

Illustrations of Some Implemented Examples with Adaptive-Timing, Electronic-Stopping and Electron-Phonon Coupling Model in TurboGAP with Details of Input and Validations

Uttiyoarnab Saha

Department of Applied Physics
Aalto University, Finland

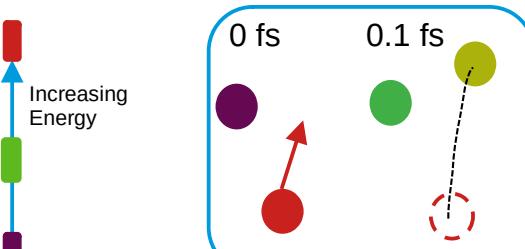
<https://github.com/mcaroba/turbogap/blob/master/docs/stopping/Illustrations-Details-of-Inputs-Validations.pdf>

To select a time step (unit of time is fs):

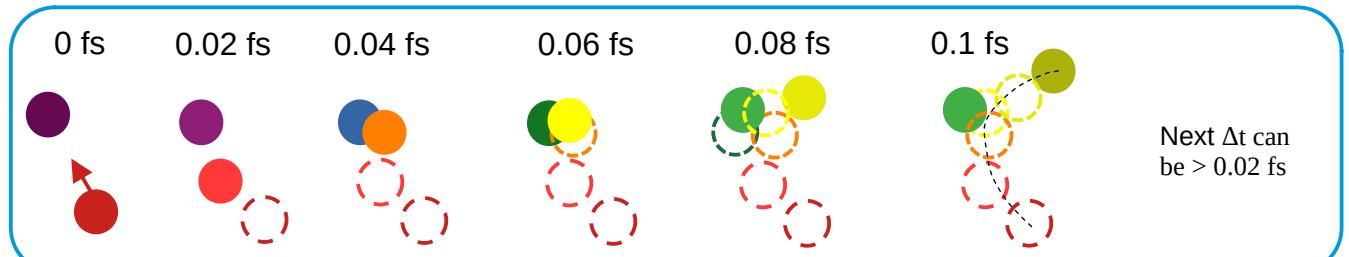
The basic relation $\Delta x = v\Delta t$ is suitable in normal situations. In radiation cascades the atoms experience very strong collisions where this time step could be very long.

Then, we will actually need $\Delta x \leq 0.0001 \text{ \AA}$ to see the exact trajectories and energy sharing by the atoms during cascade progress.

We use a x_{\max} criterion to allow maximum possible distance an atom can move in a time step and also a e_{\max} criterion to allow maximum possible change in K.E of an atom.



Fixed time step



Adaptive time step – with smaller Δt at high E and larger time steps at low E

The displacement of atoms in cascades take place through binary elastic collisions. The nuclear energy loss is important in damage calculations.

But there is electronic energy loss too, at all energies.

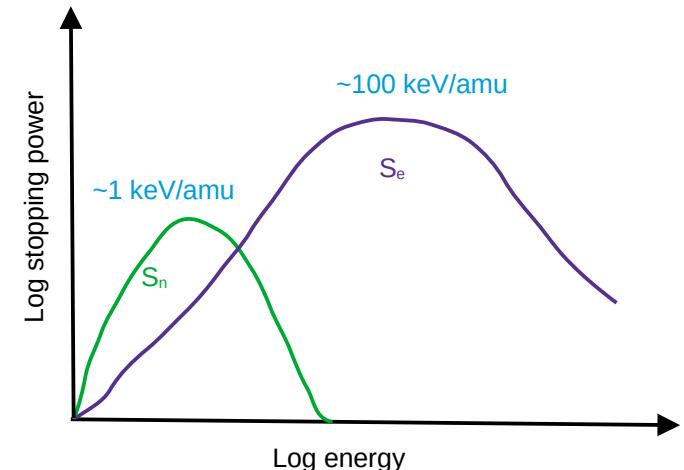
Only damage energy part of K.E of PKA is responsible for damage creation.

The damage energy depends on K.E of PKA and also on the types of the elements (PKA and the target) through the characteristic electronic energy loss.

In the past, we used to do MD with $T_{\text{dam}} = E_{\text{PKA}} - E(\text{el})$ as the PKA energy, calculated analytically from energy partition theory or from the data from SRIM software.

But advances allow for real-time EEL calculations while simulating a cascade in MD. There are a few approaches possible.

- S_e (in units eV/Å) is obtained as a function of E_{ion} from SRIM-2013.
- User provides a lower cut off energy (few tens of eV, usually) and the name of stopping data file.



Modify the forces as

$$\vec{F} = \vec{F}^0 - S_e \frac{\vec{v}}{v}$$

Adaptive time

For the user ::

Simple input keywords starting with the word 'adapt':

```
adaptive_time = .true.  
adapt_time_groupID = all  
adapt_tstep_interval = 1  
adapt_tmin = 1.0e-07  
adapt_tmax = 0.1  
adapt_xmax = 1.0e-4  
adapt_emax = 50.0
```

Electronic stopping

For the user ::

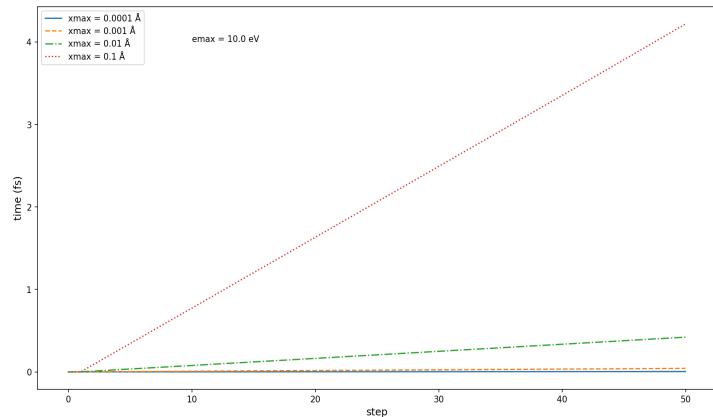
Simple input keywords

```
electronic_stopping = .true.  
eel_groupID = all  
eel_cut = 5.0  
eel_freq_out = 10  
estop_filename = 'stopping-data-file.txt'
```

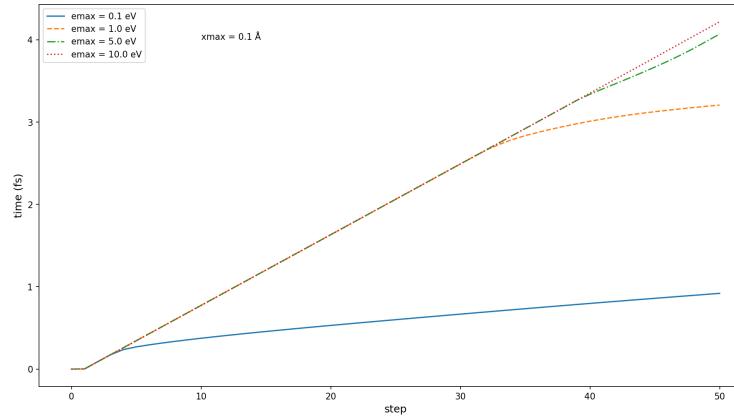
[Note: all illustrations here are shown by using a general purpose Si-GAP potential without short distance repulsive stiffening.]

Sensitivity of adaptive timing to maximum distance and energy criteria

Sensitivity to xmax



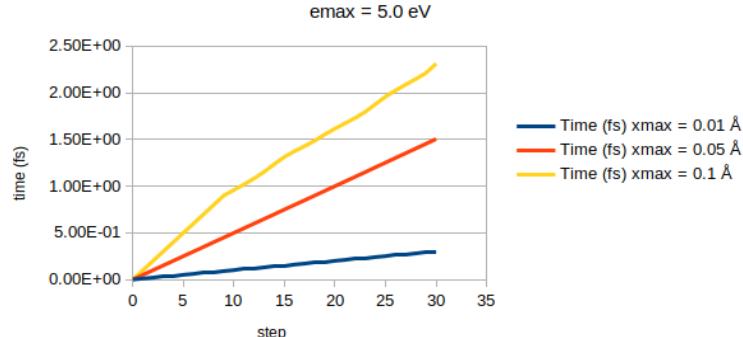
Sensitivity to emax



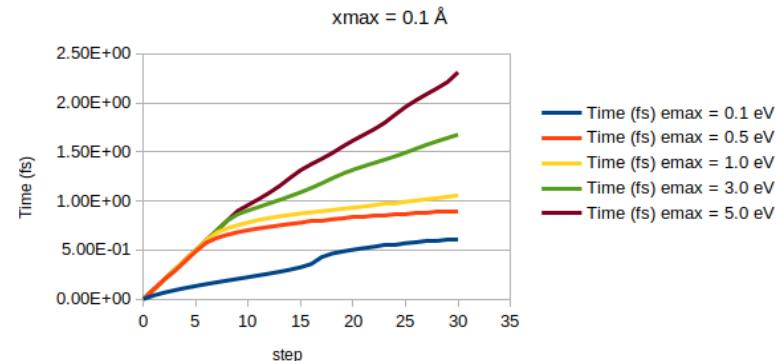
Sensitivity of time steps to xmax and emax in a binary collision simulation

Two Si atoms at $r_1 = (10.0, 10.0, 10.0)$ and $r_2 = (13.0, 10.0, 10.0)$ with $v_{1x} = 1.0 \text{ \AA/fs}$, $v_{1y} = 0.0$, $v_{1z} = 0.0$ and $v_{2x} = -1.0 \text{ \AA/fs}$, $v_{2y} = 0.0$, $v_{2z} = 0.0$ using Si (not stiffened) GAP potential.

