



TREKIS 4

Time-Resolved Kinetics in Irradiated Solids

version 29.06.2024

N. Medvedev^{a,1,2}, A.E. Volkov^{3,4}, R. Rymzhanov^{5,6}, F. Akhmetov⁷, R. Voronkov⁴

- 1) Institute of Physics, Czech Academy of Sciences, Na Slovance 2, 182 21 Prague 8, Czech Republic
- 2) Institute of Plasma Physics, Czech Academy of Sciences, Za Slovankou 3, 182 00 Prague 8, Czech Republic
- 3) National Research Centre ‘Kurchatov Institute’, Kurchatov Sq. 1, 123182 Moscow, Russia
- 4) P.N. Lebedev Physical Institute of the Russian Academy of Sciences, Leninskij pr., 53, 119991 Moscow, Russia
- 5) Joint Institute for Nuclear Research, Joliot-Curie 6, 141980, Dubna, Moscow Region, Russia
- 6) The Institute of Nuclear Physics, Ibragimov St. 1, 050032 Almaty, Kazakhstan
- 7) Industrial Focus Group XUV Optics, MESA+ Institute for Nanotechnology, University of Twente, Drienerlolaan 5, 7522 NB Enschede, The Netherlands

^a nikita.medvedev@fzu.cz

Contents

I.	Introduction	3
	Models and processes in MC:.....	3
1.	Swift heavy ion.....	3
2.	Electrons.....	3
3.	Photons	3
4.	Positrons	4
5.	Holes.....	4
6.	Geometries.....	4
7.	Experimental data to compare with.....	5
8.	Probabilities of ionization within DOS	5
9.	Electron emission	5
	Atomic motion in molecular dynamics (MD):	5
	Model for low-energy electrons:	5
II.	Structure of the code.....	5
	Some code conventions:	6
	File and module names:	6
	INPUT_DATA directory	6
1)	INPUT_DATA.txt.....	7
2)	NUMERICAL_PARAMETERS.txt.....	7
3)	Directory Atomic_parameters	7
5)	Directory Material_parameters.....	8
6)	Directory MD_input	10
7)	Directory Normalized_CDF	17
8)	Directory Electron_cross_sections	18
9)	Directory Ion_cross_sections	18
10)	Directory Photon_cross_sections	18
11)	Directory Positron_cross_sections	18
12)	Directory DSF	18
III.	Compiling the code	18
	Windows:.....	19
1.	Installation of the necessary components	19
2.	Compiling the code in Windows	19
3.	Compiling in Microsoft Visual Studio (VS)	19
4.	Dependencies and libraries.....	20
	Linux:.....	21
IV.	Input parameters: INPUT_DATA.txt.....	22
V.	Input file NUMERICAL_PARAMETERS.txt.....	24
VI.	Communication with program on-the-fly: Comunicacion.txt	31
VII.	Output data: OUTPUT	32
VIII.	References	37

I. Introduction

TREKIS 4: Time-Resolved Kinetics in Irradiated Solids

The code is work in progress, it has many unfinished options

In contrast to TREKIS-3 [1,2], TREKIS-4 can model irradiation with the following particles: photon, electrons, positron, ion. The code includes the following models:

Monte Carlo (MC), Molecular Dynamics (MD), and potentially the two-temperature model (TTM, not ready yet). Each module can be switched on or off.

A few target types can be modelled: one- or multiple-component; finite-size or bulk with periodic boundaries (various types of boundaries); the supported shapes are: parallelepiped, sphere, segment, cylinder. Using this geometries, various targets may be constructed (however, currently only parallelepiped is complete, other shapes are unfinished!).

Various geometries for data analysis and output are allowed: 3d, 2d, 1d, or 0d; cartesian, spherical, cylindrical (currently, only cartesian, and cylindrical systems in 1d and 2d are supported, the others are unfinished).

The target may consist of any chemical elements in the range 1-99 atomic number (due to limitation of the EPICS database).

Models and processes in MC:

For description of the processes included and physical models used to describe them, see Ref. [3]. Here, a brief list of them is given:

1. Swift heavy ion

- 1) Inelastic scattering: at low energies, cross section in Ritchie-Howie complex dielectric function (CDF) formalism [4], or Mermin CDF [5]; at high energies, the delta-functional CDF [6].

2. Electrons

- 1) Impact ionization, atomic approximation:
 - a) at low energies, cross section in Ritchie-Howie complex dielectric function (CDF) formalism [4], or Mermin CDF [5]; at high energies, the delta-functional CDF [6].
 - b) RBEB cross section – [7] Eq.(22) (*unfinished*)
- 2) Elastic scattering (on atomic system):
 - a) relativistic Mott's [3], p. 158, Eqs.(7.5-7.8)
 - b) at low energies, cross section in Ritchie-Howie complex dielectric function (CDF) formalism [4], or Mermin CDF [5]; at high energies, the delta-functional CDF [6].
- 3) Bremsstrahlung: PENELOPE model [8], p. 82, Eq.(4.1)

3. Photons

- 1) Photoabsorption: EPDL data from EPICS2017 [9]

- 2) Compton effect: PENELOPE [10]
- 3) Thomson / Rayleigh scattering: PENELOPE [10] Eq.(2.2)
- 4) Electron-positron pair creation: PENELOPE [10]

4. Positrons

1) Impact ionization, atomic approximation:

- a) at low energies, cross section in Ritchie-Howie complex dielectric function (CDF) formalism [4], or Mermin CDF [5]; at high energies, the delta-functional CDF [6].
- b) RBEB cross section – [7] Eq.(22) (*unfinished*)

2) Elastic scattering (atomic):

- a) relativistic Mott's [3], p. 158, Eqs.(7.5-7.8)
- b) at low energies, cross section in Ritchie-Howie complex dielectric function (CDF) formalism [4], or Mermin CDF [5]; at high energies, the delta-functional CDF [6].
- 3) Bremsstrahlung: PENELOPE model [8], p. 82, Eq.(4.1)
- 4) Annihilation into pair of photons: Heitler [11], Eq.(3.74), Eq.(3.160)

5. Holes

1) For valence-band (VB) holes, which are mobile, the transport is described using cross sections:

Elastic scattering:

- a) relativistic Mott's [3], p. 158, Eqs.(7.5-7.8)
- b) at low energies, cross section in Ritchie-Howie complex dielectric function (CDF) formalism [4], or Mermin CDF [5]; at high energies, the delta-functional CDF [6].

Inelastic scattering:

- a) at low energies, cross section in Ritchie-Howie complex dielectric function (CDF) formalism [4], or Mermin CDF [5]; at high energies, the delta-functional CDF [6]
- 2) Effective mass of a valence-band hole:
 - a) constant provided by the user,
 - b) mass evaluated from the DOS within the one-band approximation [2].
- 3) Core-shell holes may decay via two channels:
 - a) Radiative (fluorescence), characteristic times are taken from EADL database (EPICS2017 [9]).
 - b) Auger, characteristic times are taken from EADL database (EPICS2017 [9]).

6. Geometries

See description of geometries in MC code, e.g., here:

- 1) Kinematics of relativistic collisions: [10] Page 205, Appendix A

2) Rotation of the velocity vector after scattering: [10] Eq.(1.100)

By default, an incoming radiation is along Z-axis (unless specified otherwise in the input file), with the following convention:

$$X = R \sin(\varphi) \sin(\theta)$$

$$Y = R \sin(\varphi) \cos(\theta)$$

$$Z = R \cos(\varphi)$$

7. Experimental data to compare with

See SHI transport in various materials [8,12]

8. Probabilities of ionization within DOS

To select the energy level within the density of states (DOS), where an electron is ionized from, or where it jumps to, it is sampled from the DOS multiplied with the Fermi function of the given temperature (similar to e.g. [13]).

9. Electron emission

Currently, only emission of electron into vacuum is included following the model from Ref. [14].

A model of crossing boundary between two media is not included yet; nor there is any attraction of electrons back to the surface due to Coulomb interaction with the remaining holes.

Atomic motion in molecular dynamics (MD):

Atomic motion is traced with classical MD simulation using velocity Verlet algorithm [15].

Interatomic potentials do not depend on the electronic temperature, currently. The user may set which potential to use from the list of available ones. All the potential are considered short-range, smoothly cut at a given distance. For Coulomb potential, the Wolf's method of truncation is used in its energy-conserving implementation [16].

MD supercell is orthogonal, with the fixed size and shape (no barostats are implemented yet).

Model for low-energy electrons:

TTM - Unfinished!

II. Structure of the code

A few models in the code are executed in parallel, exchanging information on-the-fly [3]. Which means, all MC iteration are stored in memory to be propagated simultaneously. It requires rather large memory.

Some code conventions:

- 1) Every line (or a block of lines) of the code must be commented, explaining what it does and why it is here
- 2) Every function or subroutine call must be commented with the name of the module, where this function is coded
- 3) All global variables start with "g_", for instance g_numpar: all such variables are collected in the module "Variables", and all the global constants are in the module "Universal_constants"
- 4) All derived are in the module "Objects" (except module-specific types, such as e.g. in the module «Periodic_table»)
- 5) Module variables names start with "m_", e.g. "m_number", and are used only within the module (however, can be called from other module via *use*)
- 6) After the *use [module]* a list of the used functions should be provided using *only*
- 7) All local variables cannot start with "g_" or "m_" to avoid confusion
- 8) All paths to directories with input data are listed in the module "Read_input_data"
- 9) All paths and file names of the output data are in the module "Output"
- 10) The operator *goto* should be avoided wherever possible
- 11) If possible, functions and subroutines should have the attribute *pure*
- 12) Do-cycle and if-condition ends that do not have a marker "end(something)" should have a comment, defining which cycle they belong to, e.g. if (i<k) then ... endif ! (i<k)

File and module names:

- 1) The main file is TREKIS_main_file.
- 2) Each model should have its own file.
- 3) Modules, working with different file formats, are named as "Dealing_with_[format]", e.g. Dealing_with_cdf, Dealing_with_EADL, etc.
- 4) All module with cross sections are named as "CS_[particle]_[model]", e.g. CS_electrons_elastic, etc.
- 5) Modules, related to particular models, should have the model abbreviation in the start of their name. For example, all subroutines related to Monte Carlo are in the modules starting with "MC_", all modules relevant to molecular dynamics are in the files starting with "MD_", etc.
- 6) Modules not belonging to particular model, should have intuitive names.

INPUT_DATA directory

All the input data are in the directory **INPUT_DATA**. All the necessary folders and files must be present there to run the code, even if some of them are unused; they cannot be renamed.

The following files must be present there:

1) [INPUT_DATA.txt](#)

2) [NUMERICAL_PARAMETERS.txt](#)

3) [Directory Atomic_parameters](#)

In the folder [Atomic_parameters](#), the following files must be present:

- a) [INPUT_atomic_data.dat](#) – file containing the Periodic Table data.
- b) [ENDL_shell_designators.dat](#) – file defining shell designators in the ENDL format^b.
- c) [EADL2017.all](#) – database EADL^c with atomic parameters (data for Auger decays of outermost shells of elements #4 and #5 were added manually (from [17])).
- d) [EPDL2017.all](#) – database EPDL with photon cross sections.
- e) [Atomic_form_factors.dat](#) – atomic form-factors from PENELOPE [10].
- f) [Compton_profiles.dat](#) – Compton profiles from PENELOPE [10].
- g) [Pair_creation_coefficients.dat](#) – electron-positron pair creation parameters [10].

