamiltonian_parameters.txt or [*A*]_[*A*]_TB_Repulsive_parameters.txt, it will look up the files without "[*A*]_[*A*]_" prefix.

This also means that one can set default parameterizations in the files without prefix, and for specific elements use the files with the prefix to specify exceptions.

First line of the files define which TB model will be used. The models supported can be specified by the following code words:

- 3TB : ThreeBodyTB model for elemental solids and binary compounds (*only standard 2-body TB is implemented currently, the three-body contribution is unfinished, see below*)
- BOP : bond-order potential model for dimers [3] (*unfinished, do not use!*)
- DFTB : one of the s-, sp^3- or sp^3d^5-basis set according to DFTB [4,35]
- Fu : sp^3-basis set according to Fu [14] (*unfinished, do not use!*)
- Mehl : sp^3d^5-basis set according to NRL model [36]
- Molteni : sp^3s^*-basis set according to Molteni [16]
- Pettifor : sp^3-basis set according to Pettifor [18]
- xTB : extended tight binding [5] (*unfinished, do not use!*)

Thus, the first line in the file must contain one of the possible names of the parameterization. Depending on that, the further lines will define one or another parameterization:

1) 3TB parameterization

```
3TB
9.0     0.1      ! cut off [A], smoothing [A]
1.06             ! rc, rescaling coefficient
F                ! include 3-body terms or not (true, false)
T                ! exclude diagonal part of crystal field
```

Figure VIII.4. 3TB TB_Hamiltonian_parameters.txt

```
3TB
```

Figure VIII.5 3TB TB_Repulsive_parameters.txt

The files contain the parameters, according to ThreeBodyTB[15] parameterization, which may be s, sp^3 or sp^3d^5, depending on the element.

Line 1 specifies the model name, must be "3TB"

Line 2 defines embedding cut-off function parameters, ensuring the interaction is short-ranged

Line 3 defines rescaling coefficient *rc* in the Laguerre polynomials for the TB radial function: $L(d*rc)$, which shifts the location of the potential minimum, allowing to adjust the minimum to a desired value. A typical value here is between 1 and 1.1.

Important note: In certain cases, it is important to adjust it prior to productive calculations. Additionally, rigorous tests must be performed for each material: the shape of the cohesive energy curve and the stability of the lattice must be checked, since the parameterization was not designed for MD runs (especially in highly excited systems), and not all materials are stable and behaving well dynamically.

Line 4 currently <u>must</u> have the option *F*, meaning three-body interactions must be excluded, because this option is not yet fully implemented in XTANT.

[15] https://pages.nist.gov/ThreeBodyTB.jl/

Line 5 must have the value *T*, since the diagonal part of the crystal field should be excluded in calculations.

2) BOP parameterization

```
BOP
40.0e0  0.5e0        ! rcut [A], dcut [A]
```

Figure VIII.6. BOP TB_Hamiltonian_parameters.txt

```
BOP
```

Figure VIII.7 BOP TB_Repulsive_parameters.txt

The files contain the following lines, see for example Figure VIII.6 and Figure VIII.7.

3) DFTB parameterization

```
DFTB
matsci-0-3       ! DFTB parameterization
3.5d0    0.1d0          ! cut-off radius and smoothing dist
```

Figure VIII.8. TB_Hamiltonian_parameters.txt

```
DFTB
matsci-0-3          ! DFTB parameterization
1                   ! Type of repulsive: 0=polinomial, 1=spline
```

Figure VIII.9 TB_Repulsive_parameters.txt in DFTB format

The files contain the following lines, see for example Figure VIII.8 and Figure VIII.9 of C interaction with C (e.g. in diamond). In the Hamiltonian file (Figure VIII.8):

- The first line sets the parameterization (DFTB).
- The second line sets the set of SK parameters to be used according to DFTB, which must coincide exactly with the directory name existing within the DFTB directory.
- The third line defines the soft cut-off radius in [A] and its smoothing in [A]. Those can be adjusted empirically, but should not be larger than ~10 A, as this is typically the limit in the SK files of DFTB. It usually makes sense to set the cut-off radius somewhere after the second or third nearest neighbour.

In the Repulsive parameters file (Figure VIII.9):

- The first line sets the parameterization (DFTB). Must be the same as in Hamiltonian file.
- The second line sets the set of SK parameters to be used according to DFTB, which must coincide exactly with the directory name existing within the DFTB directory. Must be the same as in Hamiltonian file.
- The third line defines which form of the Repulsive term to use according to DFTB format: polynomial (0) or spline (1). It is recommended to use spline (since it contains exponential repulsion as short distances, which polynomial does not; also, apparently not in all sk-files polynomial form is given), thus 1 is the default choice.

All the parameters within the skf-files are described on the dftb-website[16]. Do not change those files (unless you know what exactly you want to change in the parameterization of TB Hamiltonian)!

Also note that currently XTANT only supports zero-level DFTB, non-self-consistent calculations.

[16] http://www.dftb.org/parameters/introduction/

4) *Fu parameterization*

The files contain parameters in exactly the same format as Pettifor, where only the first line should read "Fu" instead of "Pettifor". See description below. *Not recommended for use!*

5) *Mehl parameterization*

The Hamiltonian files should contain 3 introductory lines and then a set of 97 parameters as defined in the NRL format. The Repulsive potential files should contain only one line "Mehl", identifying the parameterization. They are all already prepared, so no need to change anything. To set new parameters for a new material, use existing files as an example (starting from third line, only the first column of parameters is used in the code, the others are comments explaining the meaning of the parameters).

```
NUMERICAL_PARAMETERS.txt | INPUT_MATERIAL.txt | C_C_TB_Repulsive_parameters.txt | C_C_TB_Hamiltonian_parameters.txt | C_
1   Mehl
2   1   0   1          ! parameterization index (1=with split between t2g and e2, or 0=without); (Sll'm 1=with
3   10.5   0.5                   (RCUT and SCREENL for 1-1 interactions)
4     0.159901905594E+01   0   1     lambda  (equation 7)
5     -.102789972814E+00   0   2     a_s     (equation 9)
6     -.162604640052E+01   0   3     b_s     (equation 9)
7     -.178884826119E+03   0   4     c_s     (equation 9)
8      .451611342028E+04   0   5     d_s     (equation 9)
9      .542619178314E+00   0   6     a_p     (equation 9)
10     .273454062799E+01   0   7     b_p     (equation 9)
11    -.671397098834E+02   0   8     c_p     (equation 9)
12     .438528831452E+03   0   9     d_p     (equation 9)
13     .100000000000E+02   1  10     a_t2g   (equation 9)
14     .000000000000E+00   1  11     b_t2g   (equation 9)
15     .000000000000E+00   1  12     c_t2g   (equation 9)
16     .000000000000E+00   1  13     d_t2g   (equation 9)
17     .100000000000E+02   1  14     a_eg    (equation 9)
18     .000000000000E+00   1  15     b_eg    (equation 9)
19     .000000000000E+00   1  16     c_eg    (equation 9)
20     .000000000000E+00   1  17     d_eg    (equation 9)
```

Figure VIII.10 File TB_Repulsive_parameters.txt in NRL format

Line 1 must be "Mehl" identifying the parameterization name

Line 2 contains three numbers:

- First: included split between t2g (1) or excluded (0), as specified in an original NRL parameters file[17][17]
- Second: includes the terms Sll'm with delta (1) or without (0)
- set f_bar = 0 or not (=1)

Line 3 sets the cutoff distance in [A] and its smoothing in [A].

All these parameters must be extracted from the original NRL .par files. The rest of the lines must be just copied from NRL .par files without any changes. The available NRL .par files are saved in the directory NRL_TB_database and can be used to construct required parameterizations for chosen materials.

