Details of input in Adaptive time, Electronic stopping and Non-adiabatic processes

1. Adaptive time

```
adaptive_time = .true.
adapt_tstep_interval = 1
adapt_tmin = 1.0E-07
adapt_tmax = 1.0E-01
adapt_xmax = 1.0e-3
adapt_emax = 50.0
```

The adaptive time steps can be switched off by writing *adaptive_time* = *.false.*, and then the following five lines of input will not be used. In order to apply this algorithm only after a certain number of time steps, the particular value shWhen a simulation has to be restarted from a previous one then the *eph_md_last_step* input keyword can be set with value of previous number of steps already run to keep track of the number steps. By default it is equal to 0. The output data for energy transfer and average electronic temperature can be taken at some desired frequency which is specified through input keyword *eph_freq_Tout* and keeping it equal to the frequency of the original thermo-output (in thermo.log file) can be useful. The values of electronic parameters for the mesh may be needed only at some larger intervals (or may be the last one only) because this file will be generally large, so *eph_freq_mesh_Tout* input keyword can be set accordingly. If a restarted simulation is being run and the value in *eph_md_last_step* input keyword is last step of previous simulation (say, N), then the output file for the mesh parameters shows values for the Nth + present_step. The name of file for getting this mesh parameters output is provided through input keyword *eph_Toutfile*.ould be specified to *adapt_tstep_interval*. A value of 1 means applying the algorithm at every time step and a value of 10 means applying it after every 10 time steps. The next two input lines are given to provide the maximum (*adapt_tmax*) and minimum (*adapt_tmin*) limits a time step can take. The current time step is determined primarily based on the value of maximum possible displacement for an atom that is provided in *adapt_xmax*. A value of 0.001 here implies that an atom moves maximum of 0.001 Å in a time step. If the allowed maximum displacement is larger compared the displacement possible with the current time step, then the time step is only modified based on the value of the maximum energy that can change for an atom in a time step. This is provided through the input *adapt_emax*. So, a reasonable value of *adapt_xmax* during the cascade simulation is sufficient to get a reasonable time step.

2. Electronic stopping based on stopping data from SRIM-2013

```
electronic_stopping = .true.
eel_cut = 1.0
estop_filename = 'stopping-data-file'
```

The execution of this part can be switched off by writing *electronic_stopping* = *.true.*.When it is executed the supposed electronic energy losses of the atoms are subtracted from the atoms as frictional forces depending on the stopping data from SRIM-2013 that is provided. The energy is subtracted from an atom till its energy is less than a cut-off value, which is provided through the input *eel_cut*. So, a value of 1.0 here means when an atom has energy higher than 1.0 eV electronic energy loss is calculated for it and subtracted from its energy. The stopping data is provided in a text file, the name of which is mentioned by using the input keyword *estop_filename*. The data for stopping in the text file should be provided such that it covers the value of energy given in *eel_cut*. For multiple elements in a target material, the stopping powers of the elements should be in arranged in columns in order as they appear in the 'species' input. The text file must contain data in the following format:

1st line – Any useful information
2nd line – number of rows of data points, *N*
3rd line – energy unit (eV) and symbols of elements (must be) in order as in input file
4th line onwards – energy and electronic stopping values (eV/Ang) like
4th line – E_1 $(dE/dx)_1$(element 1) $(dE/dx)_1$(element 2) $(dE/dx)_1$(element 3) ……
5th line – E_2 $(dE/dx)_2$(element 1) $(dE/dx)_2$(element 2) $(dE/dx)_2$(element 3) ……
.. ..
Nth line – E_N $(dE/dx)_N$(element 1) $(dE/dx)_N$(element 2) $(dE/dx)_N$(element 3) ……

3. Non-adiabatic processes

```
nonadiabatic_processes = .true.
eph_fdm_option = 1
eph_friction_option = 1
eph_random_option = 1
eph_betafile = 'beta-rho-data-file'
eph_Tinfile = 'input-fdm-electron-mesh-file'
box_limits = 0.0 28.0 0.0 28.0 0.0 28.0
eph_rho_e = 1.0
eph_C_e = 1.0E-06
eph_kappa_e = 0.8
eph_Ti_e = 5.0
eph_gsx = 4
eph_gsy = 4
eph_gsz = 4
eph_fdm_steps = 1
eph_md_last_step = 0
eph_md_prev_time = 0.0
eph_E_prev_time = 0.0
eph_freq_Tout = 100
eph_freq_mesh_Tout = 100000
eph_Toutfile = 'output-electron-mesh-fdm-file'
```