4) The directory [DOS](#) may contain DOS of various materials, used by default. Otherwise, free-electron DOS is used. DOS-file must contain two columns:

First, energy grid in eV. Second, electronic DOS in arbitrary units.

DOS parameters must be in accord with the parameters of the material defined in the file [\[matter\].txt](#) (see below):

a) DOS must be counted from the Fermi level, specified in that file. E.g., in dielectric or semiconductor, it is convenient to set fermi level = 0, so that it would coincide with the top of the valence band = 0. The exact zero values of the DOS exactly at the top of the valence band (VB) and the bottom of the conduction band (CB) must be present in the file: that's the way how the code interprets that it is a bandgap between these values (see example in the file [DOS_Si.dat](#)).

b) thusly defined bandgap – the distance between the top of the VB and the bottom of the CB in the file – must exactly coincide with the value of the band gap specified in the file [\[matter\].txt](#).

Any inconsistency between the parameters in the files will result in errors!

Note: in contrast to TREKIS-3, the CB DOS must be present, not only VB DOS, to define the probabilities of the electrons to be excited to. The CB data must not end with zero or near-zero values – meaning, it should be cut off at some finite value (before the artificial drop in the data due to finite basis size in calculation of the DOS).

Note #2: if you do not have the data for the CB DOS, place in the file the free-electron DOS, separated by the correct value of the bandgap in the material.

^b <https://www-nds.iaea.org/epdl97/>

^c <https://www-nds.iaea.org/epics/>

5) Directory `Material_parameters`

The folder `Materials_parameters` contains the data on materials already preset for TREKIS-4 calculations. The name of the material must coincide with the name given in the file `INPUT_DATA.txt` (see below). In each material directory, there must be a file present named: `[matter].txt` containing the following:

`[matter].txt` – the file describing material parameters (exactly in this order):

Line 1: chemical formula of the material, e.g. SiO_2 , Al_2O_3 , etc. Each element must start with a capital letter – that's how the code understand it's the start of a new element name (so, all the other letters must be small). The contribution of the element may be fractional (e.g. $\text{Lu}_{1.8}\text{Y}_{0.2}\text{SiO}_5\text{Ce}_{0.2}$).

Line 2: material density in $[\text{g}/\text{cm}^3]$

Line 3: Band gap [eV]

Line 4: Fermi energy [eV]

Line 5: Average speed of sound [m/s]

After the 5 mandatory line, optional line may be present.

CDF-parameters for VB and phonons may be set here (in the shape of Ritchie-Howie function [18]). The CDF is defined by the following type of data block:

Line 6: Valence - the marker indicating that this block defines the VB (or CB, for metals) CDF. Can be in the following form: ('VALENCE', 'Valence', 'valence', 'VAL', 'Val', 'val', 'Valent', 'valent', 'VALENT')

Line 7: Number of CDF oscillators (integer)

The following lines defined each oscillator, for the number of the function set in the previous line. The coefficients for each oscillator must follow exactly in this order in 3 columns:

A, E0, Gamma coefficients

Line N: Phonons - marker to set CDF for phonons (elastic scattering of electrons and holes on target atoms). Can be used in the following form: ('PHONONS', 'PHONON', 'Phonons', 'Phonon', 'phonons', 'phonon', 'PHON', 'Phon', 'phon')

Line N+1: number of CDF oscillators (integer)

For each oscillator, next lines specify the coefficients:

A, E0, Gamma coefficients

DOS can be set in two ways:

Option 1:

Line M: DOS - Marker to set the data block defining DOS ('DOS', 'DOs', 'Dos', 'dos')

Line M+1: `[DOS]` name of the file with the DOS (e.g., `DOS_Si.dat`)

Option 2:

Line M: BANDS – Marker indicating that the free-electron DOS will be used, specifying the bands widths of CB and VB ('BANDS', 'Bands', 'bands')

Line M+1: 4 real numbers specifying:

Bottom of the VB, top of the VB, bottom of the CB, top of the CB; all in [eV]. The distance between the top of the VB and the bottom of CB must coincide with the material bandgap (and be consistent with the specified Fermi energy specified above).

Without specified DOS, the default values are used: free electron DOS with the width of 10 eV.

One may specify the surface barrier for electron emission with the marker BARRIER or SURFACE.

Line K: BARRIER - marker to specify surface barrier ('BARRIER', 'Barrier', 'barrier', 'SURFACE', 'Surface', 'surface')

Line K+1: 3 real numbers, specifying:

Work function [eV]; surface barrier length [Å]; barrier height for electron emission [eV]

According to the model from Ref. [19].

If unspecified, the default values are used:

Work_func = 4.0 ! [eV]

Surf_bar = 10.0 ! [Å]

Bar_height = 6.0 ! [eV]

Note that currently, there are many files in this directory, but most of them are in a wrong format, and must be changed to comply with the format described above.

For many materials, the cdf-files from TREKIS-3^d can be reused, but they also have to be changed to comply with the format of TREKIS-4.

In the directory [Materials_parameters](#) one can also place the file containing the DOS. It's name must coincide with the filename provided in the file [\[matter\].txt](#) (see above). If the file is present in this directory, the code will use it. Otherwise, the code will check the given filename in the directory DOS described above.

The directory [Materials_parameters](#) will store the files with cross sections (CS) of scattering of all types of particles, calculated at the first run of the code. The files are named in the following format:

[particle]_[type]_CS_[kind]_[model].dat

where

[particle] – particle name, e.g. 'Au' for gold ion, 'Electron', etc.

[type] – scattering channel, e.g. elastic or inelastic etc.

[kind] – type, e.g. scattering on the valence band (valence), or total CS (total)

[model] – model used, e.g., Mott, CDF, PENELOPE etc.

E.g., file Electron_elastic_CS_total_Mott.dat

The CSs are calculated at first run of the code, and in all consequent simulations, they are reused from these files. That means, if any material parameters are changed in the file [\[matter\].txt](#) – such as CDF coefficients, material density, bandgap – the cross sections must be recalculated (see option to enforce CS recalculation in the file NUMERICAL_PARAMETERS, below).

^d <https://github.com/N-Medvedev/TREKIS-3>

6) Directory MD_input

This directory contains all the parameters of the supercell and MD calculations for each material. The directory of each material must coincide with the name of the material used in the directory [Materials_parameters](#). MD parameters needed for calculations are: initial atomic coordinates, their velocities, supercell size, numerical MD parameters. They can be set in the following format:

a) Setting supercell

Mandatory file [Supercell.txt](#) to specify the supercell parameters

The parameters can be set using the following markers:

Marker	Meaning
Supercell	Setting size of the supercell
Recenter	Place the supercell in the center of the simulation box (coordinates 0)
MC_MD_coupling	Specify the grid for exchange of the data between MC and MD modules
Thermostat	Parameters of the thermostat
Pressure_damp	Parameters of the pressure dampener (<i>untested!</i>)

After each marker, the following block of data must be specified exactly in the described format.

Supercell: specifies the MD supercell parameters: 3 lines defining the cartesian coordinates of the start and end of the supercell along each axis in [Å], and the index defining the boundary conditions (example in Figure 1):

```
X_start  X_end  ind_x
Y_start  Y_end  ind_y
Z_start  Z_end  ind_z
The supercell is always orthogonal.
```

```
1 Supercell
2 -10.0 10.0 1 ! Start and end of simulation box along X [Å]; 0=free boundary, 1=periodic boundary
3 -10.0 10.0 1 ! Start and end of simulation box along Y [Å]; 0=free boundary, 1=periodic boundary
4 0.0e0 10.0 1 ! Start and end of simulation box along Z [Å]; 0=free boundary, 1=periodic boundary
5
6 Recenter      ! Let the program recenter supercell around 0
7 XY            ! Around these axes
8
9 MC_MD_coupling ! Parameters for exchange of info between MC and MD
10 Cartesian 1    ! Which coord system: cartesian, spherical, cylindrical; what dimensionality (0,1,2,3)
11 3 2.0         ! Grid along which axis/axes: Cartesian:1=X,2=Y,3=Z; Spherical: 1=R,2=theta,3=phi; Cylindrical:1=R,2=L,3=phi; Grid step [Å] or [deg]
12
13 Thermostat
14 Berendsen
15 300.0e0 10.0e0 ! bath temperature [K]; characteristic time [fs]
16 2.0e0 2.0e0    ! Thickness of boundary with thermostat from X_start and from X_end of the supercell
17 2.0e0 2.0e0    ! Thickness of boundary with thermostat from Y_start and from Y_end of the supercell
18 -2.0e0 -2.0e0  ! Thickness of boundary with thermostat from Z_start and from Z_end of the supercell
19
20 !Pressure_damp
21 1000.0e0       ! Characteristic time [fs]
22 2.0e0 2.0e0    ! Thickness of boundary with pressure damping from X_start and from X_end of the supercell
23 2.0e0 2.0e0    ! Thickness of boundary with pressure damping from Y_start and from Y_end of the supercell
24 -2.0e0 -2.0e0  ! Thickness of boundary with pressure damping from Z_start and from Z_end of the supercell
```

Figure 1. example of the file Supercell.txt

If TREKIS-4 uses not only MD but also other modules (currently, MC; potentially TTM or others), the user must make sure the how coordinates of the supercell are placed with respect to the simulation window used in MC (what is their overlap, etc).

The index of the MD box boundaries (ind_[axis] along the specified axis) currently can be only:

0 – free boundary, or

1 – periodic boundary.

Note that this index does not have to coincide with the boundary conditions used in MC module: the excited particles may be treated differently from the atoms in MD. For example, excited particles may be absorbed at the boundaries, while atoms may be locked with periodic boundaries. This capabilities of using different boundary conditions may be important, since highly excited electrons may fly far away from the origin, while the atoms may be better represented with the box with periodic boundaries.

Note #2: the end of the supercell is only used if all the atomic coordinates are specified; if the supercell is constructed from the unit cells (see below), the end of the supercell will be defined automatically by the number of the given unit cells (in this case, it is most convenient to use the ‘recenter’ option to place the supercell correctly).

Recenter (optional keyword): places the supercell center to coordinate 0 along the specified axis. One can specify up to 3 cartesian axis. E.g., to shift the supercell along X and Y, one should specify XY, while for all 3 axes, use XYZ, etc. (see example in Figure 1). If no shifting of the supercell is required, delete or comment out this keyword in the file.

MC_MD_coupling (optional keyword): specifies how to transfer energy between MC and MD modules. If the keyword is absent, the default option is used: homogeneous energy transfer in the entire MD supercell: all the energy is distributed to atoms equally, independent of their coordinates. If the marker is used, it must follow by the following specifications (see example in Figure 1):

Two columns:

1) type of coordinate grid (“Cartesian”, “spherical”, “cylindrical”)

2) dimensionality of the grid: 1, 2, 3 (for 1d, 2d, or 3d grid)

Next, the number of lines must be present, specified by the dimensionality – one line for each dimension:

Each line must contain 2 columns: coordinate index (1=X, 2=Y, 3=Z (for Cartesian); 1=R, 2=theta, 3=phi (for Spherical); 1=R, 2=L, 3=phi (Cylindrical)); grid step in [Å] (for coordinates) or in [deg] (for angles in spherical or cylindrical coordinates – *untested*).