Note that NRL parameterization often needs additional short-range repulsive potential [37], as may be defined in a "wall" file, see below.

[17] The NRL parameters files used to be available at http://cst-www.nrl.navy.mil/bind/ , and later at http://esd.spacs.gmu.edu/tb/tbp.html , but currently can be extracted by means of internet archives such as Wayback Machine: https://archive.org/web/

6) *Molteni parameterization for sp³s* basis set*

The files contain the following lines, see Figure VIII.11 and Figure VIII.12 for example of Ga interaction with As (GaAs). All the parameters are described in [16,38]. Do not change these files (unless you know what exactly you want to change in the parameterization of TB Hamiltonian).

```
INPUT_MATERIAL.txt | NUMERICAL_PARAMETERS.txt | Si_Si_TB_Hamiltonian_p
Molteni
-2.657d0      ! Es, on-site energy of s-orbital [eV]
3.669d0       ! Ep, on-site energy of p-orbital [eV]
6.739d0       ! Es*, on-site energy of s*-orbital [eV]
-1.613d0      ! V0 (s s sigma)
2.504d0       ! V0 (s p sigma)
3.028d0       ! V0 (p p sigma)
-0.781d0      ! V0 (p p pi)
2.082d0       ! V0 (s* p)
2.46d0        ! r0
2.0d0         ! n
3.511d0           ! rc
13.0d0        ! nc
1.3d0         ! N; rcut = N*r0
0.1d0         ! d
```

Figure VIII.11. File TB_Hamiltonian_parameters.txt in Molteni forma

```
INPUT_MATERIAL.txt | NUMERICAL_PARAMETERS.txt | Si_Si_TB_
Molteni
1         ! NP, type of potential
2.3906d0     ! phi1
1.2347d0     ! phi2
2.46d0       ! r0
1.3d0        ! N; rcut = N*r0
0.1d0        ! d
0.3555d0     ! alpha (NP=1) or m (NP=2)
```

Figure VIII.12 File TB_Repulsive_parameters.txt

7) *Pettifor parameterization*

The files contain the following lines, see Figure VIII.13 and Figure VIII.14 for example of Carbon. All the parameters are described in [33,39].

```
INPUT_MATERIAL.txt | NUMERICAL_PARAMETERS.txt | Si_Si_TB_Hamiltonian_parar
Pettifor
-5.25d0       ! Es, on-site energy of s-orbital of Si [eV]
1.20d0        ! Ep, on-site energy of p-orbital of Si [eV]
-2.038e0      ! V0 (s s sigma)
1.745e0       ! V0 (s p sigma)
2.75e0        ! V0 (p p sigma)
-1.075e0      ! V0 (p p pi)
2.360352e0    ! r0
2.0e0         ! n
4.0e0         ! r1
4.16e0        ! rm
9.5e0         ! nc(1)
8.5e0         ! nc(2)
7.5e0         ! nc(3)
7.5e0         ! nc(4)
3.5e0         ! rc(1)
3.55e0        ! rc(2)
3.7e0         ! rc(3)
3.7e0         ! rc(4)
-6.4651e-5    ! c0(1)
0.0014704         ! c1(1)
-0.010804         ! c2(1)
0.025871      ! c3(1)
0.0026017     ! c0(2)
-0.031794         ! c1(2)
0.092548      ! c2(2)
0.028371      ! c3(2)
0.028336      ! c0(3)
-0.20485      ! c1(3)
-0.76001      ! c2(3)
5.8340        ! c3(3)
-0.011077         ! c0(4)
0.08077       ! c1(4)
0.29709       ! c2(4)
-2.2806       ! c3(4)
```

Figure VIII.13. TB_Hamiltonian_parameters.txt in Pettifor format

```
INPUT_MATERIAL.txt | NUMERICAL_PARAMETERS.txt
Pettifor
8.7393204    ! E0_TB [eV]
1.0e0        ! phi0
6.8755e0     ! m
13.017e0     ! mc
2.360352e0   ! d0
4.0e0        ! d1
4.16e0       ! dm
3.66995e0    ! dc
1.8790e-11   ! c0
1.3221e-9    ! c1
1.4324e-8    ! c2
-4.2468e-8   ! c3
0.0e0        ! a0
2.1604385    ! a1
-0.1384393   ! a2
5.8398423e-3     ! a3
-8.0263577e-5    ! a4
```

Figure VIII.14 TB_Repulsive_parameters.txt

b) [*A*]_[*A*]_vdW.txt

This file contains the parameters used in the van der Waals (vdW) potential for each pair-wise interaction of atoms [*A*]. For instance, in case of diamond we only have carbon atoms, thus only files C_C_vdW.txt should be in the folder. At present, the vdW potential is set according to Girifalco's model, in the shape of the Lennard-Jones 12-6 potential (see e.g. [40]), smoothly cut at large (for better energy conservation, and to limit it to some reasonable distance in the spirit of TB) as well as at short distances (not to overlap with the TB covalent bonds); see description of how the smooth cut-offs are constructed in Ref.[34].

The first line in the file must contain the exact name of the parameterization: Girifalco. An example of the file is shown in Figure VIII.15. By default, it is now set to something different, to exclude the vdW potential from the calculations. Change it to the proper name to include it.

If the file is absent, the calculations will proceed without vdW forces. So, this module will not affect the materials for which there is no vdW force or parameterization.

```
Girifalco
22.5d3          ! C12 [eV*A^12] Lenard-Jones C12
15.4d0          ! C6 [eV*A^6] Lenard-Jones C6
5.0d0           ! dm [A] radius where to switch to polinomial
6.0d0           ! d_cut [A] cut-off radius
-0.8253440000d-3        ! a, fitting polinomial coefficients: a*x^3+b*x^2+c*x+d
0.1313740800d-1         ! b
-0.6851174400d-1        ! c
0.1163980800d0          ! d
3.4d0           ! dsm [A] radius where to switch to polinomial at small distances
2.5d0           ! ds_cut [A] cut-off radius on small distances
0.1286478847d0          ! as, fitting polinomial coefficients: a*x^3+b*x^2+c*x+d
-1.858707955d0          ! bs
10.66718892d0           ! cs
-30.40360058d0          ! ds
43.05091787d0           ! es
-24.23710839d0          ! fs
```

Figure VIII.15 File C_C_TB_vdW.txt for C_{60}.

c) [*A*]_[*A*]_TB_Coulomb.txt

This file contains parameters of the softly-cut Coulomb potential, softly cut according to [34]. It is only used in the case we have electron emission included in the simulation (e.g. modeling thin films). An example of the file is shown in Figure VIII.16.

If the file is absent, the calculations will proceed without Coulomb forces. So, the Coulomb module will not affect the materials for which there is no Coulomb force or parameterization.

```
Coulomb_cut
20.0d0  ! dm
0.1d0   ! dd
```

Figure VIII.16 File C_C_TB_Coulomb.txt for C_{60}.

d) [*A*]_[*A*]_TB_wall.txt

This file contains parameters of the exponential repulsive potential at very short distances between the atoms. It is to be used in case we have kinetic energies of atoms higher than the TB-provided barrier at short distances. In such a case, it is necessary to make the system stable at short distances. The following shape of the potential is assumed, E_{EW}:

$$E_{EW} = \frac{1}{2}\sum_{i,j}^{N_{at}} C \exp\left(\frac{1}{r - r0}\right) f(r)$$

Where the summations by *i,j* runs through all the pairs of atoms N_{at}, C and r_0 are parameters, and smooth cut-off function $f(r)$ is Fermi-like function:

$$f(r) = \frac{1}{1 - \exp\left(\frac{r - d0}{dd}\right)}$$

An example of the file containing the parameters in the same notations is shown in Figure VIII.17.
If the file is absent or anything else but “Simple_wall” is specified in the first line, the calculations will proceed without exponential repulsive forces.

```
Simple_wall
1.0d2        ! C [eV]
0.0d0        ! r0 [A]
1.165d0          ! d0 [A]
0.01d0           ! dd [A]
```

Figure VIII.17 C_C_TB_wall.txt example for carbon

e) Unit_cell_equilibrium.txt

File contains the initial vectors of the unit cell in angstroms [A], to be later evolved according to the Parrinello-Rahman method [27] (for P=const simulation). The file contains three vectors (as columns), see Figure VIII.18.