Binary Collision using TurboGAP



Binary Collision using TurboGAP



Some Simulations with Fixed and Adaptive time steps

Silicon with 4096 atoms – one atom is given ~100 eV energy

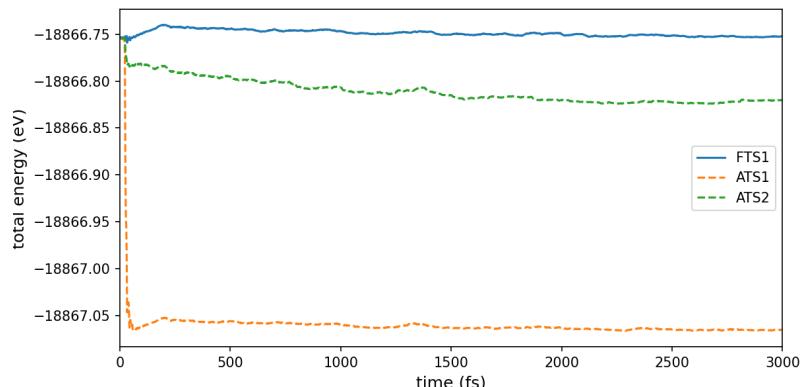
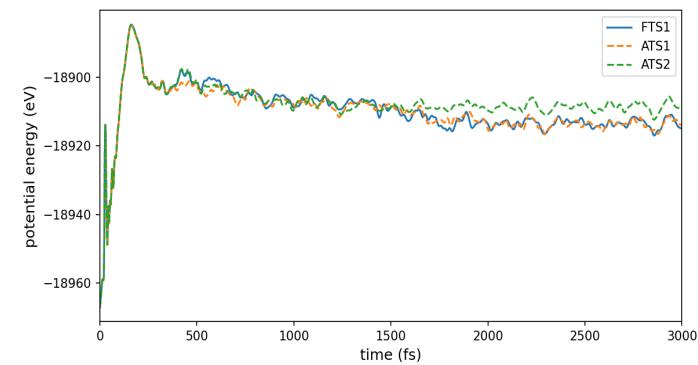
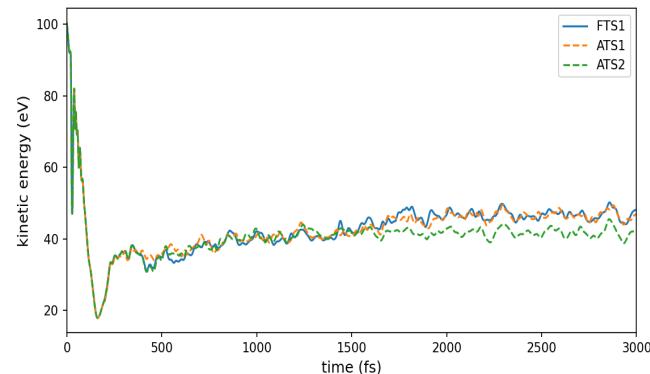
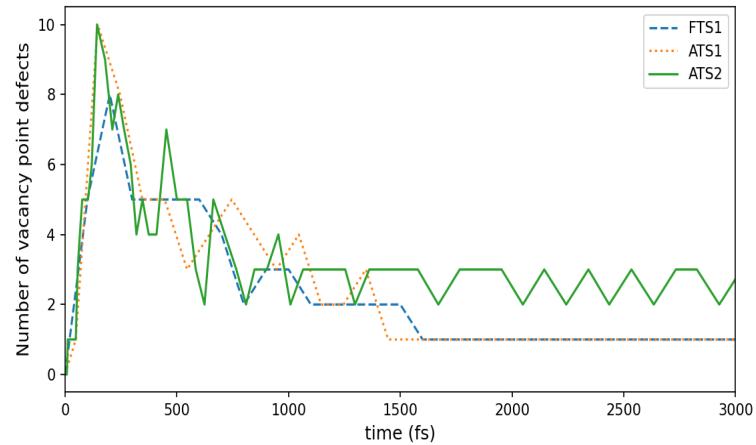
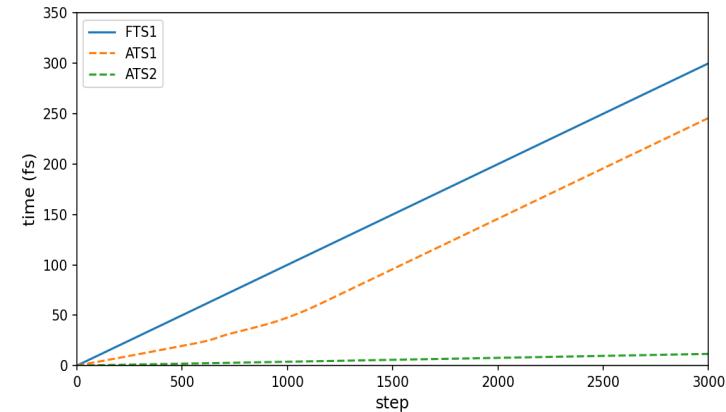
Fixed time step
= 0.1 fs → **FTS1**

ATS1

adapt_tstep_interval = 1
adapt_tmin = 1.0E-07
adapt_tmax = 1.0E-01
adapt_xmax = **1.0E-02**
adapt_emax = 30.0

ATS2

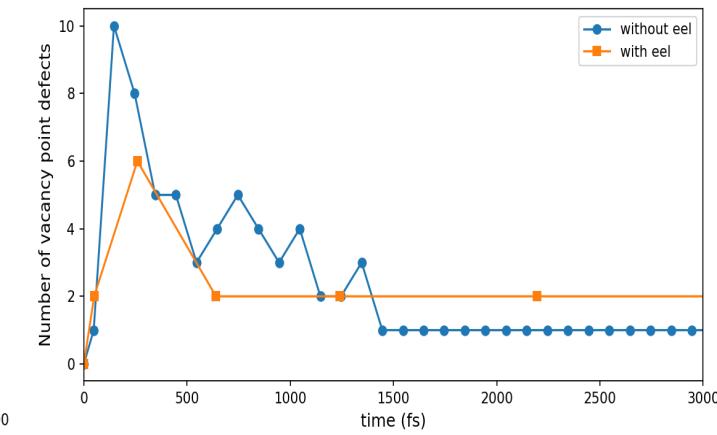
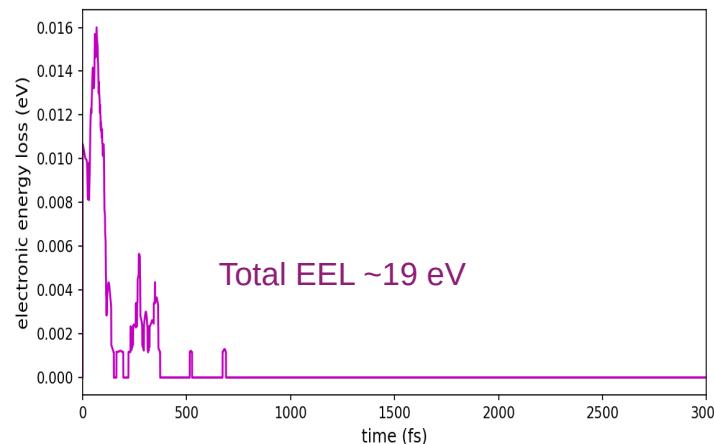
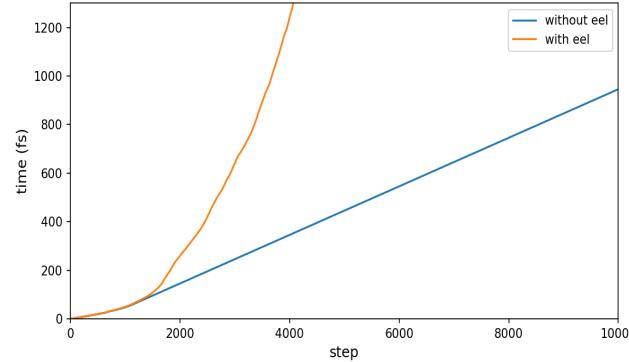
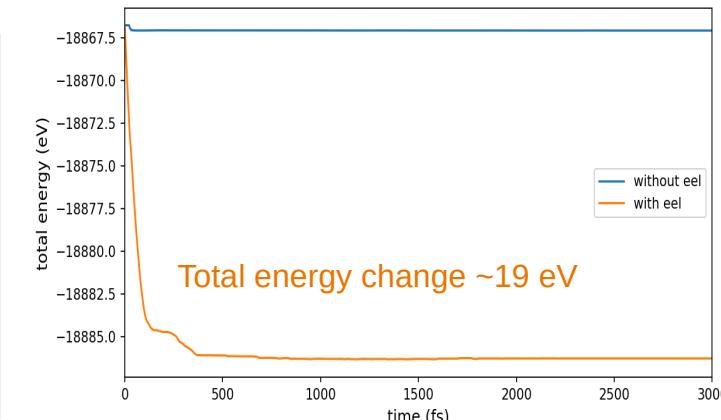
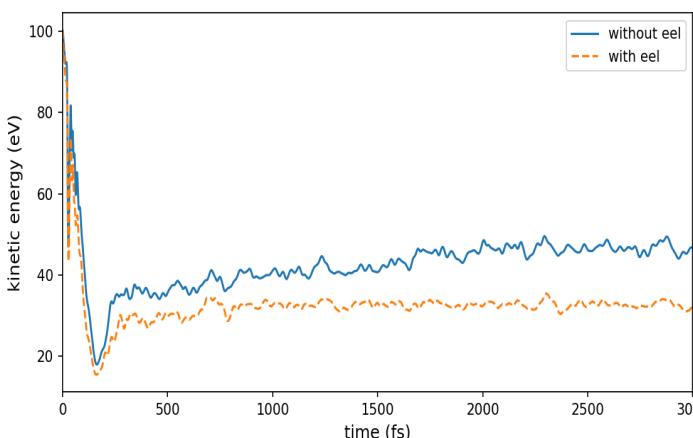
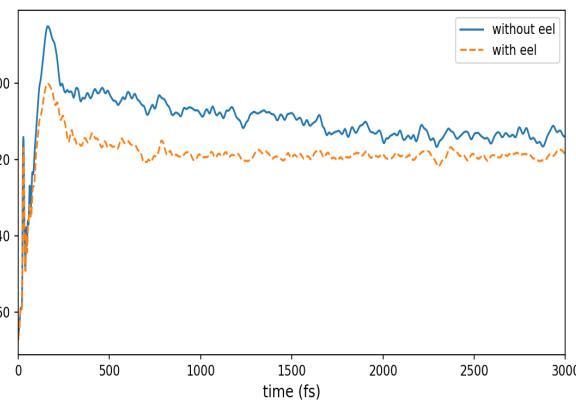
adapt_tstep_interval = 1
adapt_tmin = 1.0E-07
adapt_tmax = 1.0E-01
adapt_xmax = **1.0E-03**
adapt_emax = 30.0



Simulations with Electronic Stopping using stopping data from SRIM-2013

Silicon with 4096 atoms – one atom is given ~100 eV energy

eel_cut = 10.0 eV,
 tstep_interval = 1,
 tmin = 1.0E-07 fs, tmax = 1.0 fs,
 xmax = 0.01Å, emax = 30.0 eV



If we do not account EEL, then we allow more defect creation and also more defect annealing. The defects may also be morphologically different from reality. [Note: the numbers of defects or exact values of the quantities shown here are not important, only the general trends are shown by using a general purpose potential without short distance repulsive stiffening.]

Electronic stopping and electron-phonon coupling

During the initial non-equilibrium stages of a radiation event, electronic stopping is dominant and at the later stages to equilibrium between the lattice and the electrons, the coupling between electrons and lattice phonons become significant.

Both aspects of energy dissipation between the atoms and electrons are important.

The previous method described simply acts as a friction term

- to only reduce the energy of the atoms,
- to arbitrarily low values determined by the cut off energy specified and
- assumes that electronic stopping at low velocities is linear in velocity.

However,

- electronic stopping and e-ph coupling both must be accounted within same theory,
- a Langevin drag force can act between atoms and electronic heat bath establishing thermal equilibrium, not only just reducing the atom energy to arbitrary values,
- fixing the low energy cut off to arbitrary values may not be correct,
- at low velocities the electronic stopping power is not linear in velocity and
- the drag force connected to a stochastic force through fluctuation-dissipation theorem within the Langevin dynamical theory can model the phenomena more accurately.