E.g., to specify two-dimensional grid along X (with the step 2.5 Å) and Z (with the step 3.5 Å), the following data block must be specified:

MC_MD_coupling ! marker

Cartesian	2	! coordinate system; dimensionality
1	2.5	! First coordinate: X; Step
3	3.5	! Second coordinate: Z; Step

Note that energy transferred from MC does not always coincide perfectly with the energy received in MD: the energy conservation depends on the grids chosen and atomic position in the supercell. E.g., there may be a situation in which the MC energy is transferred to regions with no atoms, thus this portion of the energy may be lost, see illustration:

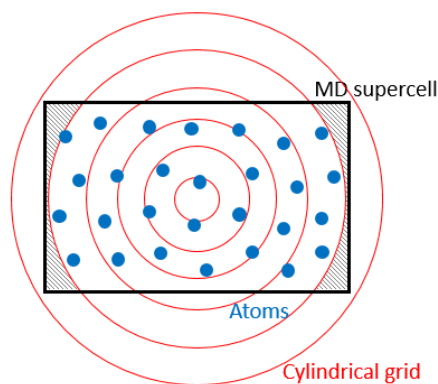


Figure 2. illustration of a mismatch of the MC grid and MD atoms – the hashed regions contain energy from MC but no atoms in MD.

Thermostat (optional keyword): parameters of the thermostat in the following format:

Line 1: thermostat type: Berendsen [20] (*currently, it is the only thermostat type defined in TREKIS-4*).

Line 2: 2 numbers real(8): thermostat temperature T_0 in [K], and its characteristic time τ in [fs].

Note that TREKIS-4 uses exact solution of the rate equation for temperature relaxation [21]: $dT/dt = (T - T_0)/\tau$, instead of the linearized solution from the original paper [20], which means, the characteristic time does *not* require the condition $\tau \gg dt$ (MD time step).

Next 3 lines specify the thicknesses of the thermostat (2 numbers), counted from the boundary of the MD supercell along each axes:

The thickness of the thermostated layer from the beginning of the cell (left wall), and the one from the end of the supercell (right wall), in [Å]; for each cartesian axes, see example in Figure 1.

To use the thermostat for the entire supercell, at least one of the thickness must be larger than the supercell size.

To switch off a thermostat along any chosen axis, a negative number can be set.

To switch off the thermostat completely, delete (or comment out) this keyword from the file (see example in Figure 1)

Pressure_damp(optional keyword):

To dampen pressure waves, one can use the method of gradual increase of viscosity at the boundaries, following Ref. [22]. In this case, specify the following line:

Line 1: characteristic time of the pressure dampening [fs]

Line 2 and on: Thickness of the layer in which the pressure is dampened (similar to specifications of the thermostat above), along each axis counted from the boundaries of the supercell.

Setting a negative number excludes pressure dampening along this axis; deleting the keyword switches off the pressure dampening in the entire supercell (see Figure 1)

Pressure dampening has not been tested yet!

b) Setting atomic coordinates

it can be done in two ways:

(i) setting coordinates of all the atoms in the supercell, or (ii) constructing the supercell from unit cells.

i) All atomic coordinates may be specified via file `Coordinates.xyz`, where all the atoms must be described in the XYZ^e format; coordinates are set in [Å]see example in Figure 3.

```
1 2
2 testing
3 Si -5.0d0 -5.0d0 5.0d0
4 Si 0.0d0 0.0d0 8.0d0
```

Figure 3. example of file `Coordinates.xyz`

ii) If there is no file `Coordinates.xyz` the second option is used. Constructing the supercell from the unit cells requires the parameters of the unit cell to be specified in the file `Unit_cell.txt` (only orthogonal cells are allowed!), in the following format:

The file must contain two blocks of data:

Coordinates – defining relative coordinates of atoms in the unit cell (in the units of the unit cell sizes along each axis),

Unit_cell – defining the unit cell size in absolute coordinates [Å].

The block *Coordinates* must contain the following lines:

Line 1: number of atoms in the unit cell

Then, the number of lines must follow, defining each atom:

Each line contains 4 columns:

- 1) Element name
- 2) Relative coordinate along X (real number between 0 and 1)
- 3) Relative coordinate along Y (real number between 0 and 1)

^e https://en.wikipedia.org/wiki/XYZ_file_format

- 4) Relative coordinate along Z (real number between 0 and 1)

In the block *Unit_cell* 2 lines must be present:

- 1) Three numbers (real), specifying the sizes of the unit cell along X, Y, Z in [Å]
- 2) Three numbers (integer) indicating how many unit cell must be used along each coordinate to construct the supercell

See example in Figure 4.

```
1 Unit_cell
2 5.42d0 5.42d0 5.42d0 ! size along X, Y, Z in [Å]
3 2 2 2 ! how many unit cells to be used in the supercell
4
5 Coordinates
6 8 ! how many atoms
7 Si 0.0e0 0.0e0 0.0e0 ! elements; relative coordinates inside of the unit cell
8 Si 0.5e0 0.5e0 0.0e0
9 Si 0.5e0 0.0e0 0.5e0
10 Si 0.0e0 0.5e0 0.5e0
11 Si 0.25e0 0.25e0 0.25e0
12 Si 0.75e0 0.75e0 0.25e0
13 Si 0.75e0 0.25e0 0.75e0
14 Si 0.25e0 0.75e0 0.75e0
```

Figure 4. Example of the Unit_cell.txt of solid Si.

c) Setting initial atomic velocities

Atomic velocities may be set in 2 ways: (i) all atoms in the supercell, or (ii) random velocities, according to the Maxwellian distribution (in the momentum space) for the given kinetic temperature.

i) To set velocities of all atoms, use the file [Velocities.xyz](#), in the save XYZ format (see example in Figure 3), where instead of the coordinates, velocities are set in the units of [Å/fs].

ii) If the file [Velocities.xyz](#) does not exist, TREKIS-4 will set the atomic coordinates randomly for the given temperature of the *first* target specified in the input file ([INPUT_DATA.txt](#)). The atoms get the initial temperature of $T_a=2*T$ and Maxwellian distribution, assuming that they are in their potential energy minimum: in this case, according to the equipartition theorem, half of the energy will be in the potential energy once they start moving, and the kinetic temperature will become equal to T after equilibration. The directions of the atomic momenta are set randomly in the solid angle.

In both cases of setting the atomic velocities, before the simulation start, the center-of-mass momentum of the supercell will be subtracted to exclude the macroscopic motion (*subtraction of the angular momentum is not ready yet, but in periodic boundaries simulation it is not important*).

d) Specifying MD interatomic potential

A file defining the potential must be present for each pair of atoms of the compound target in the following format: [\[atom1\]_\[atom2\]_Potential.txt](#)

Where [\[atom1\]](#) and [\[atom2\]](#) are the elements, whose interaction is defined with the potential in this file. For example, to defined interaction between Al and O, the file must be named Al_O_Potential.txt.

If the file containing the names of elements is absent, the code will check the file [Potential.txt](#), and use it if it exists (convenient for elemental targets, or if the same potential is used for many different element combinations).

This file must contain the following description of the potential:

Line 1: integer number, specifying how many different types of the potentials are set in this file. For each specified potential, the data block must be present:

Line 2 and on: potential marker, defined as follows:

Potential marker	Potential
LJ	Lennard Jones: $\frac{C_{12}}{r^{12}} - \frac{C_6}{r^6}$
Buck	Buckingham: $A \exp(-Br) - C/r^6$
Matsui	Matsui: $fB \exp((A - r)/B) - C/r^6$
SW	Stillinger-Weber 3-atom potential [23]: $E = \frac{1}{2} \sum_{i=1}^N \sum_{j=i_1}^{i_N} [V_{IJ}^R(r_{ij}) - V_{IJ}^A(r_{ij}) + u_{IJ}(r_{ij})$ $\times \sum_{\substack{k=i_1 \\ k \neq j}}^{i_N} u_{IK}(r_{ik}) \left(\cos \theta_{jik} + \frac{1}{3} \right)^2],$ $V_{IJ}^R(r) = A_{IJ} \varepsilon_{IJ} B_{IJ} \left(\frac{\sigma_{IJ}}{r} \right)^p \exp \left(\frac{\sigma_{IJ}}{r - a_{IJ} \sigma_{IJ}} \right),$ $V_{IJ}^A(r) = A_{IJ} \varepsilon_{IJ} \left(\frac{\sigma_{IJ}}{r} \right)^q \exp \left(\frac{\sigma_{IJ}}{r - a_{IJ} \sigma_{IJ}} \right),$ $u_{IJ}(r) = \sqrt{\lambda_{IJ} \varepsilon_{IJ}} \exp \left(\frac{\gamma_{IJ} \sigma_{IJ}}{r - a_{IJ} \sigma_{IJ}} \right),$
Power_law	Power law: $C r^n$
Exponential	Exponential potential: $A \exp(-kr)$
Coulomb	Coulomb: $\sim k Z_1 Z_2 e^2 \left(\frac{1}{r} + \frac{1}{d_{cut}} \right)$ - shifted and truncated Coulombic potential calculated using the Wolf's method [16]
Coulomb_Ewald	Coulomb: $k Z_1 Z_2 e^2 / r$ - long-range Coulombic potential, calculated Ewalds summation method (much slower then Wolf's)
Screened_Coulomb	Truncated Coulomb: $k Z_1 Z_2 e^2 / r$ - simple truncation of Coulomb (<i>not recommended, use Wolf's method instead!</i> [16])
Soft_Coulomb	F=Soft Coulomb: $k Z_1 Z_2 e^2 / (r + r_0)$ <i>Unfinished!</i>
ZBL	Universal ZBL (Ziegler, Biersack and Littmark) <i>Unfinished!</i>
Morse	Morse: $D_e [1 - \exp(-a(r - r_e))]^2 - D_e$ <i>Unfinished!</i>

Each potential requires their own number of parameters specified.