Unit_cell_equilibrium.txt - Notepa

File Edit Format View Help

```
5.431d0 0.0d0   0.0d0
0.0d0   5.431d0 0.0d0
0.0d0   0.0d0   5.431d0
```

Figure VIII.18 File Unit_cell_equilibrium.txt for silicon

f) Unit_cell_atom_relative_coordinates.txt

This file contains the coordinates of atoms inside of the equilibrium unit-cell in relative coordinates. Number of lines in this file defines how many atoms we have in the unit cell of the material. For example, it is 8 atoms for GaAs; see Figure VIII.19. These 8 lines contain the following information:

First number stands for the kind of atom according to its chemical formula given in the input file. For example, for GaAs number 1 stands for Ga, number 2 stands for As. *Make sure your order of elements in the chemical formula in the file matches the order of elements in this file!*

Next 3 numbers in each line represent initial relative coordinates S_x, S_y, S_z of 8 atoms inside the unit cell (normalized from 0 to 1).

Unit_cell_atom_relative_coordinates.txt

```
1   0.0e0    0.0e0    0.0e0
1   0.5e0    0.5e0    0.0e0
1   0.5e0    0.0e0    0.5e0
1   0.0e0    0.5e0    0.5e0
2   0.25e0   0.25e0   0.25e0
2   0.75e0   0.75e0   0.25e0
2   0.75e0   0.25e0   0.75e0
2   0.25e0   0.75e0   0.75e0
```

Figure VIII.19 File Unit_cell_atom_relative_coordinates.txt for GaAs.

g) Files SAVE_supercell.dat and SAVE_atoms.dat

An alternative way to set initial configuration of the atoms and supercell is to have these files in the directory. If these files are present, the program will use them instead of the 'Unit_cell_atom_relative_coordinates.txt' and 'Unit_cell_equilibrium.txt' files described above. With these files you can add any desired atomic configuration, not only perfect periodic crystalline lattice.

The file 'SAVE_atoms.dat' <u>must</u> contain the data in the same format as the file 'OUTPUT_coordinates_and_velosities.dat' (see below).

The format of the file 'SAVE_supercell.dat' must coincide with the format of the file 'OUTPUT_supercell.dat' (see below).

These files are, e.g., necessary for simulation of amorphous materials. They must be constructed separately as follows:

Creation of initial configuration of an amorphous material

1) Choose a material you'd like to construct (e.g. carbon or silicon based)
2) Choose a number of atoms in the simulation box, which will be used for all further simulation of the amorphous state (each number of atoms chosen requires separate preparation of the initial state!)
3) Set proper density of the desirable amorphous material by adjusting the volume in the file 'Unit_cell_equilibrium.txt' of the material we will start from to melt
4) Set parameters for quenching in the NUMERICAL_PARAMETERS input file as follows:
   1   1000.0   10.0
   The first parameter tells to include material cooling (quenching), which is made by setting atomic velocities to zero starting from the time, given in the second number (e.g. 1000.0 fs, after the melting), then propagates the atomic trajectories and repeat the procedure every (~10.0) femtoseconds (e.g. for silicon-based material, chose here ~30.0 as an optimal time). Set initial conditions that would create a melted state (e.g., by setting high atomic temperature or irradiation with a high deposited dose). Run this simulation for a few picoseconds, until the total energy stops dropping. This means the material is relaxed into its equilibrium amorphous state.
   Don't forget to switch this off (by setting first parameter to 0) for further real simulations!
5) The files 'SAVE_atoms.dat' and 'SAVE_supercell.dat' are created during the simulation run in the output folder, and are updated at each saving time step; thus, the data in them after the simulation has finished correspond to the last time-step of the simulation, and can be just copied into the input file, provided the simulation delivered desired quality of results. Place these files into the folder with the new material name, see next step.
6) Place both files into the folder 'Amorphous_[material_name]'. Copy all other input files from the directory of 'parental' material (ideal material that you just melted) into the same folder, and set the name of the folder the same as the material name in the file 'INPUT_FILE.txt'.
7) Place the files 'SAVE_atoms.dat' and 'SAVE_supercell.dat' into the same directory.
8) Check if there is no artificial void in the new created state, and the density is uniform (and any other properties that are needed to be reproduced well in your amorphous material). If the amorphous material looks good, these files with the relaxed amorphous atomic state can now be

used for further simulations of amorphous material. If not, repeat the procedure from the beginning until the quenched state produced satisfy your conditions.

h) Files PHASE_[*i*]_atoms.dat and PHASE_[*i*]_supercell.dat

The user can set initial and final states of the simulation cell for calculation of the free-electron along a reaction coordinate path. Here [*i*] runs from 1 to 2, the index of the initial and the final phase. It can be done in the following way:

Calculation of free-energy along reaction coordinate path

Set initial atomic configuration in the files PHASE_1_atoms.dat and PHASE_1_supercell.dat

Set final atomic configuration in the files PHASE_2_atoms.dat and PHASE_2_supercell.dat

Run XTANT. If these files are present in the folder with the material data, XTANT will linearly interpolate coordinates from the first to the second phase (accounting for periodic boundaries) and save all the output data along this coordinate path. The free energy will be calculated for the electronic temperature provided in the INPUT_MATERIAL.file (see above).

i) File [*Material*].cdf

This file contains all parameters needed for Monte Carlo calculations of the electron cross sections within the Ritchie complex dielectric function formalism [6]. The file contains the following lines, see Figure VIII.20. Only used if the cdf-cross sections are chosen, not used for BEB cross sections.

```
Silicon.cdf
 1  Silicon
 2  Si
 3  2.3290   8430.0   ! density [g/cm^3], speed of sound [m/s]
 4  4                 ! number of shells of the first element: Si
 5  1   1   1838.9e0    2.0 1.66e0       ! number of CDF , shell-designator, ionization potential, num
 6  1579.84393e0    236.1525e0  1192.8108e0 ! E0, A, Gamma coefficients
 7  1   3   148.7e0 2.0 0.375e0          ! number of CDF functions, shell-designator, ionization poten
 8  219.31853e0 223.14459e0     199.22160e0 ! E0, A, Gamma coefficients
 9  1   4   99.2e0  6.0 16.06e0          ! number of CDF functions, shell-designator, ionization poten
10  100 685 145                             ! E0, A, Gamma coefficients
11  2   63  1.12    4.0 1.0e23           ! number of CDF functions, shell-designator, ionization poten
12  15.9504e0   113.883e0   3.212e0         ! E0, A, Gamma coefficients
13  17.499583e0 135.1805e0  3.0291e0        ! E0, A, Gamma coefficients
```

Figure VIII.20 Example file [*Material*].cdf (Silicon)

Line 1: material name

Line 2: chemical element name(s), according to the rules described above

Line 3: density of the material in [g/cm^3] (this number is the default value, which may be overwritten by the input file, see above) and speed of sound (not used in this version of MC)

Line 4: number of shells of the first element in the compound

Line 5: for the first shell, next five numbers in the line:

1) Number of complex dielectric function (CDF) oscillators used in the formalism [7]

2) Shell designator according to the EADL[18] database (see a copy in Table VI below), with additional notation of shell number 63 corresponding to the valence band.
3) Ionization potential of this shell in [eV]
4) Number of electrons in this shell
5) Time of Auger-decay in [fs]; for the valence band set here a huge number such as 1.0d23.
   The numbers must be separated by TAB, not SPACE.