Electronic stopping and electron-phonon coupling

The equation of motion with Langevin spatial correlations is

$$\mathbf{f}_I = -\nabla_I U - \boldsymbol{\sigma}_I + \boldsymbol{\eta}_I$$

where $\boldsymbol{\sigma}_I = \sum_J B_{IJ} \mathbf{v}_J$ and $\boldsymbol{\eta}_I = \sum_J W_{IJ} \boldsymbol{\xi}_J$, are the friction and random forces.

The matrix W_{IJ} produces spatial correlations of the random forces $\boldsymbol{\eta}_I$ starting from uncorrelated random vectors $\boldsymbol{\xi}_J$. It is defined as follows:

$$W_{IJ} = -\alpha_J \frac{\rho_I(r_{IJ})}{\bar{\rho}_J} \mathbf{e}_{IJ} \mathbf{e}_{IJ}^T (I \neq J)$$

$$W_{II} = \alpha_I \sum_{K \neq I} \frac{\rho_K(r_{IK})}{\bar{\rho}_I} \mathbf{e}_{IK} \mathbf{e}_{IK}^T (I = J)$$

[PRL 120, 185501 (2018), PRB 99, 174302 (2019)]

The friction (drag) forces are obtained according to the fluctuation-dissipation theorem as follows:

$$B_{IJ} = \sum_K W_{IK} W_{JK}^T$$

The electron-ion coupling parameter is estimated using time-dependent DFT simulations and provided as input databases for the simulations using TurboGAP.

Electronic temperature: $C_e(T_e) \frac{\partial T_e}{\partial t} = \nabla \cdot (\kappa_e(T_e) \nabla T_e) + Q_{e-i} + S_{ext}$

Q_{e-i} is a local source term representing energy sharing between electrons and atoms, S_{ext} is external source.

The inputs to be given to implement this electron-phonon coupling model are described here. There are two choices:

<1> less input keywords with accompanying text file for electronic temperature mesh and

<2> using a few more input keywords, but no additional file.

If <1> method is followed, then the any common values provided in input file will not be considered.

For some calculations such as with T-dependent electronic parameters and / or with external source term, only <1> should be used.

Choice <1>

```
nonadiabatic_processes = .true.  
eph_groupID = all  
eph_fdm_option = 1  
eph_friction_option = 1  
eph_random_option = 1  
eph_betafile = 'Si_PRB2021_constant.beta'  
eph_Tinfile = 'T_input.fdm'  
eph_md_last_step = 0  
eph_md_prev_time = 0.0  
eph_E_prev_time = 0.0  
eph_freq_Tout = 10  
eph_freq_mesh_Tout = 1000  
eph_Toutfile = 'T-Si-Ta0Te50.out'
```

For Choice <1> Sample T_input.fdm

```
#  
# 3 lines of comments  
#  
2 2 2 1  
0.0 16.29  
0.0 16.29  
0.0 16.29  
ij k T_e S_e rho_e C_e K_e flag T_dyn_flag  
1 1 1 50.0 1.0 1.0 3.5E-06 0.1248 0 0  
2 1 1 50.0 1.0 1.0 3.5E-06 0.1248 0 0  
1 2 1 50.0 1.0 1.0 3.5E-06 0.1248 0 0  
2 2 1 50.0 1.0 1.0 3.5E-06 0.1248 0 0  
1 1 2 50.0 1.0 1.0 3.5E-06 0.1248 0 0  
2 1 2 50.0 1.0 1.0 3.5E-06 0.1248 0 0  
1 2 2 50.0 1.0 1.0 3.5E-06 0.1248 0 0  
2 2 2 50.0 1.0 1.0 3.5E-06 0.1248 0 0
```

If any 'flag' = 1, source S_e will be considered.

If any 'T_dyn_flag' = 1, data for C_e and K_e as functions of T_e should be provided through separate file named as "Te-dependent_e-parameters.txt"

Choice <2>

```
nonadiabatic_processes = .true.  
eph_groupID = all  
eph_fdm_option = 1  
eph_friction_option = 1  
eph_random_option = 1  
eph_betafile = 'Si_PRB2021_constant.beta'  
eph_box_limits = 0.0 16.29 0.0 16.29 0.0 16.29  
eph_rho_e = 1.0  
eph_C_e = 3.5E-06  
eph_kappa_e = 0.1248  
eph_Ti_e = 50.0  
eph_gsx = 2  
eph_gsy = 2  
eph_gsz = 2  
eph_fdm_steps = 1  
eph_md_last_step = 0  
eph_md_prev_time = 0.0  
eph_E_prev_time = 0.0  
eph_freq_Tout = 10  
eph_freq_mesh_Tout = 1000  
eph_Toutfile = 'T-Si-Ta0Te50.out'
```

Some simulations by using the e-ph coupling model in TurboGAP are shown in the following few pages.

Variations of atomic and electronic temperatures

216 Si atoms in a box of dimensions
16.29 Å on each side.

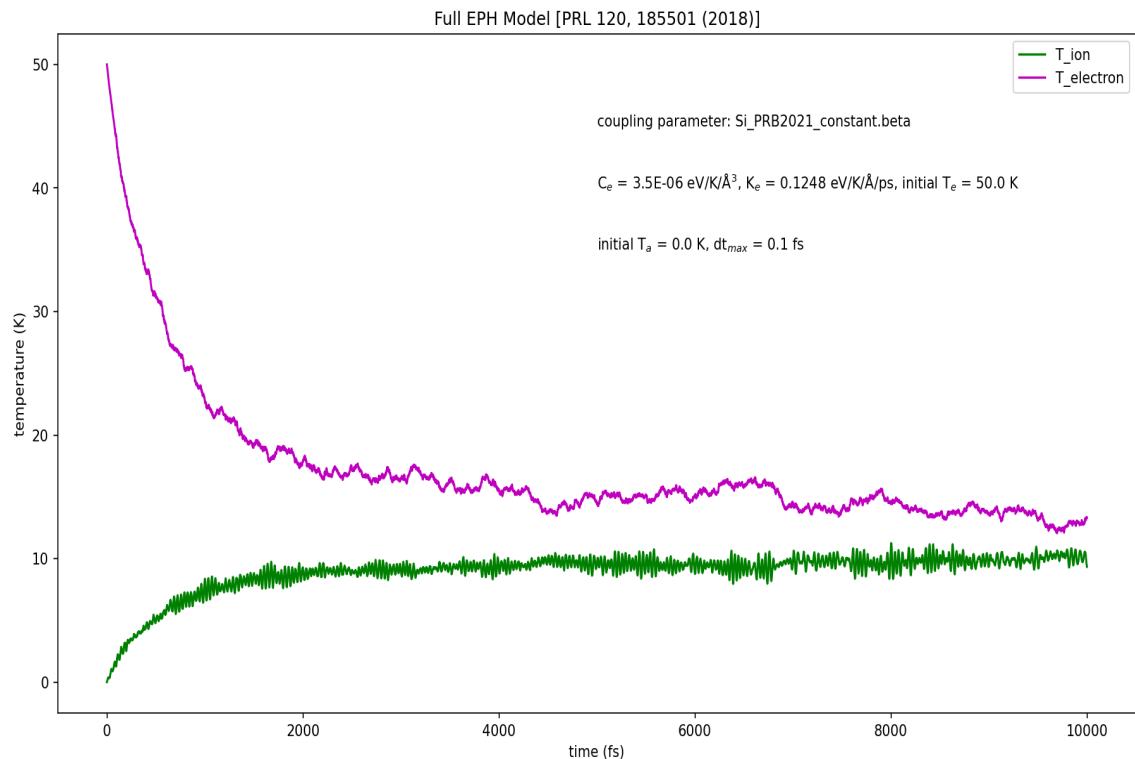
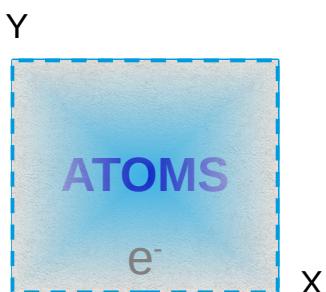
Initial atom temperature = 0.0 K

Initial electronic temperature = 50.0 K

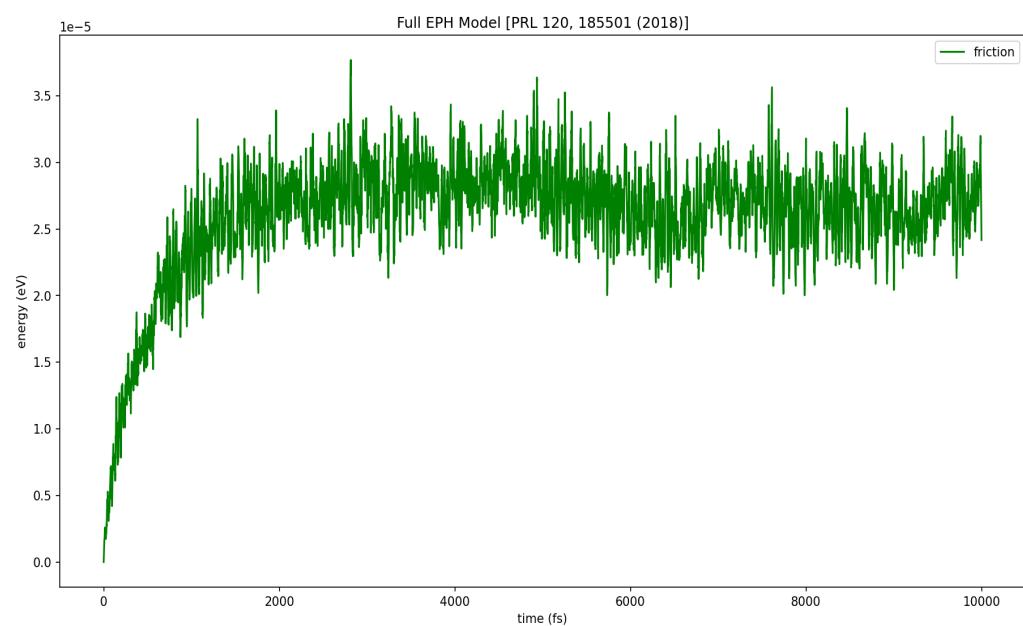
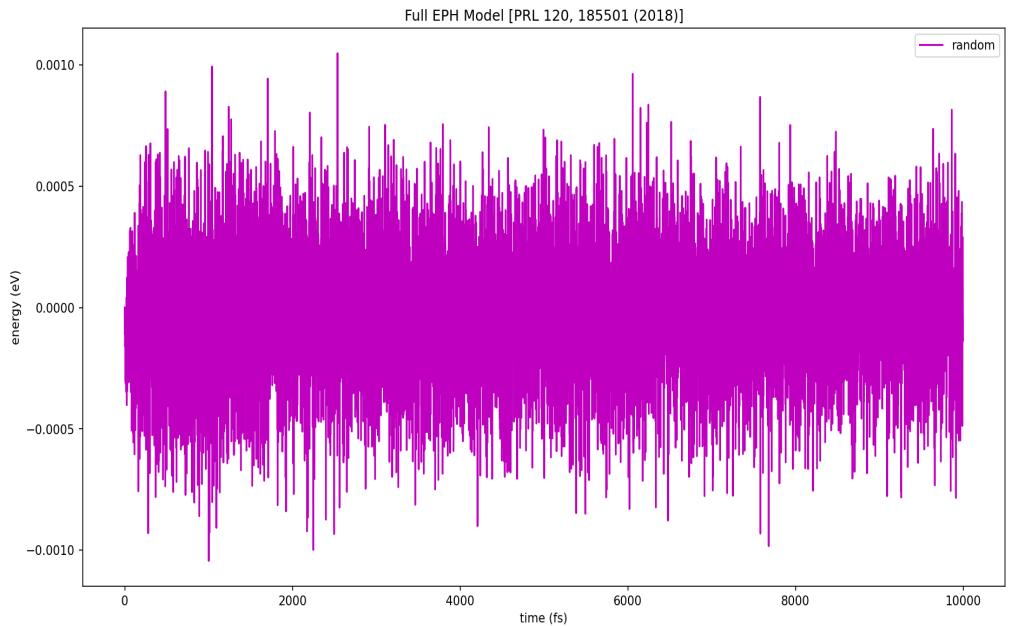
Coupling parameter – constant case
T. Jarrin et al. Phys. Rev. B 104, 195203

Electronic parameters:
 $C_e = 3.5 \times 10^{-6} \text{ eV/K/Å}^3$
 $K_e = 0.1248 \text{ eV/K/Å/ps}$

Electronic mesh:
Same size as atom box.
 $n_x = 1, n_y = 1, n_z = 1$.