Line 3 and on (mandatory under the corresponding marker of the potential):

Potential marker	Parameters
LJ	Two lines: 1) d_{cut} δd truncation function parameters (in [\AA]) 2) C_{12} C_6 in [$\text{eV} \cdot \text{\AA}^{12}$] and [$\text{eV} \cdot \text{\AA}^6$], correspondingly
Buck	Two lines: 1) d_{cut} δd truncation function parameters (in [\AA]) 2) A (in [eV]) B (in [$1/\text{\AA}$]) C (in [$\text{eV} \cdot \text{\AA}^6$])
Matsui	Two lines: 1) d_{cut} δd truncation function parameters (in [\AA]) 2) A (in [\AA]) B (in [\AA]) C (in [$\text{eV} \cdot \text{\AA}^6$])
SW	5 lines: 1) d_{cut} δd truncation function parameters (in [\AA]; <u>must be</u> $d_{cut} > 1.5\sigma a$) 2) ε (in [eV/atom]), σ (in [\AA]), a (dimensionless) 3) A, B (dimensionless) 4) p, q (dimensionless) 5) λ, γ (dimensionless)
Power_law	Two lines: 1) d_{cut} δd truncation function parameters (in [\AA]) 2) C (in [eV]) n
Exponential	Two lines: 1) d_{cut} δd truncation function parameters (in [\AA]) 2) C (in [eV]) k (in [$1/\text{\AA}$])
Coulomb	Two lines: 1) d_{cut} δd truncation function parameters (in [\AA]; <i>δd is unused in this case but some number must be present</i>) 2) Z_1 Z_2 – effective charges of the first and second elements in units of [electron charge]
Coulomb_Ewald	Two lines: 1) d_{cut} δd truncation function parameters (in [\AA]; <i>are unused in this case but some numbers must be present</i>) 2) Z_1 Z_2 – effective charges of the first and second elements in units of [electron charge]
Soft_Coulomb	Two lines: 1) d_{cut} δd truncation function parameters (in [\AA]) 2) Z_1 Z_2 r_0 – effective charges of the first and second elements in units of [electron charge], and position of the shift [r_0]=[\AA]
Morse	Two lines: 1) d_{cut} δd truncation function parameters (in [\AA])

	2) D_e , a_{re} , in the units [eV], [$1/\text{\AA}$] and [\AA], correspondingly
ZBL	One line: d_{cut} δd truncation function parameters (in [\AA])

The truncation function parameters [real(8)]: d_{cut} δd specifying the truncation function parameters (in [\AA]), are always present (even if unused for some potentials), defined the Fermi-type soft truncation function:

$$f = \frac{1}{1 + \exp((R_{ij} - d_{cut})/\delta d)}$$

To exclude truncation, set d_{cut} larger than the simulation box (however, if possible, truncation should be used to speed up the calculations). The atomic interaction is set to zero at interatomic distances $R_{ij} > d_{cut} + 10 \delta d$.

If more then one potential is used (combining various contributions), each potential should be specified with its own block of data.

7) Directory Normalized_CDF

In the folder **Normalized_CDF**, a number of directories defining optical data for various chemical elements is present. Each directory contains the numerical data for CDF extracted from EPICS-2017, named **Optical_data_[shell].dat**, where [shell] is the shell name for the given element. These data are later used to construct the Ritchie-Howie type of the CDF (see below).

Automatically fitted CDF coefficients are saved in the files **Fitted_CDF_coefficients_[shell].dat**.

When setting a new materials, it is important to check if the automatically fitted CDF coefficients from the file Optical_data_[shell].dat produce a function that follows the experimental (or tabulated) data from the files Fitted_CDF_coefficients_[shell].dat, for each element! Also, sum rules must be checked.

CDF coefficients in TREKIS-4 are saved for each core shell in this type of a file, whereas the VB (or CB) CDF is saved in the file **[matter].txt**, described above.

Fitting of the CDF coefficients is performed at first run of TREKIS-4 for the given element, for which the files do not exist yet in **Normalized_CDF**. However, note that the automatic fitting is not always working well, so one must check the sum rules printed out. If the quality of fitting is not satisfactory, the coefficients must be adjusted manually and saved in the corresponding file **Fitted_CDF_coefficients_[shell].dat** in the following format:

Line 1: A E0 Gamma - marker of the coefficients (comment line, skipped by the code; just to remind ourselves the order of coefficients in the next lines)

Line 2: A E0 Gamma coefficients - the coefficients themselves

Line 3 and on, the same coefficients for the next oscillator, if present, etc.

8) Directory `Electron_cross_sections`

The folder `Electron_cross_sections` contains directories with the precalculated electron cross sections of scattering for various chemical elements; created at first run of TREKIS-4 involving the given element; reused at following runs; e.g. `Electron_inelastic_CS_L3_RBEB.dat`

Changing the CDF coefficients in the file `Fitted_CDF_coefficients_[shell].dat`, requires recalculation of the cross sections (see option in the input file).

9) Directory `Ion_cross_sections`

Analogously, `Ion_cross_sections` contains the cross sections of ionic cross sections for all elements.

Changing CDF coefficients in `Fitted_CDF_coefficients_[shell].dat`, requires recalculation of the cross sections (see option in the input file).

10) Directory `Photon_cross_sections`

Analogously, the folder `Photon_cross_sections` contains the data on photon cross sections.

11) Directory `Positron_cross_sections`

Analogously, the folder `Positron_cross_sections` contains the cross sections of positron cross sections for all elements.

Changing CDF coefficients in `Fitted_CDF_coefficients_[shell].dat`, requires recalculation of the cross sections (see option in the input file).

12) Directory `DSF`

Directory `DSF` will contain DSFs for alternative cross sections calculations; *Not ready*.

III. Compiling the code

Use make-files:

Windows: `Make.bat` (rename `Make.txt` -> `Make.bat`)

Linux: `Makefile`

Run them in the terminal, which assumes that the environmental path variable contains the paths to intel-fortran compiler (version 12 or newer), and OpenMP^f libraries.

Other compilers were not tested.

Note: During compilation, the code may report several of the following type of warnings:

“warning #6371: A jump into a block from outside the block may have occurred. [8391]”

Those are normal and are there by design of the algorithm; nothing to worry about.

^f <https://www.openmp.org/>

Windows:

1. Installation of the necessary components

The tested way to compile and run the code (using other methods are at your own risk):

Install Microsoft Visual Studio Community, freely available in ^g. Intel Fortran compiler is freely available from the oneAPI^h.

First, install the Visual Studio (VS), reboot your computer to initialize it. Then, install oneAPI Base Toolkit, and after that oneAPI HPC Toolkitⁱ, containing Intel Fortran. This order of installation is important so that the system writes all the environmental variables in VS.

Apart from VS, one can also use the following text editors for programming, Kate^j, Notepad++^k, or others.

2. Compiling the code in Windows

Once Visual Studio and intel-fortran are installed (from oneAPI), Visual Studio contains its own terminal with all the paths included (named «Intel oneAPI command prompt for Intel 64 for Visual Studio 2022» or something similar; can be found in the «Start» menu at the top right after installation of oneAPI (look for keywords «command prompt»)). In this terminal, run the script Make.bat.

The following compilation options are possible:

Make.bat DEBUG – compiles the debugging version of TREKIS-4, with no parallelization and optimizations included, called TREKIS_DEBUG.exe (use for debugging)

Make.bat DEBUGOMP (or for short: make db) – compiles the code with debug, without optimization, but with parallelization via OpenMP (useful for testing and debugging parallelization issues). Creates TREKIS_DEBUG_OMP.exe

Make.bat FAST (or make slow) – compiling with OpenMP, but without optimizations, meaning fast compilation but not so fast execution of the code; creates TREKIS_OMP.exe

Make.bat (or: make) – compiles code for release, with all optimizations and parallelization; named TREKIS.exe

3. Compiling in Microsoft Visual Studio (VS)

- 1) Create new FORTRAN- project (File->New->Project). In the opened window, select Intel Visual Fortran->Console Application->Empty Project. Chose the directory for the project.

^g <https://visualstudio.microsoft.com/vs/community/>

^h <https://www.intel.com/content/www/us/en/developer/tools/oneapi/base-toolkit-download.html>

ⁱ <https://www.intel.com/content/www/us/en/developer/tools/oneapi/hpc-toolkit-download.html>

^j <https://kate-editor.org/>

^k <https://notepad-plus-plus.org/>

- 2) Add all the code files from Source_files to the project: In VS in the window Solution Explorer right click on the Source Files, select Add -> Existing Item, add ALL .f90 files. For a correct execution, the folder INPUT_DATA must be present in the same project as the file [project_name].vfproj
- 3) Tune the project. Option Project -> [project_name] Properties, the window [project_name] Properties Pages. Make sure the correct (Configuration) is set for compiling: Debug or Release. In the opened window, set the following:
 - a. Fortran->General, Optimization - Maximum speed plus higher level optimization (/O3)
 - b. Fortran -> Optimization, Parallelization - Yes (/Qparallel)
 - c. Fortran -> Optimization, Prefetch Insertion – Medium
 - d. Fortran -> Optimization, Interprocedural Optimization - Multi-file (/Qipo)
 - e. Fortran -> Optimization, Favor speed or size - Favor fast code
 - f. Fortran -> Preprocessor, Preprocessor Source File – Yes(/fpp)
 - g. Fortran -> Preprocessor, OpenMP Conditional Compilation – Yes
 - h. Fortran -> Language, Process OpenMP Directives – Generate Parallel Code (/Qopenmp)
 - i. Fortran -> Command Line, in the field Additional Options add /D OMP_inside

For diagnostic/debugging, use the following options:

- j. Fortran -> Diagnostics, Vectorizer Diagnostic Level - Loops Successfully Vectorized (1) (/Qvec-report1)
 - k. Fortran -> Run-time, Check Array and String Bounds – Yes
 - l. Fortran -> Run-time, Check Uninitialized Variables – Yes
- 4) Create the executable: Build -> Build [project_name]

4. Dependencies and libraries

4.1 Gnuplot

For creating the plots automatically in TREKIS-4 in Windows, GnuPlot¹ must be installed.

After installation, it may require to write the path to gnuplot.exe in the environmental variable PATH (usually it is done automatically, but if it does not work, follow these steps):

1. In **Start** right click on **Computer**. Select **Properties**. (In Windows 10, right click on **Start**, select **System**, click on **System properties** or Advanced system setting)
2. In the dialog window **System** select **Additional system parameters**.
3. In the tab **Additionally** in the window **System properties** click **Environmental variables**.
4. In the field **System variables** in the window **Environmental variables** find variable **Path**, select it and use **Change**.
5. In the opened window, **Change System variables** scroll to the end and add (;).After it, add your path to gnuplot.exe, e.g.: **C:\Program Files\gnuplot\bin**.

¹ <http://www.gnuplot.info/>

- a. In Windows 10 it is in the windows **Change Environmental variables** where use **Create** and add the path to GnuPlot.exe in the new field (e.g., **C:\Program Files\gnuplot\bin**).
6. Click OK in all the 3 windows.
7. If Visual Studio was opened, close it for the changes to take effect.

4.2 MKL and OpenMP

Math Kernel Library^m – MKL (*currently unused in TREKIS-4*). In Intel terminal, the path are written. To use MKL inside the Visual studio project, allow it:

1. Select Project > Properties > Fortran > General > Additional Include Directories. Add the architecture-specific lib folder,

for example, C:\Program Files\Intel\MKL\10.2.6.037\include

2. Select Project > Properties > Linker > General > Additional Library Directories. Add the architecture-specific lib folder,

for example, C:\Program Files\Intel\MKL\10.2.6.037\ia32\lib

3. Select Project > Properties > Linker > Input > Additional Dependencies.

Insert

mkl_intel_c.lib mkl_intel_thread.lib mkl_core.lib libiomp5md.lib

Linux:

In a terminal, use makeⁿ. All the library paths must be specified in your system/script.

Use the following compilation options:

make db=y – to compile the code with debug options and no optimizations

make OMP=no – to compile the code without OpenMP parallelization but with optimizations

make – to compile for release with all optimizations and parallelization

All options create the code with the same name TREKIS.x.