Table VI. Atomic Subshell Designators

Designator	Subshell		Designator	Subshell		Designator	Subshell	
1.	K	(1s1/2)	21.	N4	(4d3/2)	41.	P1	(6s1/2)
2.	L	(2)	22.	N5	(4d5/2)	42.	P23	(6p)
3.	L1	(2s1/2)	23.	N67	(4f)	43.	P2	(6p1/2)
4.	L23	(2p)	24.	N6	(4f5/2)	44.	P3	(6p3/2)
5.	L2	(2p1/2)	25.	N7	(4f7/2)	45.	P45	(6d)
6.	L3	(2p3/2)	26.	O	(5)	46.	P4	(6d3/2)
7.	M	(3)	27.	O1	(5s1/2)	47.	P5	(6d5/2)
8.	M1	(3s1/2)	28.	O23	(5p)	48.	P67	(6f)
9.	M23	(3p)	29.	O2	(5p1/2)	49.	P6	(6f5/2)
10.	M2	(3p1/2)	30.	O3	(5p3/2)	50.	P7	(6f7/2)
11.	M3	(3p3/2)	31.	O45	(5d)	51.	P89	(6g)
12.	M45	(3d)	32.	O4	(5d3/2)	52.	P8	(6g7/2)
13.	M4	(3d3/2)	33.	O5	(5d5/2)	53.	P9	(6g9/2)
14.	M5	(3d5/2)	34.	O67	(5f)	54.	P1011	(6h)
15.	N	(4)	35.	O6	(5f5/2)	55.	P10	(6h9/2)
16.	N1	(4s1/2)	36.	O7	(5f7/2)	56.	P11	(6h11/2)
17.	N23	(4p)	37.	O89	(5g)	57.	Q	(7)
18.	N2	(4p1/2)	38.	O8	(5g7/2)	58.	Q1	(7s1/2)
19.	N3	(4p3/2)	39.	O9	(5g9/2)	59.	Q23	(7p)
20.	N45	(4d)	40.	P	(6)	60.	Q2	(7p1/2)
						61.	Q3	(7p3/2)

Number of next lines depends on the number of CDF-oscillators specified. For each oscillator there will be a separate line, containing the following E_0, A, Γ coefficients of the CDF [7]. The numbers must be separated by TAB, not SPACE.

Then, for each shell, there will be the same set of lines with its own parameters.

If there are more than one element in the compound (e.g. GaAs), the same lines from 5 and further must be present for the second element. However, the last orbital (energy level) must be skipped, because it forms the valence band, and the valence band is already described in the first element. Thus, for all the next elements the number of shells must be one less than for the case of an isolated atom.

In case if one uses BEB cross sections (by setting EADL option in the input file NUMERICAL_PARAMETERS.txt), the file [*Material*].cdf is not necessary. And vice versa, if you do not have cdf data and the corresponding file, switch to the EADL option for the cross sections.

j) Files with electron mean free paths

[*Matter*]_Total_Electron_IMFP.txt contains the total electron mean free path in the material. First column is the electron energy in [eV], second one is the inverse mean free path in [1/Å].

[18] See details of this library here: https://www-nds.iaea.org/epdl97/libsall.htm, physical details and references for the library are here: https://www-nds.iaea.org/epdl97/document/epdl97.pdf

Files named as [*A*]_[*CS*]_Electron_IMFP_Ip=[*Ip*]eV.txt specify in the same format electron mean free paths for each shell of each kind of atom.

Here [*A*] represents the atomic species (Si, C, Ga, As…); [*CS*] represents within which formalism the cross section (and, correspondingly, mean free path) is calculated: CDF or BEB; [*Ip*] is the ionization potential of the shell, the given value of *Ip* must coincide with the ionization potential specified in the cdf-file.

If these files are not present in the folder, at the first XTANT run it will automatically calculate them for the given choice of the cross section (CDF or BEB), and save. Next time, it will read from the saved files, instead of recalculating them again. Which means, if you modify something in the atomic parameters or the cdf, you have to delete the mean free paths files and let the program recalculate the new ones at the next run.

At the first run with CDF, the code also informs you about the corresponding sum-rules for the cdf you provided [23].

k) Files with photon attenuation lengths

The files are constructed in exactly the same way as the files for electrons described above, named similarly: [*Matter*]_Total_photon_IMFP.txt. In case of given cdf, the photon attenuation lengths (mean free paths) are calculated from the cdf; in case of chosen EADL (or BEB) cross sections, the photoabsorption cross sections are extracted from the EADL (part of EPICS2017) database.

l) K-points grid

File k_grid.dat may contain a number of lines with 3 values in each specifying the grid points (k_x, k_y, k_z) in the reciprocal space for calculations of the CDF and/or DOS (if the corresponding options are set in the input file; otherwise, this file is ignored). See an example in Figure VIII.21.

If no such a file present, the Monkhorst-Pack[22] sampling of k-space points is used.

```
0.0d0   0.0d0   0.0d0
0.0d0   0.0d0   0.1d0
0.0d0   0.0d0   0.2d0
0.0d0   0.0d0   0.3d0
0.0d0   0.0d0   0.5d0
0.1d0   0.0d0   0.5d0
0.2d0   0.0d0   0.5d0
0.3d0   0.0d0   0.5d0
0.4d0   0.0d0   0.5d0
0.5d0   0.0d0   0.5d0
```

Figure VIII.21. Example of k_grid.dat file

IX. Output Files

XTANT produces a number of output files.

1) OUTPUT_Error_log.dat

file must be empty if there was no errors during the execution of the code. In this case, it is automatically deleted after the execution is finished. If it's not empty and not deleted at the end, have a look inside for the description of a known error that you would have to find later in the code and figure out why it occurred. Known types of errors and their meaning:

- Error #1: file not found

- Error #2: file could not be opened
- Error #3: file could not be read on the line number given
- Error #4: some problem with databases (EADL, EPDL, periodic table file)
- Error #5: inconsistent TB parameterization (only the same type of parameterization is allowed for all kinds of atoms within compound)
- Error #6: diagonalization subroutine with LAPACK failed (uses MKL library)
- Error #7: some errors in low-energy electrons (probably in temperature or chemical potential calculation)
- Error #8: error in optical coefficients (probably in complex Hamiltonian)

2) OUTPUT_Energy.dat

In case if you included additional option -size, the code produces this file with the following information:

Column 1 is the nearest neighbour distance in [A]

Column 2 is the total energy [eV/atom]

Column 3 is the repulsive part of the energy [eV/atom]

Column 4 is the attractive part of the energy [eV/atom]

3) Directory OUTPUT_[*material*]_hw=[*hw*]_t=[*t*]_F=[*F*]

Such a directory contains all output files with results of the code execution. Its name itself contains details of the parameters of the run:

[*material*] is the name of the material used (diamond, silicon, etc.)

[*hw*] is the photon energy of the FEL pulse used [eV]

[*t*] is the duration of the FEL pulse [fs]

[*F*] is the pulse fluence in terms of the absorbed dose in [eV/atom]

For the case of more than one FEL-pulse modelled, these parameters are shown for the *first* pulse, but the directory-name is appended with the following:

OUTPUT_[*material*]_hw=[*hw*]_t=[*t*]_F=[*F*]_[*N*]_pulses

Where *N* shows the number of pulses specified in the input file.