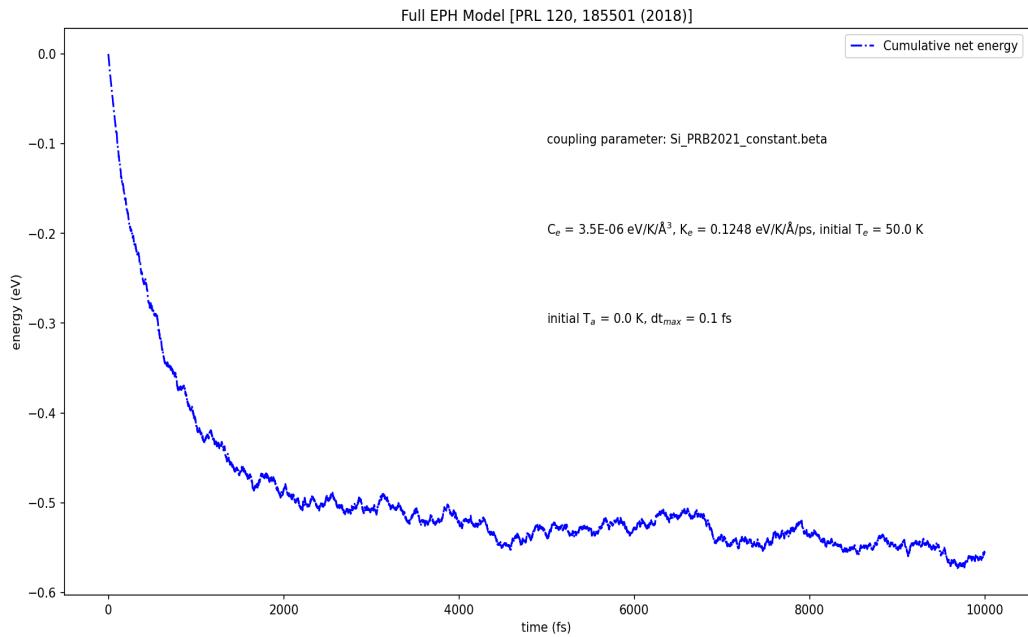


Variations of energies

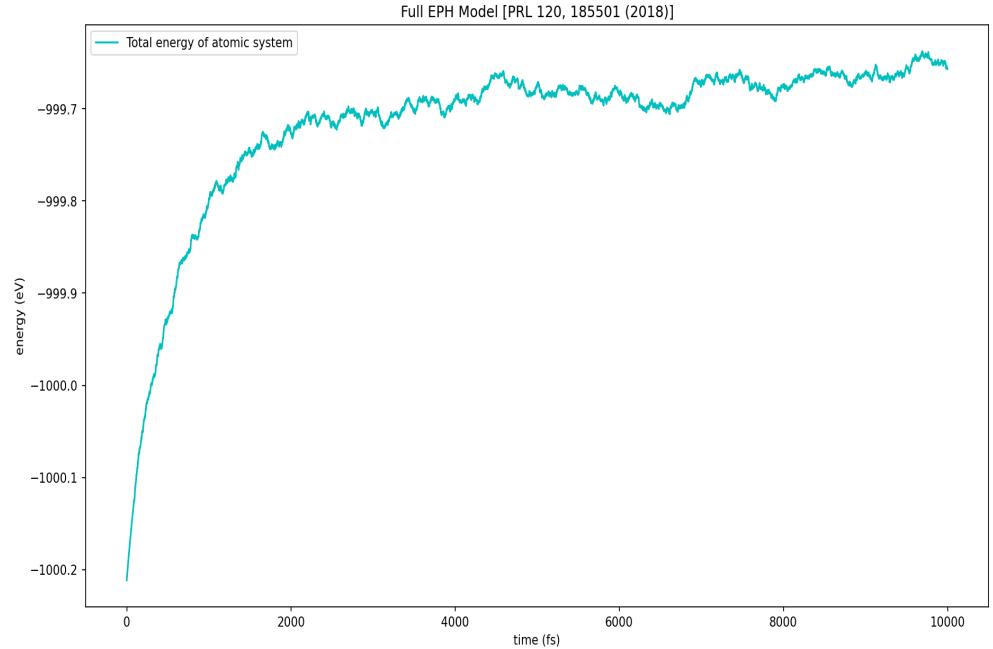


Variations of energies

Electronic system



Atomic system

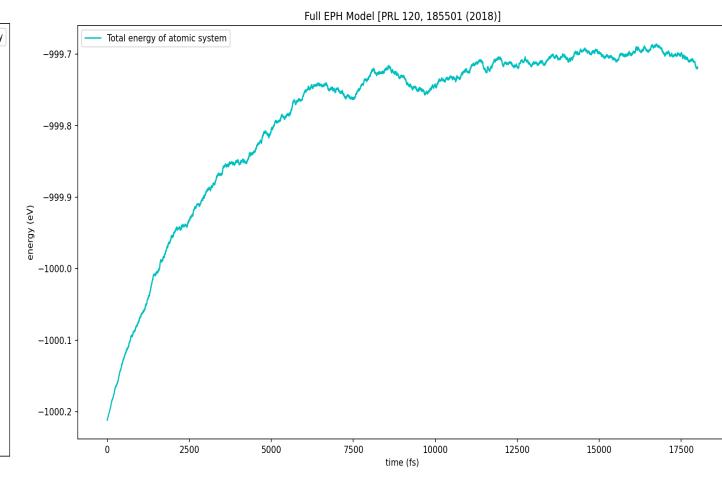
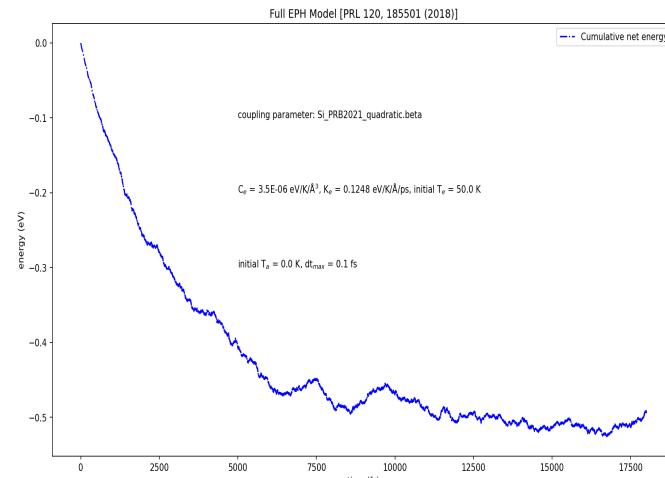
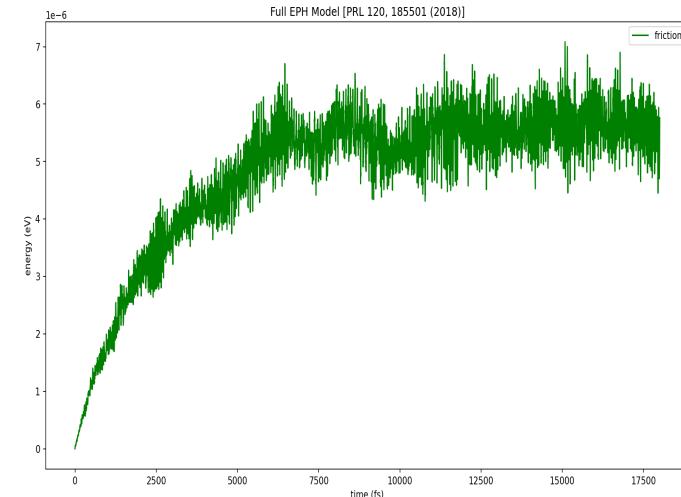
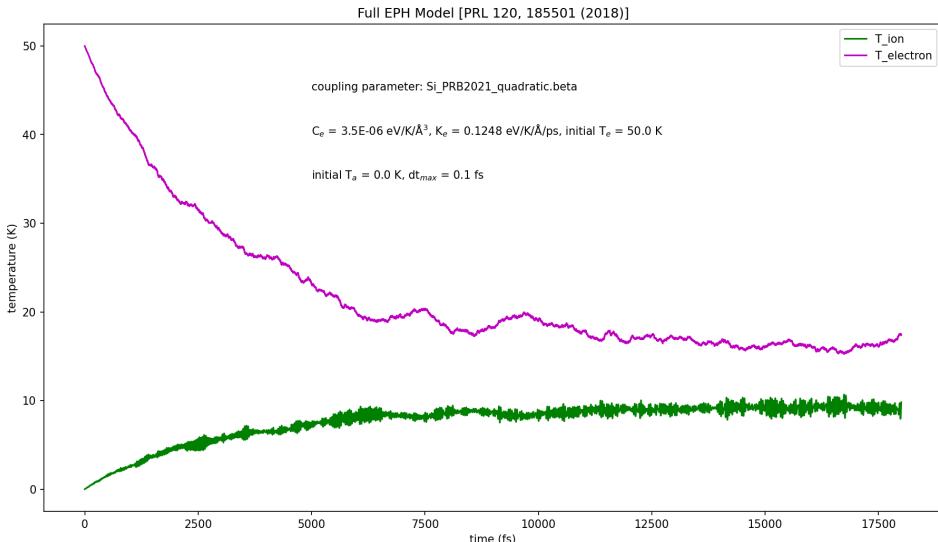


Energy is transferred from electronic to atomic system

Different coupling parameter will show some difference in rates to equilibrium.