^m <https://software.intel.com/content/www/us/en/develop/tools/math-kernel-library.html>

ⁿ E.g. https://www.tutorialspoint.com/unix_commands/make.htm

IV. Input parameters: INPUT_DATA.txt

```

1  ::: TARGET :::
2  SiO2          ! Whole sample name
3  1             ! how many target components there are
4  SiO2          ! Material of this part of the target
5  0.0           ! Temperature of the target [K]
6  0            ! type of the first component: Rectangle
7  0.0  0.0  0.0 ! coordinates of its center
8  -1e20  1e20   ! Length: its beginning and its end along X axis [A]
9  -1e20  1e20   ! Length: its beginning and its end along Y axis [A]
10 -1e20  1e20   ! Length: its beginning and its end along Z axis [A]
11 0.0e0         ! its rotation around X-axis (degree)
12 0.0e0         ! its rotation around Y-axis (degree)
13 0.0e0         ! its rotation around Z-axis (degree)
14 1             ! Type of surface potential barrier: 0=step, 1=Eckart
15 ::: RADIATION ::: SHI
16 1             ! How many incoming particles/pulses to model
17 1             ! How many particles are in the 1st bunch of radiation
18 3             ! Type of incoming particle: 0=photon, 1=electron, 2=positron, 3=SHI, 4=hole
19 0.0  0.0  0.0 ! [A] coordinates of impact: X, Y, Z
20 0.0  0.0  0.0 ! [A] spread (or uncertainty) of coordinates of impact along X, Y, Z
21 0.0  0.0      ! [degrees] angles of impact: from Z [0:180], from Y [0:360]
22 100e6        ! [eV] energy of the incoming particle / pulse
23 0.0          ! [eV] spread of energies (or energy uncertainty)
24 0.0          ! [fs] arrival time of the incoming particle / center of the pulse
25 0.0          ! [fs] FWHM-duration of the pulse (ignored for single particles)
26 79           ! Z atomic number of SHI
27 1 -1.0       ! Zeff, Meff

```

Figure 5. Example of the file INPUT_DATA.txt

The file must contain the following lines, exactly in the specified order:

- 1) ::: TARGET ::: – comment line
- 2) material name (used to name output data; 100 characters max)
- 3) number of targets (integer):
Each target may have its own elements and geometry, but this option is not ready yet – currently only 1 target is allowed
- 4) [\[matter\]](#) name of the first target material. Must coincide with the material lime file: [\[matter\].txt](#)
- 5) Initial temperature of the target [K]
- 6) The number of next lines depends on the number of the components of the target:
 Index of the geometry for each target component (integer). Currently, only these geometries are allowed (the number of lines for each type of geometries is different see below):
 Index of the type of geometry:
 0 = parallelepiped:
 Requires 7 lines (example of the numbers below can be changed):
 0 0 0 ! coordinates of its center
 -1e20 1e20 ! Length: its beginning and its end along X axis [A]
 -1e20 1e20 ! Length: its beginning and its end along Y axis [A]
 -1e20 1e20 ! Length: its beginning and its end along Z axis [A]
 0.0e0 ! its rotation around X-axis (degree)
 0.0e0 ! its rotation around Y-axis (degree)
 0.0e0 ! its rotation around Z-axis (degree)

1 = Sphere

Requires 2 lines:

0 0 0 ! coordinates of its center
10.0e0 ! its radius [A]

2 = Section of a sphere

Requires 4 lines:

0 0 0 ! coordinates of its center
0.0e0 10.0e0 ! starting and ending points by radius [A]
0.0e0 1.0e0 ! starting and ending points by phi angle within $[-\pi/2.. \pi/2]$
0.0e0 1.0e0 ! starting and ending points by theta angle within $[0..2*\pi]$

The angle section is specified in the relative coordinates between 0 and 1, meaning the phi-angle is in the absolute unit will be in the range $-\pi/2$ to $\pi/2$, while theta angle relative coordinate between 0 and 1 specifies the absolute units in the range 0 to $2*\pi$.

3 = cylinder

Requires 6 line:

0 0 0 ! coordinates of its center
0.0e0 ! radius [A]
-1e20 1e20 ! starting and ending points of the cylinder defining its length [A]
0.0e0 ! its rotation around X-axis (degree)
0.0e0 ! its rotation around Y-axis (degree)
0.0e0 ! its rotation around Z-axis (degree)

4 = Section of the cylinder

Requires 7 lines:

0 0 0 ! coordinates of its center
0.0e0 0.0e0 ! starting and ending radius (if it's cut as cylindrical layer) [A]
0.0e0 0.0e0 ! starting and ending points of the cylinder defining its length [A]
0.0e0 1.0e0 ! starting and ending points by theta angle within $[0..2*\pi]$
0.0e0 ! its rotation around X-axis (degree)
0.0e0 ! its rotation around Y-axis (degree)
0.0e0 ! its rotation around Z-axis (degree)

Currently, only parallelepiped is fully implemented in TREKIS-4

14) Model for the electron emission barrier at the surface: 0=step-function [25], 1=Eckart [19]

15 or larger) ::: RADIATION ::: - a comment line indicating the start of the next block of data, describing the incoming particles

16) Number of incident bunches of particles

17) Number of particles in the first bunch: by default it should be 1 to model an individual particle impart instead of a pulse/bunch.

Other options are currently unfinished, TREKIS-4 only allows 1 particle per bunch.

18) Type of incident particle: 0=photon, 1=electron, 2=positron, 3=SHI

19) Three numbers (real) – coordinates of arrival of the particle to the «surface» of the material along X, Y and Z in [Å].

20) Three numbers – scattering/uncertainty of the coordinates along X, Y and Z (to mimic a pulse of a finite width – each MC iteration will randomly place the particle around the given coordinates with the given width); set zeros to have an impact exactly at the given coordinates.

21) two numbers defining angle of incidence: [degrees] from Z [0:180], from Y [0:360]

22) Particle energy in [eV]

23) Scattering/uncertainty in energy (to mimic pulse/bunch, in [eV])

24) Arrival time of the particle / center of the pulse in [fs].

25) Duration of the pulse/bunch [fs] – scattering of the arrival time around the given central value.

The next two lines must only be present in case of an incoming ion (swift heavy ion, SHI); they should not be there for other types of irradiation:

26) atomic number of the ion in the Periodic Table

27) Two numbers: charge of the SHI in [e], its mass [a.m.u]. These numbers are used in case in in the NUMERICAL_PARAMETERS.txt (see below) the model of the charge is chosen to be the fixed ion charge; otherwise, another model will recalculate an effective charge, see below. To use a default ion mass (from the file INPUT_atomic_data.dat), set any negative number here.

After these lines, the same lines from 14 to 24 (or 26) can be used for the next incoming bunch, if there were more than 1 particle specified. See example in Figure 5.

V. Input file NUMERICAL_PARAMETERS.txt

This file contains definitions of the models and numerical parameters.

```

1  ::: NUMERICAL PARAMETERS :::
2  100      ! number of MC iterations to be performed
3  3        ! number of threads for parallel calculations with OpenMP (1 = nonparallelized; <=0 = all available processors)
4  1.0d0    ! [fs] Time-step
5  1.0d0    ! [fs] How often to print out data into files;  [tgrid.txt]
6  -1.0d0   ! [fs] when to start simulation
7  100.0d0  ! [fs] when to stop simulation
8  1        ! Activate MC module (1) or not (0)
9  1        ! Activate MD module (1) or not (0)
10 0        ! Activate TTM module (1) or not (0)
11 F        ! Recalculate cross sections and MFPs (T=true; F=false)
12 -1.0e10  1.0e10 F ! Simulation box size along X [A], and whether to reset it to coincide with MD supercell (True;False)
13 -1.0e10  1.0e10 F ! Simulation box size along Y [A], and whether to reset it to coincide with MD supercell (True;False)
14 0.0e0    10.0e0 T ! Simulation box size along Z [A] (for SHI set default: 0 10.0e0 ; for other particles may use -1.0e10 1.0e10)
15 0 0 1    ! Kind of boundary along X, Y, Z (0=absorbing; 1=periodic; 2=reflective; 3=white)
16 ::: MODELS FOR ELECTRONS :::
17 0        ! MC or MD target model: 0=MC, 1=MD
18 0        ! Include forces and fields among electrons: 0=exclude, 1=include
19 3        ! inelastic scattering: 0=excluded, 1=relativ.CDF, 2=RBEb, 3=delta, 4=nonrelativ.CDF (DO NOT USE!), 5=SPdelta
20 1        ! elastic scattering: 0=excluded, 1=CDF, 2=Mott, 3=DSF (NOT READY YET!), 5=SPdelta
21 1        ! Bremsstrahlung: 0=excluded, 1=BHW
22 0        ! Cherenkov radiation: 0=excluded, 1=... (NOT READY YET!)
23 1.0d0    ! [eV] Cut-off energy (electrons with lower energies are excluded from calculation)
24 ::: MODELS FOR PHOTONS :::
25 1        ! Photoabsorption CSs: 0=excluded, 1=CDF, 2=EPDL
26 1        ! Compton effect: 0=excluded, 1=PENELOPE
27 1        ! Thomson / Rayleigh scattering: 0=excluded, 1=PENELOPE
28 1        ! Electron-positron pair creation: 0=excluded, 1=PENELOPE
29 0        ! Photonuclear Physics: 0=excluded, 1=included (NOT READY YET!)
30 -8.9d20  ! [eV] Cut-off energy (photons with lower energies are excluded from calculation)
31 -1       ! Photon attenuation length [A] (<0 takes values from EPDL; >=0 uses value given here)
32 ::: MODELS FOR SHI :::
33 3        ! inelastic scattering: 0=excluded, 1:3=delta, 4=nonrelativ.CDF (DO NOT USE!), 5=SPdelta
34 0        ! Charge state: 0=Barkas; 1=Bohr; 2=Nikolaev-Dmitriev; 3=Schwartz-Grande; 4=fixed Zeff, 5=charge exchange
35 0        ! SHI charge shape: 0=point-like charge; 1=Brandt-Kitagawa ion
36 -10.0d20 ! [eV] Cut-off energy (SHI with lower energies are excluded from calculation)
37 ::: MODEL FOR POSITRONS :::
38 3        ! inelastic scattering: 0=excluded, 1:3=delta, 5 = SPdelta
39 0        ! elastic scattering: 0=excluded, 1=CDF, 2=Mott, 3=DSF (NOT READY YET!)
40 0        ! Bremsstrahlung: 0=excluded, 1=BHW
41 0        ! Annihilation: 0=excluded, 1=Heitler
42 10.0d20  ! [eV] Cut-off energy (positrons with lower energies are excluded from calculation)
43 ::: MODEL PARAMETERS FOR CDF :::
44 0        ! CDF model for slow particles: 0 = Ritchie CDF, 1 = Mermin CDF
45 0 -1     ! target dispersion relation: 0=free electron, 1=plasmon-pole, 2=Ritchie; gff.mass [in me] (0=from DOS of VB; -1=free-electron)
46 0        ! Include plasmon integration limit (0=no, 1=yes)
47 10.0d0   ! coeff.k where to switch from nonrelativistic to Delta CDF for INELASTIC scattering: E = k * Wmin (default is 10.0)
48 10.0d0   ! coeff.k where to switch from nonrelativistic to Delta CDF for ELASTIC scattering: E = k * Wmin (default is 5.0)
49 0        ! use for target atoms Zeff (set 0), or Z=1 (set 1)
50 50       ! n grid points for INELASTIC cross section integration over energy (E): dE = max((E - E0(:)), G(:))/n
51 100      ! n grid points for INELASTIC cross section integration over momentum (Q): dQ = max((Q - (W-E0(:))), G(:))/n
52 10       ! n grid points for ELASTIC cross section integration over energy (E): dE = max((E - E0(:)), G(:))/n
53 100      ! n grid points for ELASTIC cross section integration over momentum (Q): dQ = max((Q - (W-E0(:))), G(:))/n
54 ::: MODELS FOR HOLES :::
55 1        ! Auger decays: 0=excluded, 1=EADL
56 1        ! Radiative decays: 0=excluded, 1=EADL
57 -1       ! [me] effective valence hole mass (-1=from DOS; 0=free electron; >0=fixed mass in units of electron mass)
58 1        ! inelastic scattering: 0=excluded, 1=CDF, 2=RBEb, 5=SP-CDF
59 1        ! elastic scattering: 0=excluded, 1=CDF, 2=Mott, 3=DSF (NOT READY YET!), 5=SP-CDF
60 1.0d0    ! [eV] Cut-off energy (holes with lower energies are excluded from calculation)
61 ::: MD MODEL PARAMETERS :::
62 F 0.0d0  1.0d0 ! Use quenching (T=true; F=false), when to start [fs], how often nullify velocities [fs]
63 ::: OUTPUT DATA :::
64 PNG      ! format of figures prepared by gnuplot (eps, jpeg, gif, png, pdf); set 0 if no gnuplot scripts are required
65 F        ! printout DOS (T=true, F=false)
66 T        ! printout MFPs (T=true, F=false)
67 F        ! printout Ranges (T=true, F=false)
68
69 !print_each_step ! to printout timing of each MD step
70
71 !cohesive        ! calculate cohesive energy only, no TREKIS run
72
73 print_MC_MD_energy ! printout MC-MD energy transfer
74
75 print_XYZ         ! to printout atomic coordinates from MD