Alternatively, for no pulse calculations (*F*=0), the name will be
OUTPUT_[*material*]_Te=[*Te*]_Ta=[*Ta*]_[*coupling*] where

[*Te*] is the initial electron temperature [K]

[*Ta*] is the initial atomic temperature [K]

[*coupling*] will be either "no_coupling" (if no electron-phonon coupling is included), or "with_coupling" if the nonadiabatic coupling is switched on.

If a run with the same parameters already was performed, and the data file with the same name already exists, the new file will be created with a number at the end, e.g.

OUTPUT_[*material*]_hw=[*hw*]_t=[*t*]_F=[*F*]_v1

An output directory will also contain a number of files, including a copy of the INPUT_MATERIAL.txt and NUMERICAL_PARAMETERS.txt for your records, and the file

!OUTPUT_[*material*]_Parameters.txt with essentially the same information, plus the atomic data that are extracted from either cdf-file, or EADL database, and duration of execution of the program. Also, if you use communication file (see the next subsection), its results will be saved here for your information.

Files SAVE_[something].dat can be ignored, they are supposed to save the transient data at each simulation step, but the option is not finished. Use them only for creating amorphous state as described above.

4) Communication with the program on-the-fly

In the output folder, XTANT creates a text-file named Comunication.txt. This file is checked by the program at each saving-time-step. You can send the following messages to the program that it will interpret and act upon:

- time "number" : to change total duration of the simulation (type 'time' and the new number in [fs], without quotation marks, e.g. time 10000, or Time 2e3)
- SAVEdt "number" : to change how often outputs are saved (type new number in [fs], e.g. SAVEdt 2.0 – will make the program to save output data with the time step of 2 fs)
- MDdt "number" : to change the timestep of MD simulation (type new number in [fs], e.g. MDdt 0.01).
- OMP "number" : to change the number of OpenMP threads in the parallel calculations (integer). Setting here zero or negative number will set the number of threads equal to the maximal number of threads on your machine.

At the end of simulation this file is deleted.

5) Plotting: OUTPUT_Gnuplot_all.sh

Execute this file to create all the plots of all results of calculations. You can do that even if XTANT is still running, then it will give you transient results. At the end of the simulation run, this command will also be executed automatically.

This is a gnuplot shell script that is created by XTANT to execute all other gnuplot shell scripts in the folder that are plotting all the essential quantities:

OUTPUT_bands_Gnuplot.sh – plots the bottom of the valence band (VB), top of the VB, bottom of the conduction band (CB) and top of the CB, and chemical potential.

OUTPUT_CB_electron_Gnuplot.sh – plots the density of conduction band electrons.

OUTPUT_coupling_parameter_Gnuplot.sh – plots the electron-phonon coupling parameter.

OUTPUT_deep_shell_holes_Gnuplot.sh – plots the density of deep shell holes in each shell of each atom of the compound.

OUTPUT_electron_Ce.sh – plots the heat capacity of electrons.

OUTPUT_electron_distribution_Gnuplot.sh – plots the electron distribution function (overlap of all timesteps, not very useful). Takes a few minutes to plot. Better not to use, but set your own timesteps to print out if needed.

OUTPUT_electrons_and_holes_Gnuplot.sh – plots the high-energy electrons and core holes densities.

OUTPUT_energies_Gnuplot.sh – plots the total, potential, and atomic energies.

OUTPUT_energy_levels_Gnuplot.sh – plots the electron energy levels (eigenvalues of the TB hamiltonian). Takes a few minutes to plot.

OUTPUT_mean_displacement_Gnuplot.sh – plots the (mean atomic displacements)^*N* with respect to the initial positions.

OUTPUT_mu_and_Egap.sh – plots the electron chemical potential and band gap.

OUTPUT_Mulliken_charges_Gnuplot.sh – plots charges of different types of atoms.

OUTPUT_optical_coefficients.sh – plots optical R, T, A for specified probe pulse wavelength.

OUTPUT_optical_n_and_k.sh – plots corresponding real and imaginary parts of the refractive index.

OUTPUT_pressure_Gnuplot.sh – plots total pressure in atomic system the simulation box.

OUTPUT_stress_tensor_Gnuplot.sh – plots components of the atomic pressure tensor.

OUTPUT_temperatures_Gnuplot.sh – plots the electron and atomic temperatures.

OUTPUT_volume_Gnuplot.sh – plots the volume of the supercell.

In case if you set a probe-pulse to be included, additional gnuplot files of the convolved data will be created (see below), that will be named exactly the same way with the word 'CONVOLVED' added at the end, e.g. convolved electron heat capacity would be in a file named

OUTPUT_electron_Ce_CONVOLVED.sh

Note that in case of Windows operating system, instead of shell-scripts the program will create cmd batch files (with the same name, just different extension: .cmd instead of .sh). They will need windows version of gnuplot installed[19], and proper paths written in the environment variables[20].

6) Output data files

OUTPUT_atomic_coordinates.xyz – contains the atomic positions at each time-step in [A] (saving time-step specified in the input-file, not the numerical time-step used in the MD) in the XYZ format.

OUTPUT_atomic_coordinates.cif – contains the atomic positions at each time-step in [A] (saving time-step specified in the input-file, not the numerical time-step used in the MD) in the CIF format.

---

[19] http://www.gnuplot.info/

[20] This, and many other useful things, can be done with help of Cygwin: https://www.cygwin.com/

OUTPUT_coordinates_and_velosities.dat – contains the atomic coordinates and velocities for all atoms at each timestep. First thee values and the coordinates in [A], last three are the velocity in each line in [A/fs]. A line describes one atom in the super cell. After all atoms' data for one timestep there are two empty lines. After that, the next timestep is starting. Use it for quick look with gnuplot, e.g.:

sp "OUTPUT_coordinates_and_velosities.dat" i 201 u 1:2:3 pt 6 ps 3

for step number 201, coordinates (columns 1:2:3).

Also use it for calculations of atomic velocity autocorrelators and phonon spectra (see below).

OUTPUT_coupling.dat – electron-ion coupling parameter. Contains the following lines:

1) Time [fs]
2) Total coupling parameter [W/m^{3}K]
3) and further: Partial coupling parameter for each type of pair atoms [A]-[B] in the compound. For example, for elemental Al targets there will be one column Al-Al. For compound AlCu, there will be 4 columns: Al-Al, Al-Cu, Cu-Al, Cu-Cu. Etc.
4) Next columns will be partial coupling for each type of orbitals (defined by the basis set) for each kind of atoms in the compound. For example, for sp^{3}d^5 basis set, there will be couplings for the pairs of levels: s-s, s-p, s-d, p-s, p-p, p-d, d-s, d-p, d-d (for each kind of elements [A]-[B] in the compound).

OUTPUT_deep_shell_holes.dat – contains timestep [fs], number of holes in each shell of each kind of atoms in the compound in the next columns (normalized to the number of atoms).

OUTPUT_DOS.dat – contains blocks of data separated by two empty lines. Each block contains a few columns:

1) energy in [eV]
2) total DOS in [states/eV]
3) partial DOS (PDOS) for the first atomic shell of the first element in the compound [states/eV]
4) PDOS for the next shell of the first element, etc.

PDOS corresponding to atomic orbital contributions into DOS will be printed out only if the user set the PDOS parameter to 1 (see input file NUMERICAL_PARAMETERS above). The number of columns will depend on the number of elements in the compound and basis set used. E.g., for one element with sp^{3}d^5 basis set, there will be three columns with PDOS, corresponding to s, p and d PDOS. For N_{elem} elements in the compound and sp^{3}d^5 basis set, there will be 3xN_{elem} columns; for sp^3 basis set, there will be 2 columns per element: with s and p PDOS, etc.

The file is created only if printing out DOS is set by the user.