There will also be some differences in the energies dissipated through friction and random forces

Coupling parameter – quadratic case T. Jarrin et al. Phys. Rev. B 104, 195203



Variations of atomic and electronic temperatures

The electronic mesh can be coarse or fine

216 Si atoms in a box of dimensions
16.29 Å on each side.

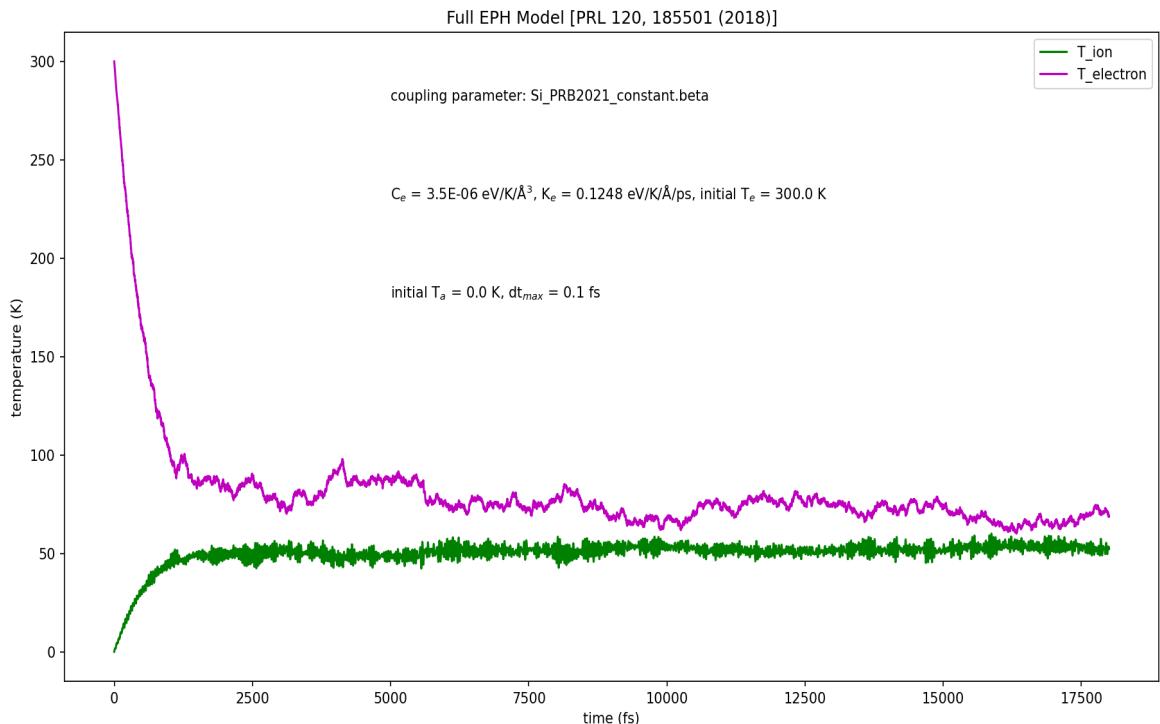
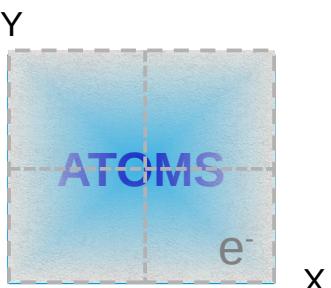
Initial atom temperature = 0.0 K

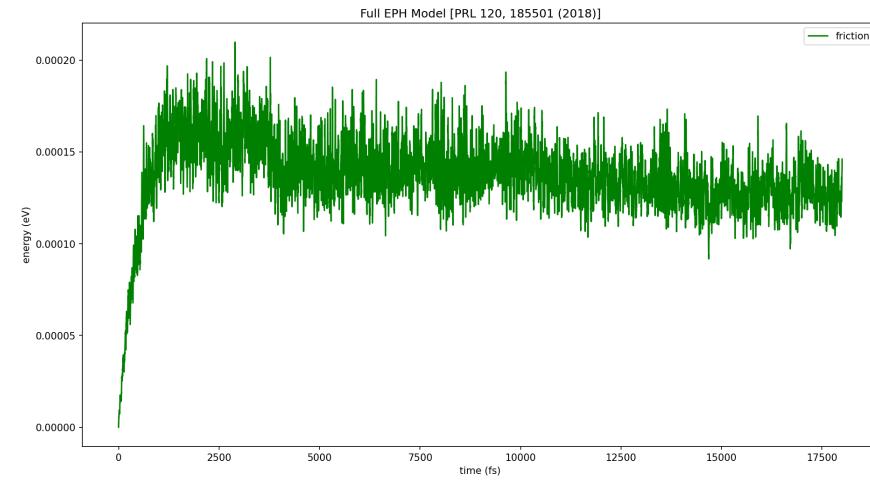
Initial electronic temperature = 300.0 K

Coupling parameter - constant case
T. Jarrin et al. Phys. Rev. B 104, 195203

Electronic parameters:
 $C_e = 3.5 \times 10^{-6} \text{ eV/K/Å}^3$
 $K_e = 0.1248 \text{ eV/K/Å/ps}$

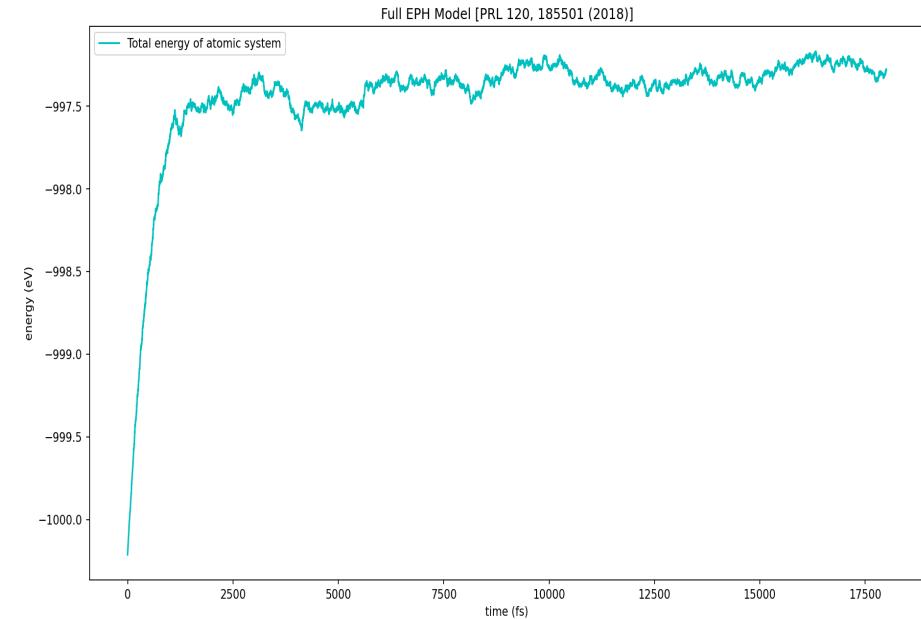
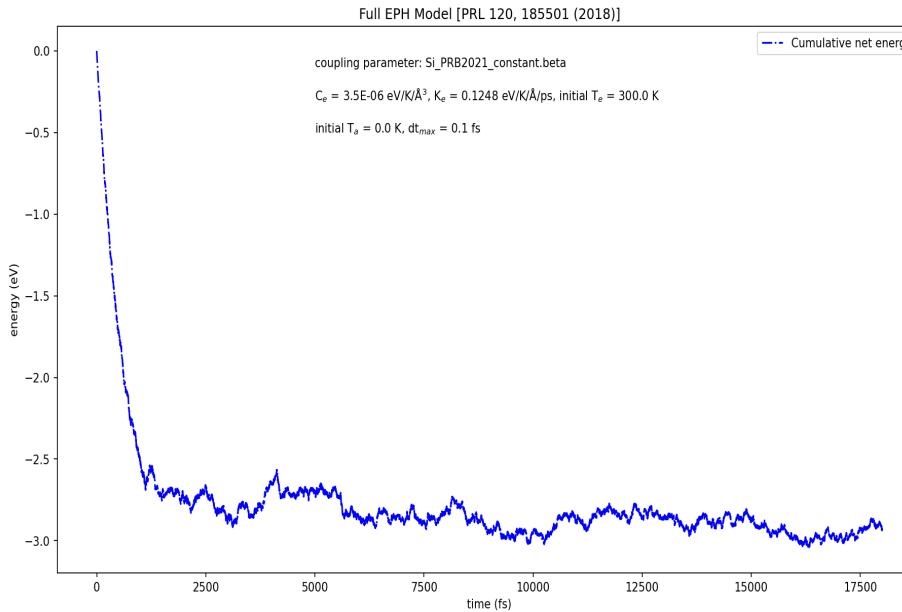
Electronic mesh:
Same size as atom box.
 $n_x = 2, n_y = 2, n_z = 2$.





Variations of energies

Energy is transferred from the electrons to the atoms



Along with the Coupling parameter, Electronic parameters also have effect on the Temperatures at equilibrium and the Energies dissipated

216 Si atoms in a box of dimensions 16.29 Å on each side.

Initial atom temperature = 0.0 K

Initial electronic temperature = 50.0 K

Coupling parameter – constant case
T. Jarrin et al. Phys. Rev. B 104, 195203

Electronic parameters:

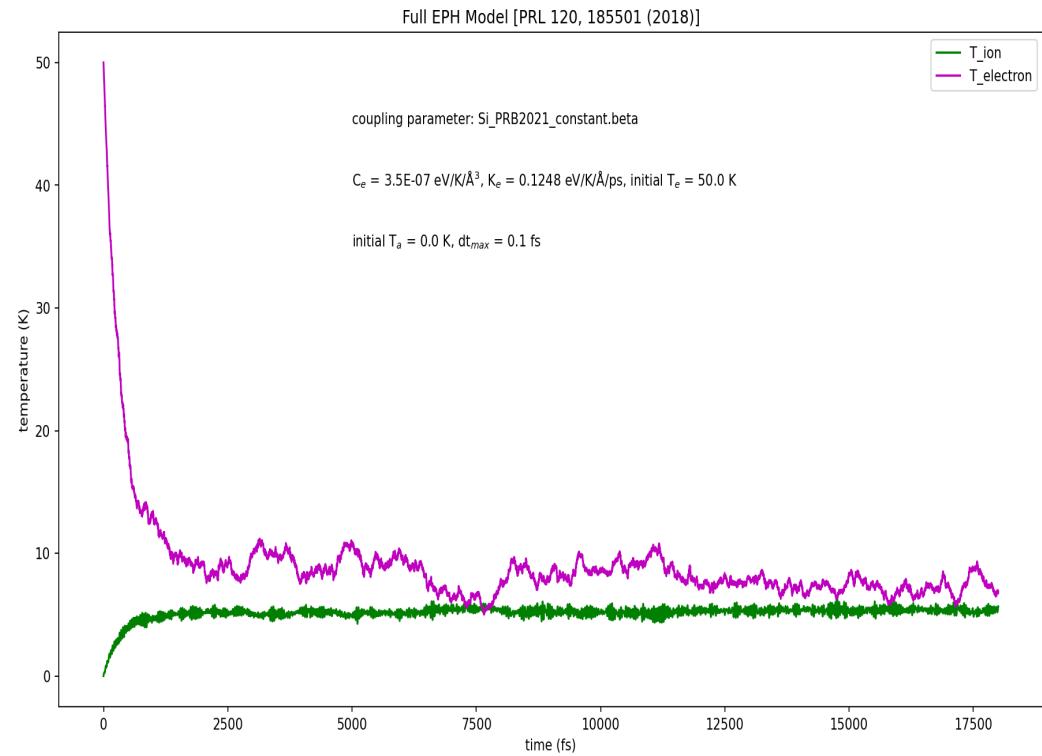
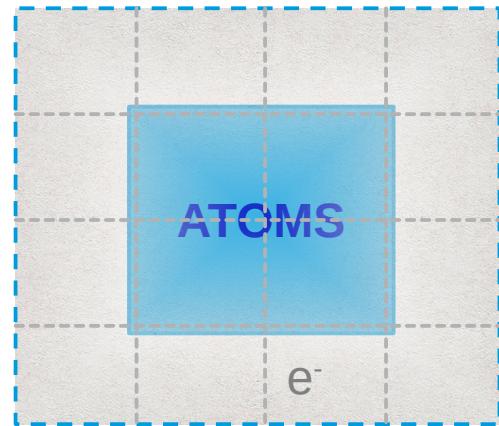
$C_e = 3.5 \times 10^{-7} \text{ eV/K/Å}^3$