```

Figure 6. example of the file NUMERICAL_PARAMETERS.txt

Must contain the following lines:

- 1) ::: NUMERICAL PARAMETERS ::: - comment line
- 2) Number of MC iterations (integer)
- 3) Number of threads for parallelization in OpenMP (integer). A negative number sets it equal to the maximal number of threads in the system.
- 4) MD timestep. May be set in two ways:
 - a) constant step in [fs] (real)
 - b) Filename (character) – containing an instruction of how the step must be changed in time. The file must have 2 columns: first one specifies the time instance, and which the MD timestep must be changed; the second one specifies the values of the timestep to be used starting from this time instant.

For example, we want to set at the time -100 fs the step of 1 fs; then, from 0 fs (time of SHI arrival) we change the step to 0.1 fs, after the system is cooled down at 500 fs, we again increase the step to 1 fs – then the file must contain the following:

```
-100.0 1.0
0.0    0.1
500.0  1.0
```

If such a file is present, it has the priority over the number set in the line 6 below.

5) How often to printout the output data, can be set in 2 ways:

A) timestep for printout [fs] (real) (cannot be smaller than the MD step).

B) Filename (character) – setting the output time grid. The file must be preset in the directory INPUT_DATA. It must contain at least one column: time grid, at which instants to printout the data (fs). The second column (optional) may set the MD timestep too, analogously to the setting of the MD step above. For example, a file containing:

```
0.1    0.01
1.0    0.1
10     1
```

Indicates that the MD step will be changed to 0.01 fs until the time instant of 0.1 fs, and at the time 0.1 fs, there will be a printout. Then, the MD simulation timestep will be changed to 0.1 fs, until 1 fs, when the new printout will take place. Then, the MD step will change to 1 fs, until 10 fs.

6) Simulation start [fs] (real) – can be set before the arrival time of the incident particle (important for MD simulations to thermalize the atomic system, but not necessary if no MD simulation is used)

7) Duration of the simulation [fs] (real)

8) Use MC module or not (1=use, 0=off).

9) Use MD module or not (1=use, 0=off).

10) Use TTM module or not (1=use, 0=off). *Not ready*

11) Recalculate the cross sections (T), or use the precalculated from the files, if available (F). (logical: T=true, F=false)

12) Two real numbers and a logical: the beginning and the end of the simulated window along X [Å]; marker to use the sizes equal to those specified in the MD supercell (T) or not (F)

13) Two real numbers and a logical: the beginning and the end of the simulated window along Y [Å]; marker to use the sizes equal to those specified in the MD supercell (T) or not (F)

14) Two real numbers and a logical: the beginning and the end of the simulated window along Z [Å]; marker to use the sizes equal to those specified in the MD supercell (T) or not (F)

15) Three integer numbers: indicating types of the boundaries of the simulation window along X, Y, Z.

The types are set as:

0=absorbing boundary – an MC particle arriving to the boundary is absorbed and disappears from simulation.

1=periodic – periodic boundary conditions.

2=reflective – particle is elastically reflected from the boundary.

3=white – «white noise», particle is reflected from the boundary at a random angle.

These conditions are set for all secondary particles produced in MC simulation, whereas the incoming particles from the bunch always disappears reaching the back boundary (goes through the target).

16) ::: MODELS FOR ELECTRONS ::: - comment line

17) What model to use for target structure in MC (set 0 in this line): random homogeneous target (standard MC), or use MD-atomic coordinates (set 1 in this line – *not ready, cannot be used*)

18) Include Coulomb interaction among the MC particles, or use asymptotic trajectories. 0=exclude, 1=include. *Not ready, only asymptotic trajectory can be used currently*

19) Index of the model for inelastic scattering cross section of electrons: 0=excluded, 2=RBEB, 3=delta CDF [6], 5=SPdelta

Currently, only indices 0 (switch off), 3 or 5 for CDF models are available. *Options 1 and 4 should not be used – they are too slow. The RBEB model (index 2) is unfinished and cannot be used.*

20) Index of the model for elastic scattering of electrons: 0=excluded, 1=CDF, 2=Mott, 3=DSF
DSF model is not ready

21) Index of Bremsstrahlung model: 0=excluded, 1=BHW

22) Index of model for Cherenkov radiation: 0=excluded. *Not ready*

23) Energy cut off threshold in [eV] (electrons with smaller energies are stopped). To exclude cut off, set any non-positive number.

24) ::: MODELS FOR PHOTONS ::: - comment line

25) Index of the model of photoabsorption: 0=exclude, 1=CDF, 2=EPDL2017

26) Index of the model of Compton effect: 0=exclude, 1= PENELOPE

27) Index of the model of Thompson (elastic) scattering of photons: 0=exclude, 1= PENELOPE

28) Index of the model of electron-positron pair creation: 0=exclude, 1= PENELOPE, 2=include Landau-Pomeranchuk effects (*not ready*)

29) Index of the model of photo-nuclear effect: 0=exclude (*not ready*)

30) Energy cutoff threshold for photon in [eV]. To exclude cut off, set any non-positive value.

31) Effective attenuation length of photon (in [Å]). (setting <0 – assumes the default value from EPDL; >=0 uses the given value)

32) ::: MODELS FOR SHI ::: - comment line

33) Index of the model of inelastic scattering of swift heavy ion: 0=excluded, 1-3=delta CDF [6], 5=SPdelta

34) Index of the model for effective charge of SHI: 0=Barkas; 1=Bohr; 2=Nikolaev-Dmitriev; 3=Schiwietz-Grande, 4=fixed Zeff, 5=charge exchange (*5 is not ready*)

35) SHI charge shape: 0=point-like, 1=Brandt-Kitagawa model (*1 is not ready*)

36) Energy cutoff threshold for photon in [eV]. To exclude cut off, set any non-positive value.

37) ::: MODEL FOR POSITRONS ::: - comment line

38) Index of the model of inelastic scattering of positron: 0=excluded, 1-3=delta CDF [6], 5=SPdelta

39) Index of the model of inelastic scattering of positron: 0=excluded, 1=CDF, 2=Mott, 3=DSF (*3 is not ready*)

40) Bremsstrahlung: 0=excluded, 1=BHW

41) Index of the model of positron annihilation: 0=excluded, 1= Heitler

42) Energy cutoff threshold for positron in [eV]. To exclude cut off, set any non-positive value.

- 43) ::: MODEL PARAMETERS FOR CDF ::: - comment line
- 44) Index of the model for CDF for slow particles: 0 = Ritchie CDF, 1=Mermin CDF, 2=Full conserving CDF (*not ready*). Note: for fast particles, delta-CDF is always used.
- 45) Two numbers specifying:
- a) target dispersion relation: 1=free electron, 2=plasmon-pole, 3=Ritchie;
 - b) effective mass [in m_e] used ofr CDF cross section (0=effective mass from DOS of VB; -1=free-electron)
- 46) Account for plasmon-integration-limit or not (0=no, 1=yes; default is 0)
- 47) Coefficient k, setting the energy value where to switch from slow-particle Ritchie model to fast-particle delta-CDF in *inelastic* scattering: $E = k * E_{eq}$ (E_{eq} is the minimal energy for each delta-oscillator, see [6])
- 48) Coefficient k, setting the energy value where to switch from slow-particle Ritchie model to fast-particle delta-CDF in *elastic* scattering: $E = k * E_{eq}$ (E_{eq} is the minimal energy for each delta-oscillator, see [6])
- 49) Index of the model of effective charge of target atoms: 0=Barkas-like; 1=1 (constant)
- 50) Parameter n setting the energy integration grid in Ritchie or Mermin CDF numerical integration in the *inelastic* scattering: $dE = \max((E - E_0(:)), \text{Gamma}(:))/n$, with E being the incident particle energy, E_0 and Gamma – CDF oscillators parameters.
- 51) Parameter n setting the momentum integration grid in Ritchie or Mermin CDF numerical integration in the *inelastic* scattering: $dQ = \min\{Q, \max((Q - (W - E_0(:))), \text{Gamma}(:))\}/n$, where Q is the momentum (recoil energy), W is the transferred energy.
- 52) Parameter n setting the energy integration grid in Ritchie or Mermin CDF numerical integration in the *elastic* scattering.
- 53) Parameter n setting the momentum integration grid in Ritchie or Mermin CDF numerical integration in the *elastic* scattering.
- 54) ::: MODELS FOR HOLES ::: - comment line
- 55) Index of the Auger-decay model for core holes: 0=excluded, 1= EADL2017
- 56) Index of the radiative decay model for core holes: 0=exclude, 1=EADL2017
- 57) Effective mass of valence hole in [m_e], used in the dispersion law (negative value sets to use the mass from DOS; 0 = free electron mass; >0 = given value in the units of m_e)
- 58) Index of the model of the inelastic scattering of VB hole: 0=excluded, 1=Ritchie CDF
- 59) Index of the model of the elastic scattering of VB hole: 0=excluded, 1=CDF, 2=Mott, 3=DSF (*3 is not ready*)
- 60) Energy cut off threshold for VB hole transport in [eV].
- 61) ::: MD MODEL PARAMETERS ::: - comment line
- 62) Three columns:
- a) Use quenching (logical: T=true, F=false)
 - b) Starting time of quenching [fs]
 - c) How often atomic velocities should be nullified [fs]
- 63) ::: OUTPUT DATA ::: - comment line
- 64) Format of the gnuplot files output: eps, jpeg, gif, png, pdf. To exclude gnuplotting, set 0.