This calculation option can be switched off by writing *nonadiabatic_processes* = *.true.*, then none of the following lines of input will be used. The electron-phonon coupling that governs the energy dissipation during a radiation cascade event is described by the e-ph model [PhysRevLett.120.185501, PhysRevB.99.174302]. This model is now implemented in TurboGAP. The output data for energy transfer and average electronic temperature can be taken at some desired frequency which is specified through input keyword *eph_freq_Tout* and keeping it equal to the frequency of the original thermo-output (in thermo.log file) can be useful. The values of electronic parameters for the mesh may be needed only at some larger intervals (or may be the last one only) because this file will be generally large, so *eph_freq_mesh_Tout* input keyword can be set accordingly. When a simulation has to be restarted from a previous one then the *eph_md_last_step* input keyword can be set with value of previous number of steps already run to keep track of the number steps. By default it is equal to 0. Likewise, the input keywords *eph_md_prev_time* and *eph_E_prev_time* are then set to the MD time and the net cumulative energy transferred values corresponding to the last step of the previous run. If a restarted simulation is being run and the value in *eph_md_last_step* input keyword is set to the value of last step of previous simulation (say, N), then the output file for the mesh parameters shows values for the Nth + present_step. The MD time and cumulative net energies in the output file for energies and electronic temperature (which is fixed by the name as "eph-EnergySharingData.txt") are also updated by addition to their last values from the previous run. Note, the actual MD time in the thermo.log file has to be manually added from previous data in case of restarted runs. The name of the file for getting this mesh parameters output is provided through input keyword *eph_Toutfile*. By default its value is ‘NULL’. The data for electron-ion coupling parameter obtained from TD-DFT simulations is provided in a text file using the input keyword *eph_betafile*. By default its value is ‘NULL’. The input keyword *eph_Tinfile* is used to provide the text file name where several of the input parameters are provided. By default its value is ‘NULL’. If this file is not provided then the input keywords in the next nine lines of input are required. Either the data provided through text file in *eph_Tinfile* or the following nine input keywords will define the mesh for electronic heat bath along with the required parameters. The format of this parameter file is as follows:

1st line – any comments
2nd line – any comments
3rd line – any comments
4th line – gsx gsy gsz number_of_fdm_steps
5th line – mesh_xlimit_low mesh_xlimit_high
6th line – mesh_ylimit_low mesh_ylimit_high
7th line – mesh_zlimit_low mesh_zlimit_high
8th line – i j k T_e S_e rho_e C_e K_e flag T_dyn_flag
9th line – values corresponding to column headers
10th line – values corresponding to column headers

….
….
Values are provided for the full mesh.

Alternative to providing the above data, the mesh can also be constructed for solving the heat diffusion equation with the following inputs:
box_limits = 0.0 28.0 0.0 28.0 0.0 28.0 make the boundaries of the mesh where the values are for xlow, xhi, ylow, yhi, zlow and zhi, respectively;
eph_rho_e = 1.0 value of rho_e as in data file;
eph_C_e = 1.0E-06 value of C_e as in data file;
eph_kappa_e = 0.8 value of K_e as in data file;
eph_Ti_e = 5.0 value of T_e as in data file;
eph_gsx = 4 number of small mesh boxes along x;
eph_gsy = 4 number of small mesh boxes along y;
eph_gsz = 4 number of small mesh boxes along z;
eph_fdm_steps = 1 same as the number_of_fdm_steps which is used to find the time step for solving heat diffusion equation.

By default the full electron-phonon coupling along with electronic friction is implemented. But the user has some options for implementing only the parts of the model if required. There are three input keywords by which this can be done, viz., *eph_fdm_option*, *eph_friction_option, eph_random_option*. The finite difference method for solving heat diffusion equation for the electrons is governed by *eph_fdm_option*. The electronic energy loss from the atoms is governed by *eph_friction_option* and the mutual transfer of energy between the electrons and atoms (lattice phonon) through random forces is governed by *eph_random_option*. All of these three input keywords can take two values as input either 0 or 1, where 0 and 1 imply switch on and switch off, respectively, of the process. If the *eph_fdm_option* is kept as ‘0’, then the updating of electronic temperature by solving the heat diffusion equation will not be done. If *eph_friction_option* is kept ‘0’ the frictional forces due to ion-electron interactions will not be calculated and if *eph_random_option* is kept ‘0’ the random forces of electron-phonon coupling will not be calculated. Since by default the full model is implemented, all these input keywords have default values set to ‘1’. So providing these inputs may be skipped unless any one of these needs to be turned off. Note that if both *eph_friction_option* and *eph_random_option* are turned off, then the *eph_fdm_option* should as well be turned off.