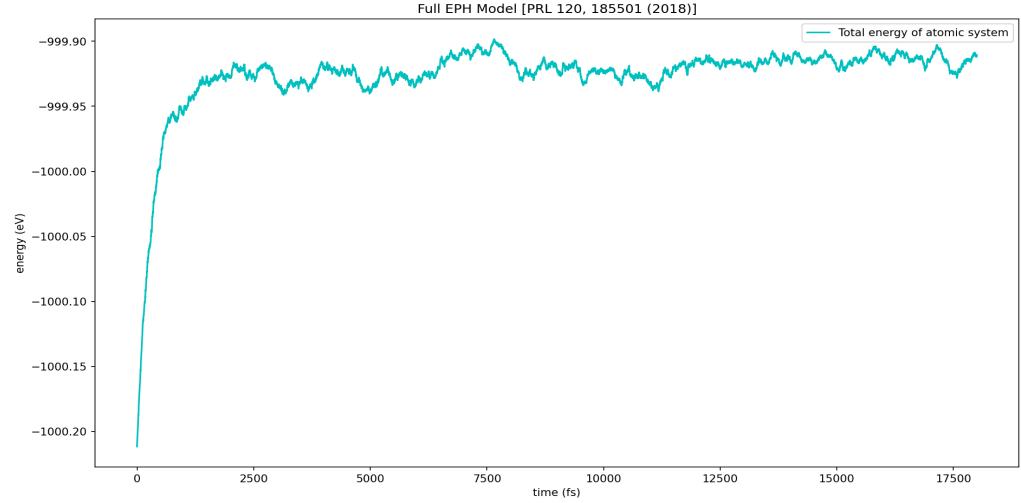
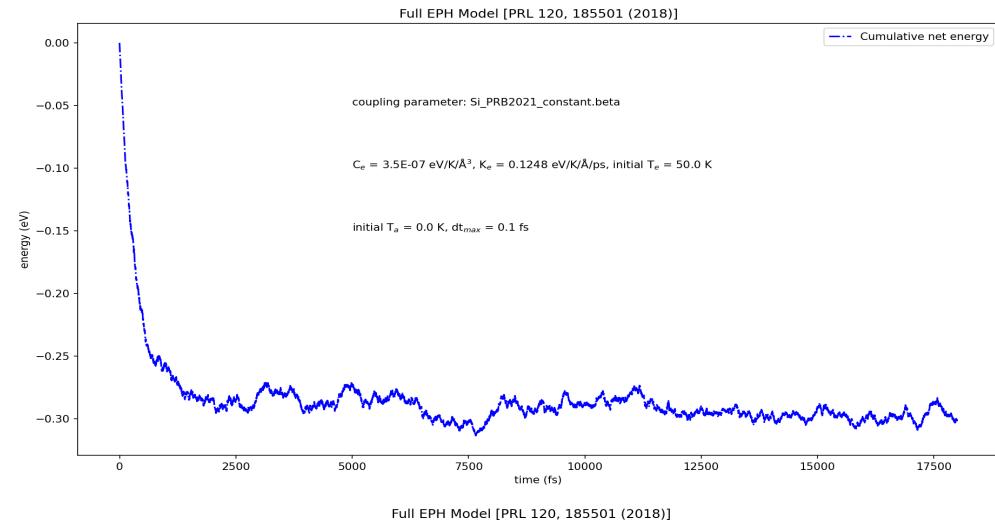
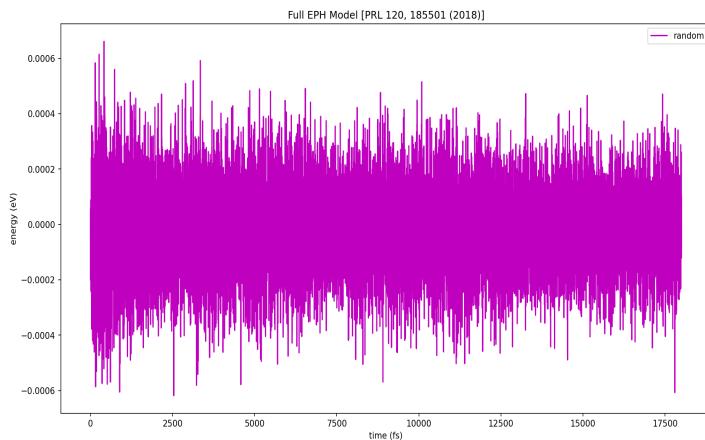
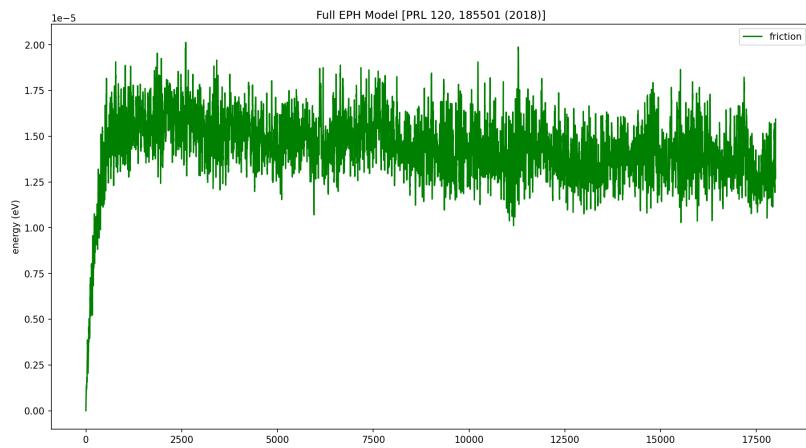
$K_e = 0.1248 \text{ eV/K/Å/ps}$

Electronic mesh:

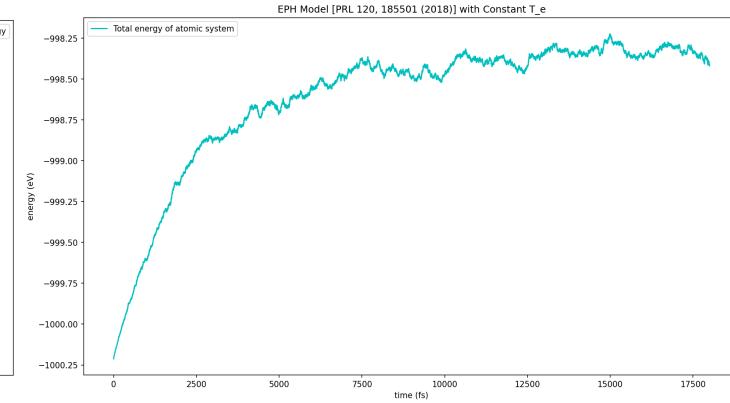
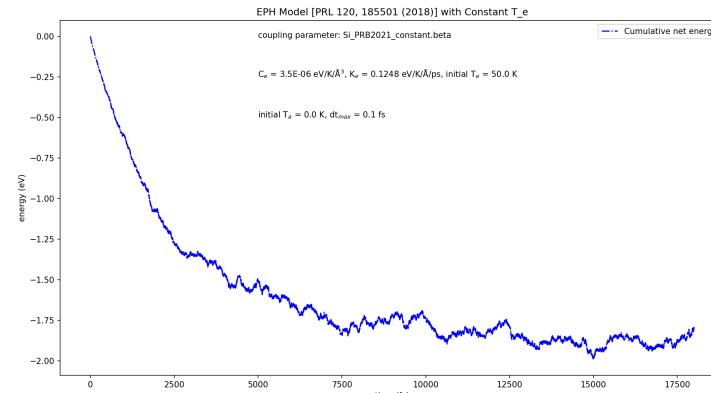
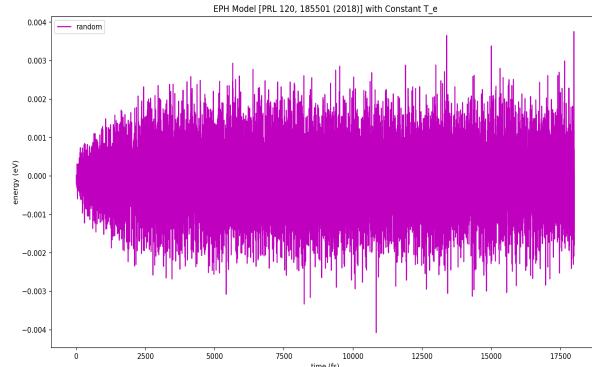
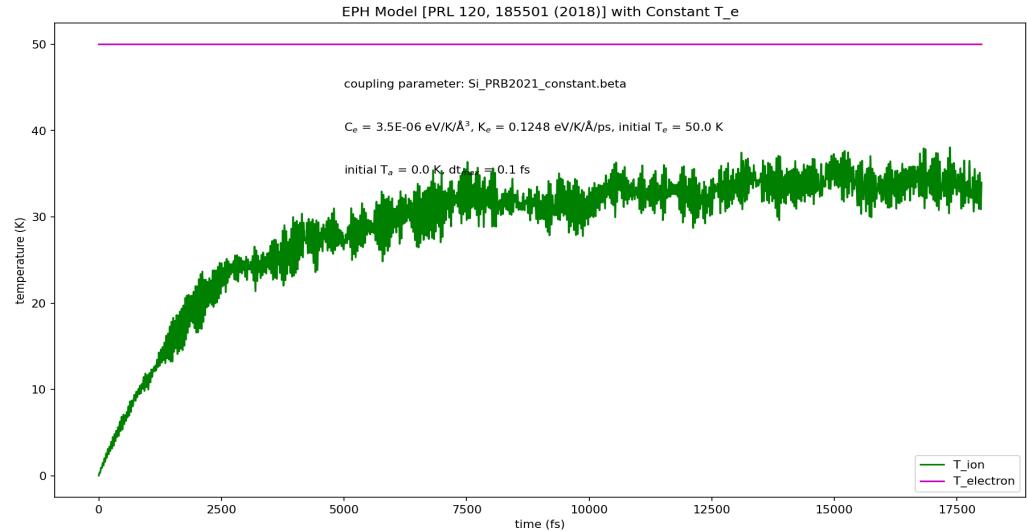
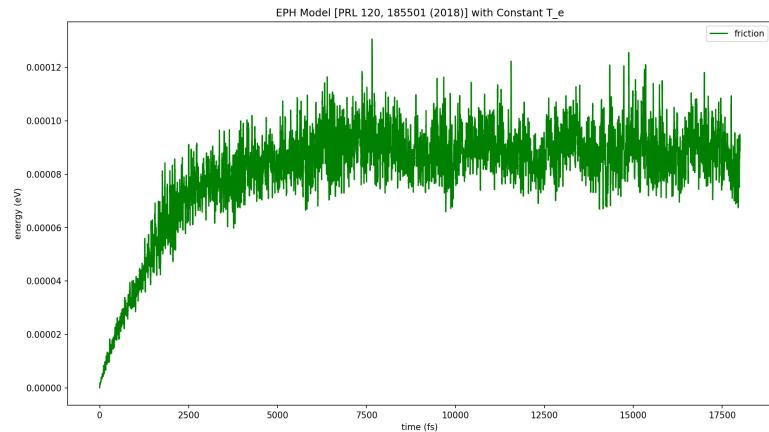
Larger than atom box
(-17.0 to +17.0 on each side).