- 65) Printing out DOS (+ effective momentum, effective mass) (logical: T=true, F=false)
 66) To printout mean free paths of not (logical: T=true, F=false)
 67) To printout Range of SHI and electrons (logical: T=true, F=false)

After the mandatory lines, some optional lines may be used to specify more output. Each output requires a data block of two or three lines:

- Marker / codeword
- Type of the output grid
- three numbers (real): start, end, and step fo the grid

The following markers are allowed in TREKIS-4:

Table 1. Keywords for output data printout.

General parameters	
print_each_step	Sets to printout the timestep index on the screen for each MD step (instead of the output steps only); no lines are required after this one, only the keyword.
MC data	
Energy	Printout energy spectrum of particles
Spectra_[axis]	Printout spatially resolved energy spectrum along the axis [axis] (X,Y,Z)
print_theta	Printout the velocity angular distribution of particles relative to theta angle counted from Z
X	Printout the spatial distribution of particles along X
Y	Printout the spatial distribution of particles along Y
Z	Printout the spatial distribution of particles along Z
XY	Printout the spatial distribution of particles in the plane X-Y (<i>not ready</i>)
XZ	Printout the spatial distribution of particles in the plane X-Z (<i>not ready</i>)
YZ	Printout the spatial distribution of particles in the plane Y-Z (<i>not ready</i>)
XYZ	Printout the spatial distribution of particles in the volume X-Y-Z (<i>not ready</i>)
R	Printout the spatial distribution of particles by radius R in cylindrical coordinates
L	Printout the spatial distribution of particles by depth L in cylindrical coordinates (<i>not ready</i>)
Phi	Printout the spatial distribution of particles by angle phi in cylindrical coordinates (<i>not ready</i>)
RL	Printout the spatial distribution of particles by radius and depth R-L in cylindrical coordinates
RTheta	Printout the spatial distribution of particles by radius and angle R-theta in cylindrical coordinates (<i>not ready</i>)
RLTheta	Printout the spatial distribution of particles by volume in cylindrical coordinates (<i>not ready</i>)

Rs	Printout the spatial distribution of particles by radius R in spherical coordinates (<i>not ready</i>)
Thetas	Printout the spatial distribution of particles by angle theta in spherical coordinates (<i>not ready</i>)
Phis	Printout the spatial distribution of particles by angle phi in spherical coordinates (<i>not ready</i>)
RPhis	Printout the spatial distribution of particles by radius and angle R-phi in spherical coordinates (<i>not ready</i>)
RThetas	Printout the spatial distribution of particles by radius and angle R-theta in spherical coordinates (<i>not ready</i>)
RThetaPhis	Printout the spatial distribution of particles by volume in spherical coordinates (<i>not ready</i>)
MD parameters	
numeric_force	Marker to calculate the interatomic forces numerically (slow option, use for debugging only); no lines are required, only the keyword
Cohesive	Use MD module to construct cohesive energy of the cell
Displacement	Set the power N of the mean atomic displacement: $\langle [R(t)-R(t=0)]^N \rangle$ (the default option is $N=1$)
print_XYZ	Printout atomic coordinates in XYZ format at each output-timestep
print_V_XYZ	Printout atomic velocities in XYZ format at each output-timestep
print_MC_MD_energy	Printout energy transfer between MC and MC modules
print_LAMMPS	Printout atomic coordinates in LAMMPS ^o output format at each output-timestep

The marker print_LAMMPS requires specifying the units type used in LAMMPS^p: real, metal, lj, si, cgs, electron, micro, nano. The default option is “real” as the one closest to the units used in TREKIS-4.

The output grid index used to specify log-scale grid (logical: T=true, F=false). “F” sets the linear grid.

In the second line, instead of specifying the log-grid parameter, specifies the filename with the grid, the file is used instead; e.g. see file Rgrid.txt.

The start, end, and step in the third line must use the units used in the corresponding value being printed out: eV for energy; Å for coordinates, rad for angles.

Note #1: Only one grid per value (per coordinate; energy, angle) can be used; TREKIS-4 cannot create multiple grids for the same value.

^o <https://www.lammps.org/>

^p <https://docs.lammps.org/units.html>

Note #2: to printout coordinate-resolved spectra ‘Spectra_[axis]’ (specifying the coordinate grid), it must be preceded by the option ‘Energy’ to printout the energy spectrum (to specify the energy grid).

For example, to printout spectra at different depth along Z, the following option can be used:

```
Energy ! To print spectra out
T      ! use logscale (T=true, F=false)
1.0d0  1.0d3  1.0d-1      ! Start grid, End grid, Step coeff
Spectra_Z ! To print spectra out
F      ! use logscale (T=true, F=false)
-1.0d3  1.0d3  1.0d3      ! Start grid, End grid, Step coeff
```

Note that here we set the beginning and the end of the Z-grid such that only two sections of space are included: above Z=0 (top surface of the material) and below it, thus printing out electrons emitted and those inside the material.

Note #3: For two-dimensional printout (RL) grid along R and L can be set independently:

- a) keyword
- b) Type of the grid 1 (or filename 1)
- c) three numbers: star of grid 1, end of grid 1, step of grid 1
- d) Type of the grid 1 (or filename 1)
- e) three numbers: star of grid 1, end of grid 1, step of grid 1

The first grid corresponds to the first coordinate (R), the second one to the second one (L).

The cylindrical coordinates are always aligned with Z; currently it cannot be rotated.

See an example in Figure 6.

Some errors that TREKIS-4 can identify will be printed out in the file [Error_log.txt](#).

VI. Communication with program on-the-fly: [Communication.txt](#)

Some parameters may be changed during execution of the program, using the file [Communication.txt](#) created in each output folder during execution of the code. TREKIS-4 allows to change dynamically the duration of simulation, timestep, printout-timestep, number of OpenMP threads.

In the output data directory [OUTPUT_...](#) (see below) for each calculation (see below) the file [Communication.txt](#) created is checked by TREKIS-4 at each printout-timestep. Any changes in the file (user entered data) will be read and interpreted by the code. The accepted commands are:

time [x]	To change the duration of execution of the program. Time set in [fs]. E.g., «time 1000.0» sets the duration of simulation to 1000 fs, overwriting the value given in the input file
----------	--

dt [x]	Time-step in [fs]
Save_dt [x]	Printout timestep (how often output data are printed out) in [fs]
OMP [x]	Number of threads in OpenMP (integer)

VII. Output data: OUTPUT

The directory with output data are named according to the following format:

OUTPUT_[target]_[particle]_[energy]_[number of bunches]

where *[target]* is the target name specified in the file INPUT_DATA.txt

[particle] name of the first incident particle

[energy] energy of the first incident particle in [eV]

[number of bunches] number of pulses/bunches (only written if there are more than one bunch)

If the directory with the identical name already exists, a new one will be created with the added number at the end:“_vi”, where *i* is the next number of the folder.

For example:

OUTPUT_Silicon_Xe_ion_2187000000.00_2_bunches

After running TREKIS-4 again for the same parameters, the next directories will be created:
 OUTPUT_Silicon_Xe_ion_2187000000.00_2_bunches_v1, then
 OUTPUT_Silicon_Xe_ion_2187000000.00_2_bunches_v2 etc.

The output directory will contain the following data files:

1) Copy of the input files INPUT_DATA.txt and NUMERICAL_PARAMETERS.txt (for track records and reproducibility).

2) File **!OUTPUT_parameters.txt**, containing basic information about the target and incident particles, including that pulled out from the databases (such as ionization potential, core-hole decay times, etc.). The file will also contain information on the dynamical change of the parameters via **Communication.txt** file.

3) The **Communication.txt**, described above.

4) Directory **MFPs_and_Ranges_in_[matter]**, containing the mean free paths (MFP) of various particles in the given target, if the user set to print them out in the file **NUMERICAL_PARAMETERS.txt**. The MFP files will be named as follows: **OUTPUT_[CS]_[matter]_[element]_[particle]_[model]_[total].dat**,

[CS] – cross section type, e.g. IMPF for inelastic mean free path

[*matter*] – target material

[*element*] – chemical element in the compound (for core-shell scattering)

[*particle*] – incident particle, e.g. electron, Au, hole, etc.

[*model*] – model used, e.g. CDF etc.

[*total*] – total MFP (valence band is in the same file); this marker is absent for atomic shells

E.g.: OUTPUT_IMFPs_SiO2_O_Au_CDF.dat,

OUTPUT_IMFPs_SiO2_electron_CDF_delta_total.dat, etc.

In the files, the first line comments on what is in the columns; energy is in [eV] or [MeV] (for ions); then, MFP or Ranges for each shell are given.

TREKIS-4 also creates corresponding gnuplot⁹ shell-scripts (cmd-files for Windows, sh for Linux), which will be executed to create figures (if gnuplot is installed in your system).

5) File OUTPUT_DOS_of_[*matter*].dat, with DOS, if the user defined to print it out. The file contains 4 columns: E (eV), DOS (1/Atom/eV), k (1/m), me_eff (m_e), and corresponding gnuplot scripts.

6) Files OUTPUT_total_[*matter*].dat containing the data on total energies of various sorts of particles. The first two lines specify what is in the file:

Time (fs) Nph Ne Nh Np Eph Ee Eh_kin Eh_pot Ep Eat Etot,

Where

Time – is the time instant in [fs]

Nph – total number of photons in the simulation

Ne – total number of electrons in the simulation

Nh – total number of holes in the simulation, in all holes (core and VB); must coincide with the number of electrons without electron emission or absorbing boundaries

Np – total number of positrons in the simulation

Eph – total energies of photons in the simulation [eV]

Ee – total energies of electrons in the simulation [eV]

Eh_kin – kinetic energies of holes in the simulation [eV]

Eh_pot – potential energies of holes in the simulation [eV]

Ep – total energies of positrons in the simulation [eV]

Eat – total energy transferred to atoms in electron and hole scattering in the simulation [eV]

⁹ <http://www.gnuplot.info/>

Etot – total energy of all particles in the simulation [eV]; must conserve except for the time of energy deposition by external bunch and emission/absorption of particles at the boundaries.

The gnuplot scripts will create figures named in the following format:

`OUTPUT_total_[quantity]_in_[matter].[format]`, where

Quantity – numbers or energy of particles

Matter – materials

Format = eps, jpeg, gif, png, or pdf, specified in the input file

7) File `OUTPUT_total_above_cutoff_[matter].dat` containing the numbers and energies of the particles with energies above given cut-off (set in the file `NUMERICAL_PARAMETERS`); in the save format as `OUTPUT_total_[matter].dat`, except the absent of the column for atomic energy, since atoms are not included in MC.

This file may be used for cross check with the energy exchange between MC and MD. The following value should be conserved:

The last column in the file (total energy above cutoff E_{cutoff}) + sum of the last column in the file `OUTPUT_MCMD_energy_transfer.txt` (see below; sum, because not an integral but energy transferred at each step is saved in this file) = should be equal to the total energy in the system (last column in the file `OUTPUT_total_[matter].dat`). Mismatch between these values indicates some energy loss in the coupling between MC+MD; possibly due to MC grid and MD-atoms, as discussed above.