OUTPUT_dielectric_function.dat – contains blocks of data separated by two empty lines. Each block contains 16 columns:

1) energy in [eV]
2) real part of CDF
3) imaginary part of CDF
4) loss function

5) reflectivity
6) transmission
7) absorption
8) optical n (real part of the refraction coefficient)
9) optical k (imaginary part of the refraction coefficient)
10) dc-conductivity
11) Real part of the (x,x) component of the CDF
12) Imaginary part of the (x,x) component of the CDF
13) Real part of the (y,y) component of the CDF
14) Imaginary part of the (y,y) component of the CDF
15) Real part of the (z,z) component of the CDF
16) Imaginary part of the (z,z) component of the CDF

File created only if printing out spectrum of optical coefficients is set by the user. From this file, parameters for a chosen photon energy may be extracted in post-processing (see below).

OUTPUT_electron_distribution.dat – contains electron distribution function on the current energy levels, for each timestep separated by the double empty line.

OUTPUT_electron_hole_numbers.dat – contains the following data:

1) Time [fs];
2) Number of valence band electrons;
3) Number of conduction band electrons;
4) Number of high-energy electrons;
5) Total number of core holes (in all shells summed up);
6) Error in the particle conservation;
7) Number of photons as sampled.

Data are normalized per the number of atoms.

OUTPUT_electron_properties.dat – contains the following data:

1) Time [fs]
2) Number of electrons [%, per atom]
3) Chemical potential [eV]
4) Band gap [eV]
5) Electrons heat capacity [J/m^{3}K]
6) Electron-phonon coupling parameter [W/m^{3}K]
7) Bottom of the VB [eV]
8) Top of the VB [eV]
9) Bottom of the CB [eV]
10) Top of the CB [eV]
11) Mulliken charges for all types of atoms in the modelled compound [electron charge]

OUTPUT_energies.dat – contains the following data:

1) Time [fs]
2) Energy of electrons [eV/atom]

3) Energy of all core holes [eV/atom]
4) Potential energy of atoms [eV/atom]
5) Kinetic energy of atoms [eV/atom]
6) Total energy of atoms [eV/atom]
7) Total energy of atoms and electrons [eV/atom]
8) Total energy in the system (atoms, electrons, holes) [eV/atom] – should be always conserved, except during an FEL pulse
9) Van der Waals energy (if included in the simulation) [eV/atom]

OUTPUT_energy_levels.dat – contains all the eigenstates of the Hamiltonian, at each timestep, separated by two empty lines, in [eV].

OUTPUT_optical_coefficients.dat – Optical coefficients for the given probe photon energy, printed in the same format as 16 columns in the file OUTPUT_dielectric_function.dat, except the first column in time in [fs]. The file is created only if probe pulse is set by the user.

OUTPUT_pressure_and_stress.dat – Contains:

1) Time [fs]
2) Pressure [GPa]
3) 9 columns with the components of the pressure tensor Pressure(a,b), with a=x,y,z and b=x,y,z, all in [GPa]

OUTPUT_supercell.dat – contains the data for the supercell: time, volume [A^3], 9 super-cell vectors [A], and their 9 velocities [A/fs].

OUTPUT_temperatures.dat – contains the following data:

1) Time [fs]
2) Electron temperature [K]
3) Atomic temperature [K]: first column is the average, then for each element of the compound
4) (Mean atomic displacement)^N [A^N]: first column is the average, then for each element of the compound

7) …_CONVOLVED.dat output data files

If in the input file you set a finite positive duration of the probe pulse, all the data from the output files mentioned above will be additionally convolved with a Gaussian function of a given FWHM. The resulting data will be saved in the new output data files under the same names with the tag 'CONVOLVED' added to them, e.g. temperatures after the convolution will be in the file:

OUTPUT_temperatures_CONVOLVED.dat

These files will be used by gnuplot to prepare convolved figures.

X. Post-processing

XTANT package contains a few programs for post-processing of the output files, if required. The following post-processing is possible:

1) Extracting pair correlation function

If the user did not set to printout the pair correlation function (PCF) in the input files, it can be extracted in the post-processing using the XTANT_atomic_data_analysis.exe, which must be placed into the folder with the output data.

This program requires the following output data to be present:

'OUTPUT_coordinates_and_velosities.dat'

'OUTPUT_supercell.dat'

'OUTPUT_energies.dat'

'OUTPUT_temperatures.dat'

And can be run as follows: XTANT_atomic_data_analysis.exe in the command line. It will construct and printout the pair correlation function at each time instant. It creates the output file OUTPUT_PCF.dat, where the PCF saved in the same format as OUTPUT_pair_correlation_function.dat described above.

Note, however, that PCF can easily be obtained by standard molecular dynamics visualization software, such as VMD[21], OVITO[22], etc.

2) Calculation of velocity autocorrelation and phonon spectra

A program for calculation of atomic velocity autocorrelation functions and phonon spectra XTANT_autocorrelators.exe must be placed into the folder with the output data. To execute, simply run XTANT_autocorrelators.exe in the command line. The program will ask the user to input two parameters: alpha and time step. Alpha is the exponential factor suppressing correlation at long times *exp*(-alpha*t). The time step *tim_step* sets how often to print out autocorrelators and phonon spectra: it will divide the data into *tim_step* steps, and calculated data on them. E.g. if you had a simulation run from 0 to 1000 fs, and set *tim_step*=10, it will print 10 files, at each 200 fs.

The program will create a set of files

OUT_VAF_[*time*].dat

OUT_vibrational_spectrum_[*time*].dat

Where [*time*] is a timestamp at which the velocity autocorrelation function (VAF) and phonon spectrum are calculated. The program uses Fourier transform to get spectrum from VAF [1].

The files OUT_VAF_[*time*].dat contain two columns:

1) Time [fs]
2) VAF [arb. units]

The files OUT_vibrational_spectrum_[*time*].dat contain two columns:

1) Phonon frequency [THz]
2) Phonon spectrum [arb. units]

[21] https://www.ks.uiuc.edu/Research/vmd/

[22] https://www.ovito.org/

3) Calculation of electron-ion coupling parameter $g(T_e)$, $C_e(T_e)$ and $\mu(T_e)$

To calculate electron-ion coupling parameter as a function of electron temperature $g(T_e)$, electronic heat capacity $C_e(T_e)$ and chemical potential $\mu(T_e)$, the following procedure can be used [37]:

- Set a simulation run with two pulses: first one with the fluence of ~3-4 eV/atom, duration of ~30 fs, and low photon energy of ~10 eV, the position of the maximum at 0 fs. The second one with the fluence of ~1.0e-6, duration of 10 fs, and photon energy of 10 eV, and the position of the maximum at ~ -400 fs. The second pulse serves as an additional time for the system to thermalize prior to the actual pulse arrival.
- Create a few additional input files with incrementally increasing numbers to run a few simulation in a sequence (see above how to do that, Page 20). In each of these files, vary slightly the parameters of the first pulse, e.g. chose different fluences between 3 and 4 eV/atom, chose different pulse durations between 10-60 fs, initial electronic temperature between 100 and 300 K, and slightly different arrival times of the second pulse between -300 fs to -500 fs. This will randomize the parameters and exclude artificial correlation effects [37]. Recommended number of thusly-created simulation runs: 10 or more.
- Run XTANT to create 10 (or more) output folders with the slightly different parameters as said above. Place all output folders into a separate directory, and place the file XTANT_coupling_parameter.exe in the same directory.
- Execute it as follows: XTANT_coupling_parameter.exe alpha

where "alpha" is an optional argument meaning the time [fs] from which to start averaging the coupling parameter (used to exclude early times where atoms are not equilibrated yet). For example, set alpha=-150 fs, i.e. run XTANT_coupling_parameter.exe -150, which will exclude earlier times during which the system was still equilibrating.