$nx = 4, ny = 4, nz = 4$.

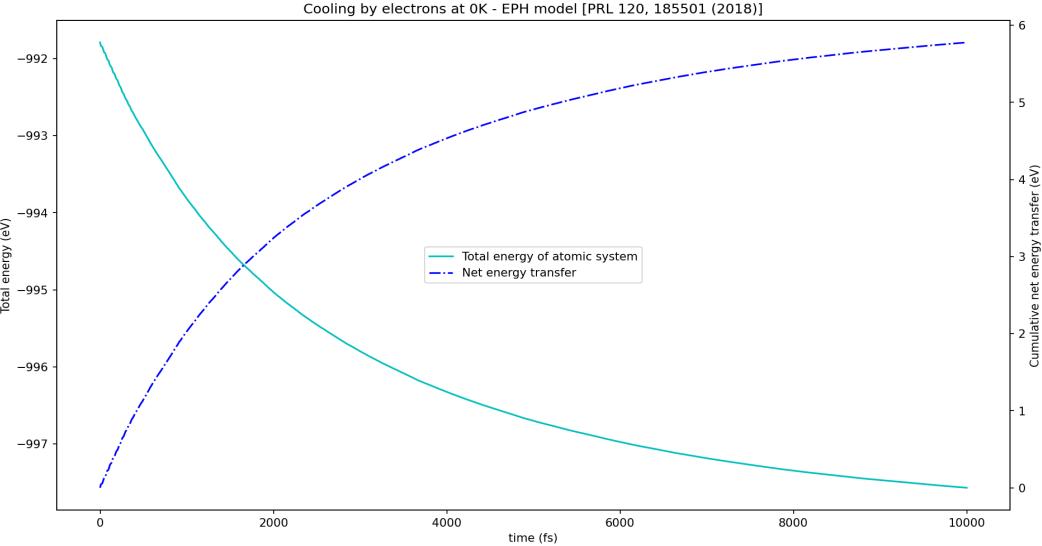
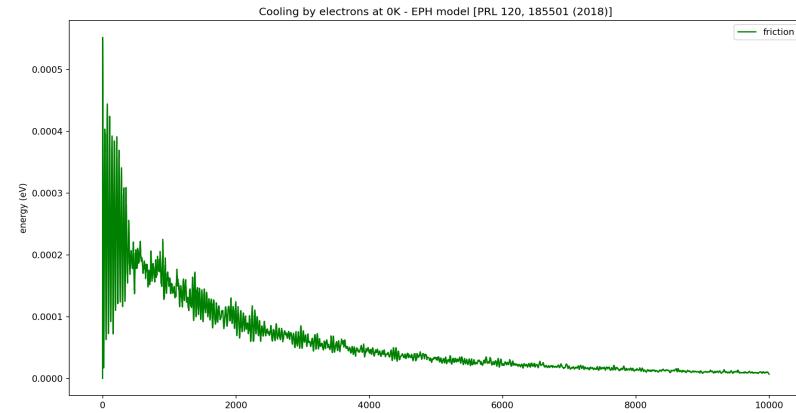
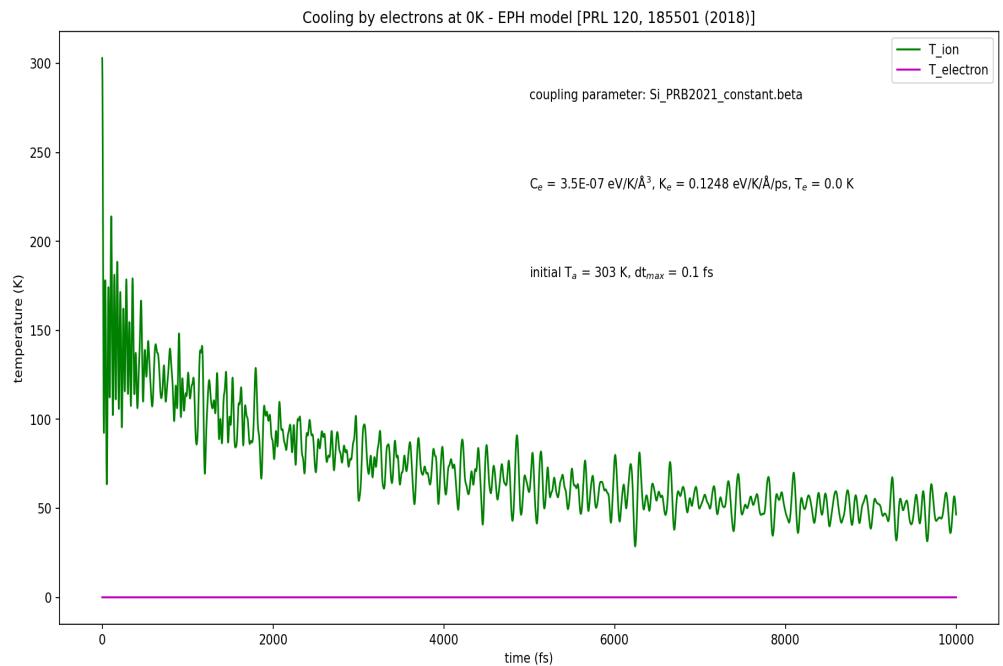




The electronic temperature is kept constant at 50 K and not updated



The system is cooled by friction forces only when the electrons are held at 0 K temperature



TurboGAP Simulations of 0.1 keV PKA with the EPH Model

EPH Model using Only Friction with 0.1 keV PKA

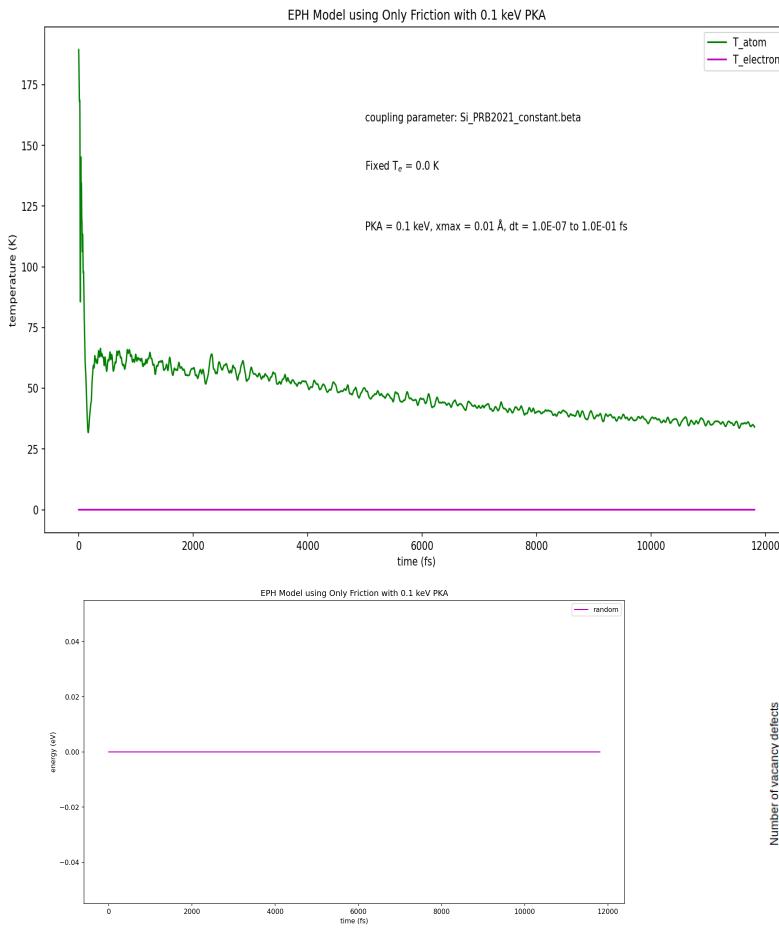
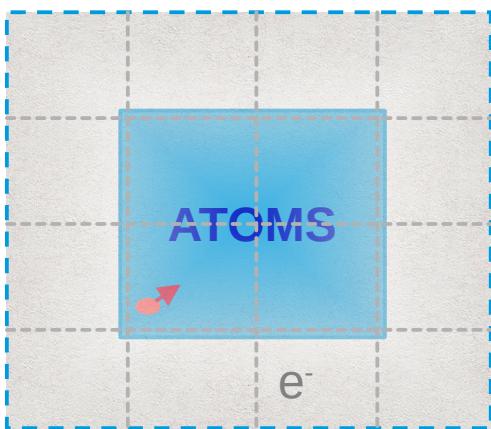
Only friction forces are applied

4096 Si atoms in a box of dimensions 43.44 \AA on each side.

Initial atom temperature $\sim 190 \text{ K}$ due to a 0.1 keV PKA

Initial and fixed electronic temperature = 0.0 K

Coupling parameter – constant case
T. Jarrin et al. Phys. Rev. B 104, 195203



TurboGAP Simulations of 0.1 keV PKA with the EPH Model

4096 Si atoms in a box of dimensions 43.44 Å on each side.

Initial atom temperature ~ 190 K due to a 0.1 keV PKA

Initial electronic temperature = 0.0 K

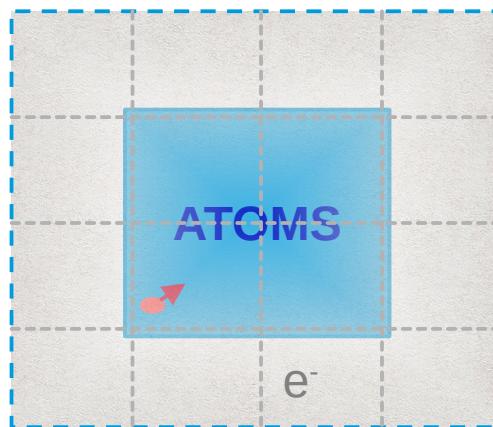
Coupling parameter – constant case
T. Jarrin et al. Phys. Rev. B 104, 195203