8) Set of files `OUTPUT_[particle]_spectrum_[matter].dat`, containing spectra of particles *[particle]* in the target *[matter]* if the user specified to printout them. All files are constructed as follows: first 2 lines are comments; then, 3 columns are:

First: time in [fs]; second: energy grid in [eV]; third: distribution in [1/eV].

Each timestep is separated from the block of the previous timestep by a single empty line.

9) Set of files `OUTPUT_[particle]_spectrum_[dim]_[axis].dat`, with spectra on the special grid along given axis: for particles *[particle]*, *[dim]* – dimensionality of the grid, *[axis]* – axis along which the grid is used. The energy grid is the same as in the file with total spectrum, above.

Files contain columns:

First: time in [fs],

Second: energy grid [eV],

Third and so on: distribution function in [1/eV] in the given section of space (according to the spatial grid indicated in the third line of the file, after the two comment lines).

Each next time step is separated from the previous block by an empty line.

10) Set of files `OUTPUT_[particle]_[quantity]_[dimension]_[index]_[matter].dat`, where:

[particle] – type of particle

[quantity] – density or energy of particles

[dimension] – 1d, 2d, 3d

[index] – axis along which the data are printed, see Table 1

[matter] - target material

The file have the same format as the spectra, but the grid is in [Å], distribution is in [1/Å³], and energy in [eV/Å³].

11) Set of files `OUTPUT_[particle]_velocity_theta_[matter].dat`, contains the distribution of the velocities by theta-angle (with respect to Z-axis), for particle type [particle] in the target [matter]. The columns are:

First: time in [fs]; Second: angular grid in [deg]; Third: distribution in [1/degree].

12) All MD files have the marker ‘MD’ in them: `OUTPUT_MD_[data].[ext]`.

If the user calculated cohesive energy (in NUMERICAL_PARAMETERS.txt), the files contain the marker:

`OUTPUT_MD_cohesive_energy.txt`

The file then contains the following columns:

First, Rescaling coefficient (relative size of the supercell).

Second, nearest neighbour distance [Å]

Third: Total energy in [eV/atom]

Columns 4-6: Pressure along 3 axis [GPa] (*untested!*)

In a regular MD run, the following files are created:

`OUTPUT_MD_energies.txt` – with 4 columns:

1) Time [fs]

2) Mean kinetic energy of atoms [eV/atom]

3) Mean potential energy of atoms [eV/atom]

4) Total energy of atoms [eV/atom] – should be conserved without thermostats and energy transfer from MC

[OUTPUT_MD_average_parameters.txt](#) – with columns:

- 1) Time [fs]
- 2) Mean kinetic temperature of atoms [K]
- 3) Mean pressure $P=1/3*(P_{xx}+P_{yy}+P_{zz})$ [GPa] (*untested!*)
- 4) Pressure tensor: P_{xx} P_{xy} P_{xz} P_{yx} P_{yy} P_{yz} P_{zx} P_{zy} P_{zz} [GPa] (*untested!*)

[OUTPUT_MD_mean_displacements.txt](#) – with columns:

- 1) Time [fs]
- 2) Mean atomic displacement from the original positions [\AA^N]

If it is a compound, next columns will contain mean displacements for each sort of atoms.

The degree N is defined by the user in the input file: $D=\langle[R(t)-R(t=0)]^N\rangle$. By default, $N=1$.

If the keyword “print_XYZ” was used in the input file, the XYZ^r file with output atomic coordinates at each printout-timestep will be created (can be visualized with standard software such as VMD or OVITO): [OUTPUT_MD_coordinates.xyz](#).

If the keyword “print_V_XYZ” is set in the input file, the atomic velocities are printed out: [OUTPUT_MD_velocities.xyz](#)

If the keyword «print_MC_MD_energy» is used in the input file, the file with the energy transfer between MC and MD is created: [OUTPUT_MCMD_energy_transfer.txt](#)

With the columns:

- 1) Time [fs]
- 2) Coordinate along the first axis (e.g., X, R, or R, in cartesian, spherical or cylindrical coordinates), set in the [Supercell.txt](#) (the axis undefined by the user will be marked with O).
- 3) Coordinate along the second axis (e.g., Y, Theta, or L, in cartesian, spherical or cylindrical coordinates), set in [Supercell.txt](#)
- 4) Coordinate along the third axis (e.g., Z, Phi, or Phi, in cartesian, spherical or cylindrical coordinates), set in [Supercell.txt](#)
- 5) Energy transferred to atoms in elastic scattering of electrons in [eV], at this grid point at this timestep.
- 6) Energy transferred to atoms in elastic scattering of VB holes in [eV], at this grid point at this timestep.
- 7) Energy transferred to atoms from electrons “disappearing” when following below E_{cutoff} in [eV], at this grid point at this timestep.
- 8) Energy transferred to atoms from VB holes “disappearing” when following below E_{cutoff} in [eV], at this grid point at this timestep. Includes kinetic energy of the hole, and its potential energy equal to the bandgap (E_{gap}).

This file allows to cross check the energy conservation in the transfer of energy between MC and MD:

^r https://en.wikipedia.org/wiki/XYZ_file_format

A) Total energy in MC and MD should be conserved, as discussed above, see description of the file `OUTPUT_total_above_cutoff_[matter].dat` above.

B) Energy, transferred to MD supercell (the sum in the last column) should coincide with the increase of the total energy in the MD file `OUTPUT_MD_energies.txt` (last column, minus its first value, to trace the relative energy or energy change) – within the MD precision.

Those should work without the thermostats, and the easiest way is to trace the total values, without the spatial resolution in the supercell, for coinciding MC and MD simulation boxes.

VIII. References

- [1] N. A. Medvedev, R. A. Rymzhanov, and A. E. Volkov, *Time-Resolved Electron Kinetics in Swift Heavy Ion Irradiated Solids*, J. Phys. D. Appl. Phys. **48**, 355303 (2015).
- [2] R. A. Rymzhanov, N. A. Medvedev, and A. E. Volkov, *Effects of Model Approximations for Electron, Hole, and Photon Transport in Swift Heavy Ion Tracks*, Nucl. Instruments Methods Phys. Res. Sect. B Beam Interact. with Mater. Atoms **388**, 41 (2016).
- [3] N. Medvedev, F. Akhmetov, R. A. Rymzhanov, R. Voronkov, and A. E. Volkov, *Modeling Time-Resolved Kinetics in Solids Induced by Extreme Electronic Excitation*, Adv. Theory Simulations **5**, 2200091 (2022).
- [4] R. H. Ritchie and A. Howie, *Electron Excitation and the Optical Potential in Electron Microscopy*, Philos. Mag. **36**, 463 (1977).
- [5] R. Garcia-Molina, I. Abril, I. Kyriakou, and D. Emfietzoglou, *Inelastic Scattering and Energy Loss of Swift Electron Beams in Biologically Relevant Materials*, Surf. Interface Anal. **49**, 11 (2017).
- [6] N. Medvedev and A. E. Volkov, *Analytically Solvable Model of Scattering of Relativistic Charged Particles in Solids*, J. Phys. D. Appl. Phys. **53**, 235302 (2020).
- [7] Y.-K. Kim, J. P. Santos, and F. Parente, *Extension of the Binary-Encounter-Dipole Model to Relativistic Incident Electrons*, Phys. Rev. A **62**, 52710 (2000).
- [8] T. M. Jenkins, W. R. Nelson, and A. Rindi, *Monte Carlo Transport of Electrons and Photons* (Springer US, Boston, MA, 1988).
- [9] D. E. Cullen, EPICS2017: Electron Photon Interaction Cross Sections: W-Nds.Iaea.Org/Epics/, 2018.
- [10] F. Salvat and M. Fern, *PENELOPE-2014 – A Code System for Monte Carlo Simulation of Electron and Photon Transport*, 2015th ed. (NUCLEAR ENERGY AGENCY, Organisation for Economic Co-operation and Development, Barcelona, Spain, 2015).
- [11] D. Bank, *PENELOPE – A Code System for Monte Carlo Simulation of Electron and Photon Transport A Code System for Monte Carlo* (2001).
- [12] R. Christensen, *The Dosimetry of Ionizing Radiation, Volume 1 Edited by Kenneth R. Kase, Bengt E. Bjärngard, and F. H. Attix*, Vol. 5 (1990).
- [13] B. Ziaja, R. A. London, and J. Hajdu, *Unified Model of Secondary Electron Cascades in Diamond*,

J. Appl. Phys. **97**, 064905 (2005).

- [14] R. A. Rymzhanov, N. A. Medvedev, and A. E. Volkov, *Electron Emission from Silicon and Germanium after Swift Heavy Ion Impact*, Phys. Status Solidi B **252**, 159 (2015).
- [15] L. Verlet, *Computer “Experiments” on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules*, Phys. Rev. **159**, 98 (1967).
- [16] C. J. Fennell and J. D. Gezelter, *Is the Ewald Summation Still Necessary? Pairwise Alternatives to the Accepted Standard for Long-Range Electrostatics*, J. Chem. Phys. **124**, 234104 (2006).
- [17] O. Keski-Rahkonen and M. O. Krause, *Total and Partial Atomic-Level Widths*, At. Data Nucl. Data Tables **14**, 139 (1974).
- [18] N. Medvedev, *Modeling Ultrafast Electronic Processes in Solids Excited by Femtosecond VUV-XUV Laser Pulse*, AIP Conf. Proc. **582**, 582 (2012).
- [19] C. Bouchard and J. D. Carette, *The Surface Potential Barrier in Secondary Emission from Semiconductors*, Surf. Sci. **100**, 251 (1980).
- [20] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, *Molecular Dynamics with Coupling to an External Bath*, J. Chem. Phys. **81**, 3684 (1984).
- [21] N. Medvedev, F. Akhmetov, R. A. Rymzhanov, R. Voronkov, and A. E. Volkov, *Modeling Time-Resolved Kinetics in Solids Induced by Extreme Electronic Excitation*, Adv. Theory Simulations **5**, 2200091 (2022).
- [22] S. Namilaie, D. M. Nicholson, P. K. V. V. Nukala, C. Y. Gao, Y. N. Osetsky, and D. J. Keffer, *Absorbing Boundary Conditions for Molecular Dynamics and Multiscale Modeling*, Phys. Rev. B **76**, 144111 (2007).
- [23] X. W. Zhou, D. K. Ward, J. E. Martin, F. B. Van Swol, J. L. Cruz-Campa, and D. Zubia, *Stillinger-Weber Potential for the II-VI Elements Zn-Cd-Hg-S-Se-Te*, Phys. Rev. B - Condens. Matter Mater. Phys. **88**, 085309 (2013).
- [24] Z. Jurek, S.-K. Son, B. Ziaja, and R. Santra, *XMDYN and XATOM: Versatile Simulation Tools for Quantitative Modeling of X-Ray Free-Electron Laser Induced Dynamics of Matter*, J. Appl. Crystallogr. **49**, 1048 (2016).
- [25] M. Azzolini, M. Angelucci, R. Cimino, R. Larciprete, N. M. Pugno, S. Taioli, and M. Dapor, *Secondary Electron Emission and Yield Spectra of Metals from Monte Carlo Simulations and Experiments*, J. Phys. Condens. Matter **31**, 55901 (2019).