This program will scan through all the output folders (note that no other folders or files in this directory should start with the word 'OUTPUT', as the program will use them and crush), and use files OUTPUT_electron_properties.dat, OUTPUT_temperatures.dat, OUTPUT_coupling.dat (if exists) and OUTPUT_pressure_and_stress.dat to extract g, C_e, μ and Gruneisen parameter as functions of time, then sort them according to the electronic temperatures (T_e) at those time, interpolate on a grid of T_e, average over all 10 (or more) simulation runs, and print out averaged values into the following files:

OUT_average_coupling.dat

Which contains 3 columns:

1) Electron temperature [K]
2) Total coupling parameter [W/(m^{3}K)]
3) Standard deviation of the coupling parameter (error bars)

OUT_average_parameters.dat

Which contains 4 columns:

1) Electron temperature [K]
2) Chemical potential [eV]

3) Electron heat capacity [J/(m^{3}K)]
4) Standard deviation of the electron heat capacity (error bars)

OUT_average_partial_couplings.dat

Which contains a number of columns corresponding to the number of partial couplings for each pairs of elements and shells (according to the file OUTPUT_coupling.dat in the same format).

OUT_average_pressure.dat

Which contains 4 columns:

1) Electronic temperature Te in [K]
2) Electron energy in [eV/atom]
3) Pressure in [GPa]
4) Electronic Gruneisen parameter, defined as dP/dE in [Pa/(J/atom)]

Note that an alternative definition of the electronic Gruneisen parameter is P/E (instead of derivative) [41], which can be calculated from the columns 2 and 3 in this file. In this case, it is important to subtract the room temperature value from the pressure (since it is rarely exactly zero in the simulation). This definition is useful, since the definition based on the derivative often produces too noisy results.

4) Extracting optical parameters for given wavelength from the spectrum

If you run XTANT simulation with printing out optical spectrum, you can extract optical parameters as function of time for any photon energy (wavelength) within the spectrum interval, for *s* and *p* polarizations, and given angle of probe incidence and substrate.
To run, place the program XTANT_dielectric_function_analysis.exe into the output folder with the results which must contain the file OUTPUT_dielectric_function.dat, and simply call it as: XTANT_dielectric_function_analysis.exe *hw*, where *hw* is the probe photon energy in [eV].
It also requires the following additional input file to be present: OPTICAL_PARAMETERS.dat (place it manually into the same directory with the output data).

```
50.0d0       ! thickness [nm]
90.0d0       ! with respect to surface [deg]
1.0d0   0.0d0           ! n and k of the material above the target
3.6941d0    0.0065435d0 ! n and k of the material below the target
```

Figure X.1 Example of input file for optical parameters extraction.

The lines in this file must be as follows:

1) Thickness of the target in [nm]
2) Probe incidence angle with respect to the surface [degree]
3) Optical *n* and *k* of the material above the target (typically, air)
4) Optical *n* and *k* of the substrate (substrate material below the target, may also be air)

The program will create 2 output files:

1) OUTPUT_dielectric_function_[*hw*].dat with the same format as the file OUTPUT_optical_coefficients.dat described above (just by interpolating the data from the file with the dielectric function printed out). The columns are marked in the first line of the file, as follows: time hw Re_eps Im_eps LF R T A n k. The optical coefficients are calculated for s-polarization only. *Note that it might not work well, and the file may contain just zeros!*
2) OUTPUT_dielectric_function_[*hw*]_RTA.dat with recalculated optical parameters taking into account properties of materials above and below the target, following the formalism described in [42]. The columns are marked in the first line of the file, as follows:

1. time [fs]
2. R_s – the first ray reflection for s-polarization
3. T_s – the first ray transmission for s-polarization
4. A_s – the first ray absorption for s-polarization
5. R_p – the first ray reflection for p-polarization
6. T_p – the first ray transmission for p-polarization
7. A_p – the first ray absorption for p-polarization
8. R_s_all – coherently summed all rays reflection for s-polarization
9. T_s_all – coherently summed all rays transmission for s-polarization
10. A_s_all – coherently summed all rays absorption for s-polarization
11. R_p_all – coherently summed all rays reflection for p-polarization
12. T_p_all – coherently summed all rays transmission for p-polarization
13. A_p_all – coherently summed all rays absorption for p-polarization

XI. References

[1] N. Medvedev, V. Tkachenko, V. Lipp, Z. Li, B. Ziaja, Various damage mechanisms in carbon and silicon materials under femtosecond x-ray irradiation, 4open. 1 (2018) 3. https://doi.org/10.1051/fopen/2018003.

[2] K.F. Garrity, K. Choudhary, Fast and Accurate Prediction of Material Properties with Three-Body Tight-Binding Model for the Periodic Table, (n.d.). https://pages.nist.gov/ (accessed June 3, 2022).

[3] J. Jenke, A.N. Ladines, T. Hammerschmidt, D.G. Pettifor, R. Drautz, Tight-binding bond parameters for dimers across the periodic table from density-functional theory, Phys. Rev. Mater. 5 (2021) 23801. https://doi.org/10.1103/PhysRevMaterials.5.023801.

[4] D. Porezag, T. Frauenheim, T. Köhler, G. Seifert, R. Kaschner, Construction of tight-binding-like potentials on the basis of density-functional theory: Application to carbon, Phys. Rev. B. 51 (1995) 12947–12957. https://doi.org/10.1103/PhysRevB.51.12947.

[5] P. Pracht, E. Caldeweyher, S. Ehlert, S. Grimme, A Robust Non-Self-Consistent Tight-Binding Quantum Chemistry Method for large Molecules, (2019). https://doi.org/10.26434/CHEMRXIV.8326202.V1.

[6] R.H. Ritchie, A. Howie, Electron excitation and the optical potential in electron microscopy, Philos. Mag. 36 (1977) 463–481. https://doi.org/10.1080/14786437708244948.

[7] N. Medvedev, Modeling ultrafast electronic processes in solids excited by femtosecond VUV-XUV laser Pulse, AIP Conf. Proc. 582 (2012) 582–592. https://doi.org/10.1063/1.4739911.

[8] Y.-K. Kim, M. Rudd, Binary-encounter-dipole model for electron-impact ionization, Phys. Rev. A. 50 (1994) 3954–3967. https://doi.org/10.1103/PhysRevA.50.3954.

[9] N. Medvedev, Femtosecond X-ray induced electron kinetics in dielectrics: application for FEL-pulse-duration monitor, Appl. Phys. B. 118 (2015) 417–429. https://doi.org/10.1007/s00340-015-6005-4.

[10] S. Hammes-Schiffer, J.C. Tully, Proton transfer in solution: Molecular dynamics with quantum transitions, J. Chem. Phys. 101 (1994) 4657. https://doi.org/10.1063/1.467455.

[11] J.C. Tully, Molecular dynamics with electronic transitions, J. Chem. Phys. 93 (1990) 1061. https://doi.org/10.1063/1.459170.

[12] F. Trani, G. Cantele, D. Ninno, G. Iadonisi, Tight-binding calculation of the optical absorption cross section of spherical and ellipsoidal silicon nanocrystals, Phys. Rev. B. 72 (2005) 075423. https://doi.org/10.1103/PhysRevB.72.075423.

[13] M. Harmand, R. Coffee, M. Bionta, M. Chollet, D. French, D.M. Zhu, D.T. Fritz, H. Lemke, N. Medvedev, B. Ziaja, S. Toleikis, M. Cammarata, Achieving few-femtosecond time-sorting at hard X-ray free-electron lasers, Nat Phot. 7 (2013) 215–218. https://doi.org/10.1038/nphoton.2013.11.

[14] C.-C. Fu, M. Weissmann, Tight-binding molecular-dynamics study of amorphous carbon deposits over silicon surfaces, Phys. Rev. B. 60 (1999) 2762–2770. https://doi.org/10.1103/PhysRevB.60.2762.

[15] J.C. Slater, G.F. Koster, Simplified LCAO Method for the Periodic Potential Problem, Phys. Rev. 94 (1954) 1498–1524. https://doi.org/10.1103/PhysRev.94.1498.

[16] C. Molteni, L. Colombo, L. Miglio, Tight-binding molecular dynamics in liquid III-V compounds. I. Potential generation, J. Phys. Condens. Matter. 6 (1994) 5243–5254. https://doi.org/10.1088/0953-8984/6/28/003.

[17] D.A. Papaconstantopoulos, M.J. Mehl, The Slater Koster tight-binding method: a computationally efficient and accurate approach, J. Phys. Condens. Matter. 15 (2003) R413–R440. https://doi.org/10.1088/0953-8984/15/10/201.

[18] Harald O. Jeschke, Theory for optically created nonequilibrium in covalent solids, Technical University of Berlin, 2000. http://www.physics.rutgers.edu/~jeschke/phd.html.

[19] H.J.C. Berendsen, J.P.M. Postma, W.F. van Gunsteren, A. DiNola, J.R. Haak, Molecular dynamics with coupling to an external bath, J. Chem. Phys. 81 (1984) 3684–3690. https://doi.org/10.1063/1.448118.

[20] J.F. Littmark, J.P. Ziegler, U. Biersack, The Stopping and Range of Ions in Solids, (1985) http://www.srim.org. http://www.srim.org.

[21] N. Medvedev, H.O. Jeschke, B. Ziaja, Nonthermal graphitization of diamond induced by a femtosecond x-ray laser pulse, Phys. Rev. B. 88 (2013) 224304. https://doi.org/10.1103/PhysRevB.88.224304.

[22] H.J. Monkhorst, J.D. Pack, Special points for Brillouin-zone integrations, Phys. Rev. B. 13 (1976) 5188–5192. https://doi.org/10.1103/PhysRevB.13.5188.

[23] N.A. Medvedev, R.A. Rymzhanov, A.E. Volkov, Time-resolved electron kinetics in swift heavy ion irradiated solids, J. Phys. D. Appl. Phys. 48 (2015) 355303. https://doi.org/10.1088/0022-3727/48/35/355303.

[24] L. Verlet, Computer “Experiments” on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules, Phys. Rev. 159 (1967) 98–103. https://doi.org/10.1103/PhysRev.159.98.

[25] H. Yoshida, Construction of higher order symplectic integrators, Phys. Lett. A. 150 (1990) 262–

268. https://doi.org/10.1016/0375-9601(90)90092-3.

[26] G.J. Martyna, M.E. Tuckerman, Symplectic reversible integrators: Predictor–corrector methods, J. Chem. Phys. 102 (1995) 8071. https://doi.org/10.1063/1.469006.

[27] M. Parrinello, A. Rahman, Crystal Structure and Pair Potentials: A Molecular-Dynamics Study, Phys. Rev. Lett. 45 (1980) 1196–1199. https://doi.org/10.1103/PhysRevLett.45.1196.

[28] N.A. Medvedev, H.O. Jeschke, B. Ziaja, Non-thermal phase transitions in semiconductors under femtosecond XUV irradiation, SPIE Proc. 8777 (2013) 877709-877709–10. https://doi.org/10.1117/12.2019123.

[29] C.J. Fennell, J.D. Gezelter, Is the Ewald summation still necessary? Pairwise alternatives to the accepted standard for long-range electrostatics, J. Chem. Phys. 124 (2006) 234104. https://doi.org/10.1063/1.2206581.

[30] M. Elstner, SCC-DFTB: What Is the Proper Degree of Self-Consistency?†, J. Phys. Chem. A. 111 (2007) 5614–5621. https://doi.org/10.1021/JP071338J.

[31] N. Medvedev, Z. Li, V. Tkachenko, B. Ziaja, Electron-ion coupling in semiconductors beyond Fermi's golden rule, Phys. Rev. B. 95 (2017) 014309. https://doi.org/10.1103/PhysRevB.95.014309.

[32] F. Tavella, H. Höppner, V. Tkachenko, N. Medvedev, F. Capotondi, T. Golz, Y. Kai, M. Manfredda, E. Pedersoli, M.J. Prandolini, N. Stojanovic, T. Tanikawa, U. Teubner, S. Toleikis, B. Ziaja, Soft x-ray induced femtosecond solid-to-solid phase transition, High Energy Density Phys. 24 (2017) 22. https://doi.org/10.1016/j.hedp.2017.06.001.

[33] N. Medvedev, H.O. Jeschke, B. Ziaja, Nonthermal phase transitions in semiconductors induced by a femtosecond extreme ultraviolet laser pulse, New J. Phys. 15 (2013) 015016. https://doi.org/10.1088/1367-2630/15/1/015016.

[34] M. Toufarová, V. Hájková, J. Chalupský, T. Burian, J. Vacík, V. Vorlíček, L. Vyšín, J. Gaudin, N. Medvedev, B. Ziaja, M. Nagasono, M. Yabashi, R. Sobierajski, J. Krzywinski, H. Sinn, M. Störmer, K. Koláček, K. Tiedtke, S. Toleikis, L. Juha, Contrasting behavior of covalent and molecular carbon allotropes exposed to extreme ultraviolet and soft x-ray free-electron laser radiation, Phys. Rev. B. 96 (2017). https://doi.org/10.1103/PhysRevB.96.214101.

[35] J. Frenzel, A.F. Oliveira, N. Jardillier, T. Heine, G. Seifert, Semi-relativistic, self-consistent charge Slater-Koster tables for density-functional based tight-binding (DFTB) for materials science simulations., Dresden, 2009. http://www.dftb.org/parameters/download/matsci/matsci-0-3-cc/.

[36] M.J. Mehl, D.A. Papaconstantopoulos, Applications of a tight-binding total-energy method for transition and noble metals: Elastic constants, vacancies, and surfaces of monatomic metals, Phys. Rev. B. 54 (1996) 4519–4530. https://doi.org/10.1103/PhysRevB.54.4519.

[37] N. Medvedev, I. Milov, Electron-phonon coupling in metals at high electronic temperatures, Phys. Rev. B. 102 (2020) 064302. https://doi.org/10.1103/PhysRevB.102.064302.

[38] T. DUMITRICA, R.E. ALLEN, Nonthermal transition of GaAs in ultra-intense laser radiation field, Laser Part. Beams. 20 (2002) 237–242. https://doi.org/10.1017/S026303460220213X.

[39] I. Kwon, R. Biswas, C. Wang, K. Ho, C. Soukoulis, Transferable tight-binding models for silicon, Phys. Rev. B. 49 (1994) 7242–7250. https://doi.org/10.1103/PhysRevB.49.7242.

[40] A. Carlson, An Extended Tight-Binding Approach for Modeling Supramolecular Interactions of Carbon Nanotubes, UNIVERSITY OF MINNESOTA, 2006. http://www.me.umn.edu/%7B~%7Ddtraian/tony-thesis.pdf.

[41] K.P. Migdal, D.K. Il'nitsky, Y. V Petrov, N.A. Inogamov, Equations of state, energy transport and two-temperature hydrodynamic simulations for femtosecond laser irradiated copper and gold,

J. Phys. Conf. Ser. 653 (2015) 12086. https://doi.org/10.1088/1742-6596/653/1/012086.

[42] P. Yeh, Optical Waves in Layered Media, Volume 61, Wiley, the University of California, Santa Barbara, 2005. http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0471731927.html (accessed December 15, 2015).