Electronic parameters:
 $C_e = 3.5 \times 10^{-6} \text{ eV/K/Å}^3$
 $K_e = 0.1248 \text{ eV/K/Å/ps}$

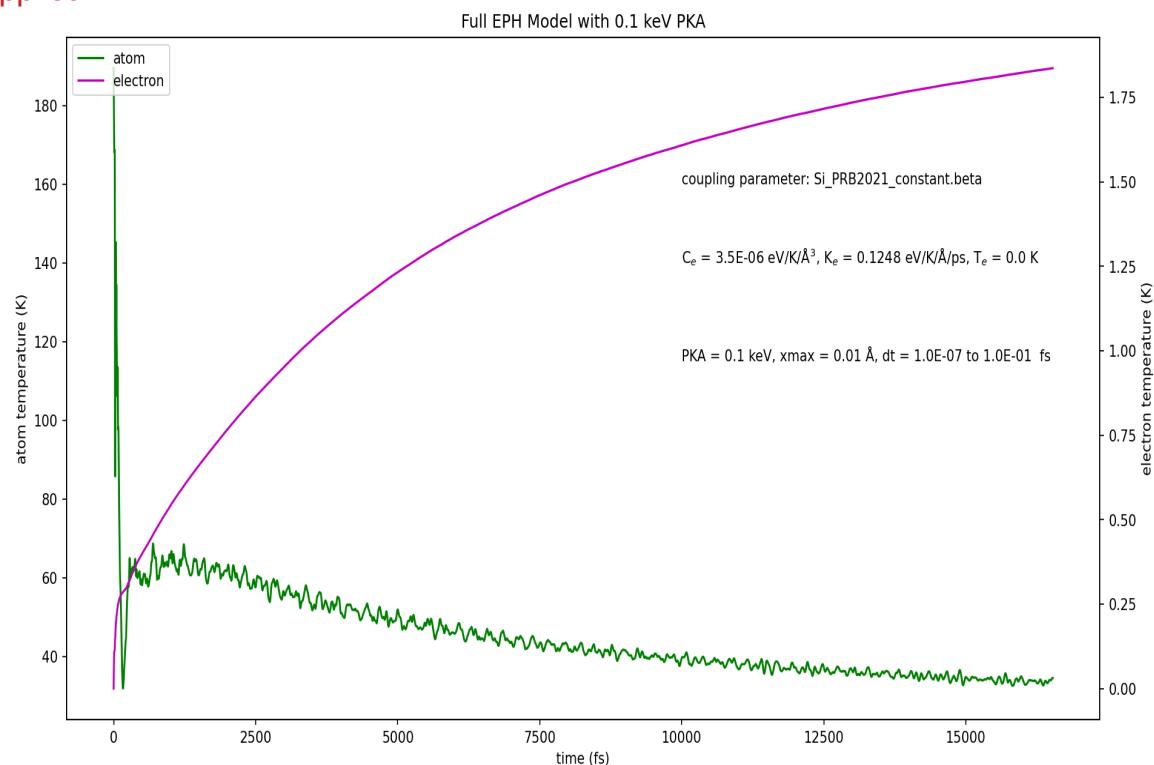
Electronic mesh:

Larger than atom box
(-100.0 to +100.0 on each side).

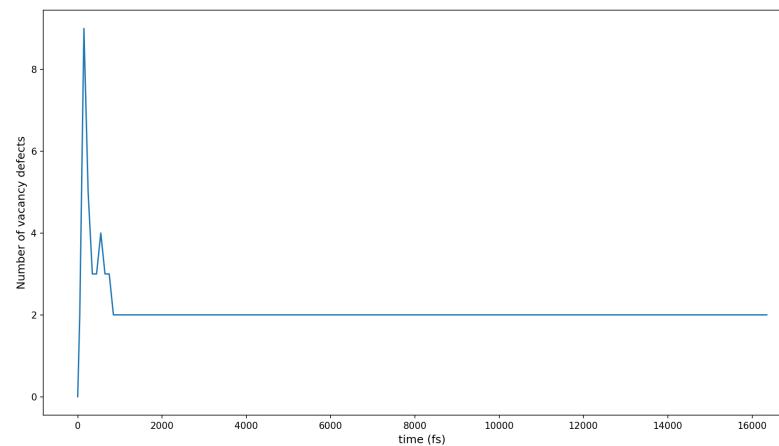
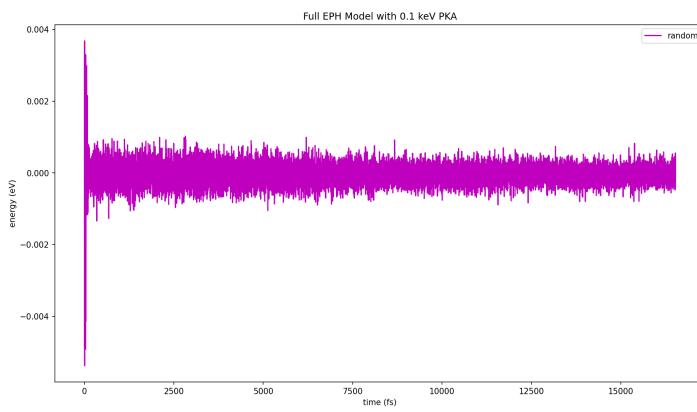
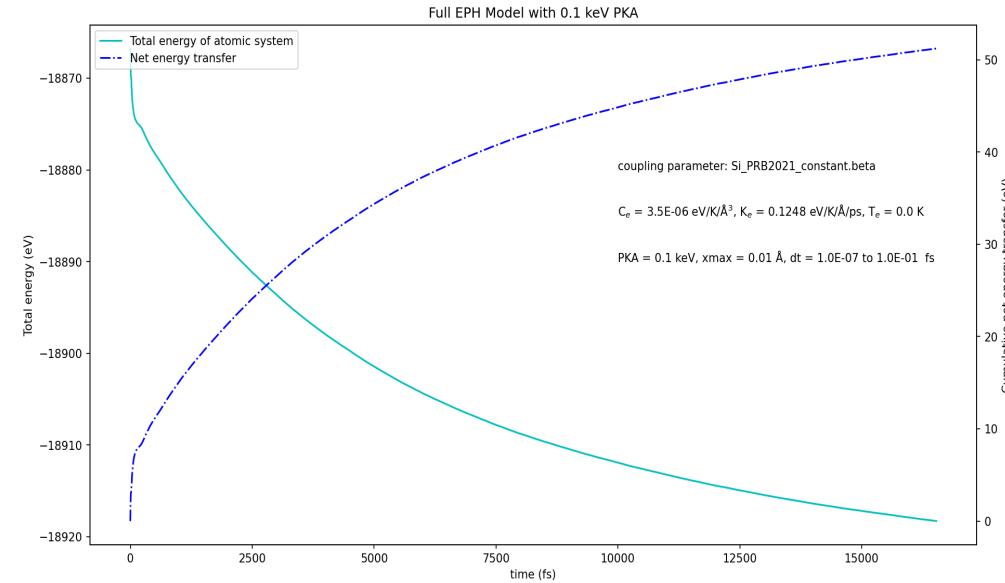
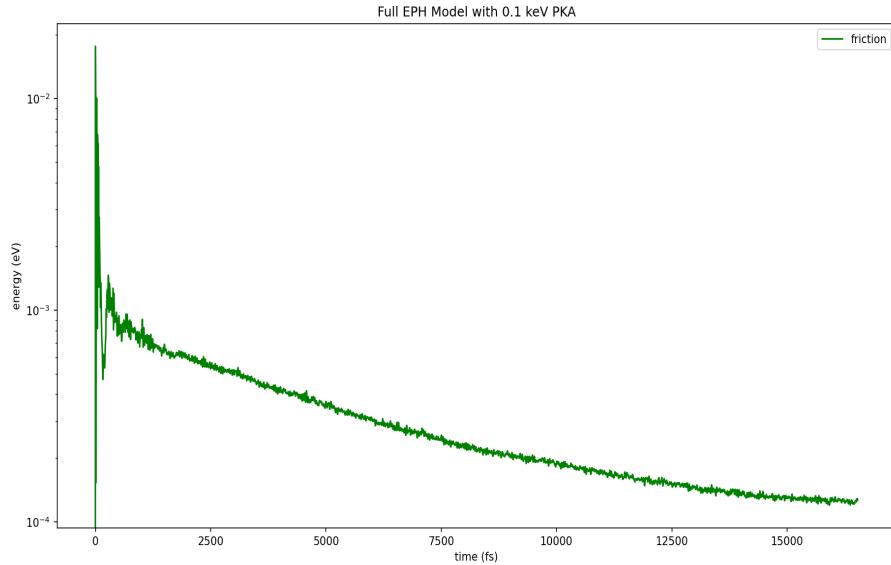
$nx = 4, ny = 4, nz = 4$.



Full model is applied



TurboGAP Simulations of 0.1 keV PKA with the EPH Model

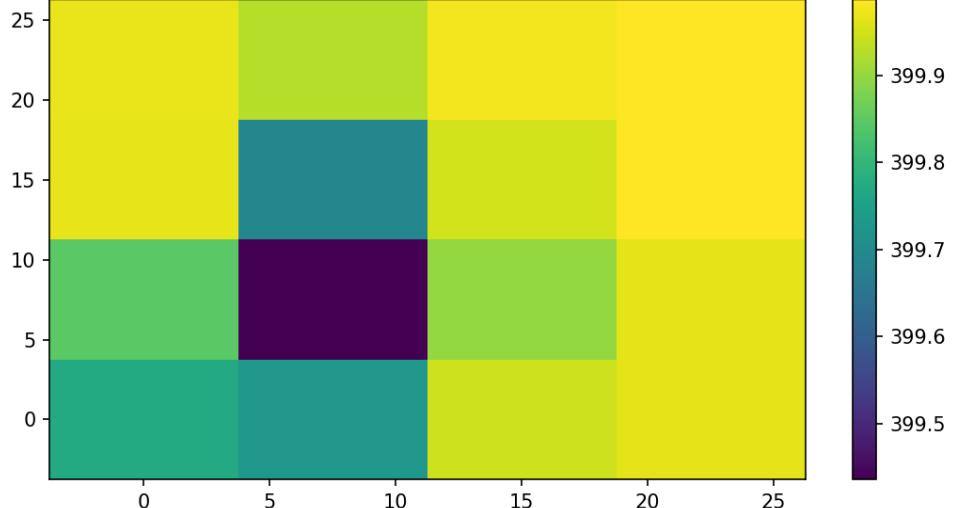


The electronic system with T-dependent parameters and external source – EPH model

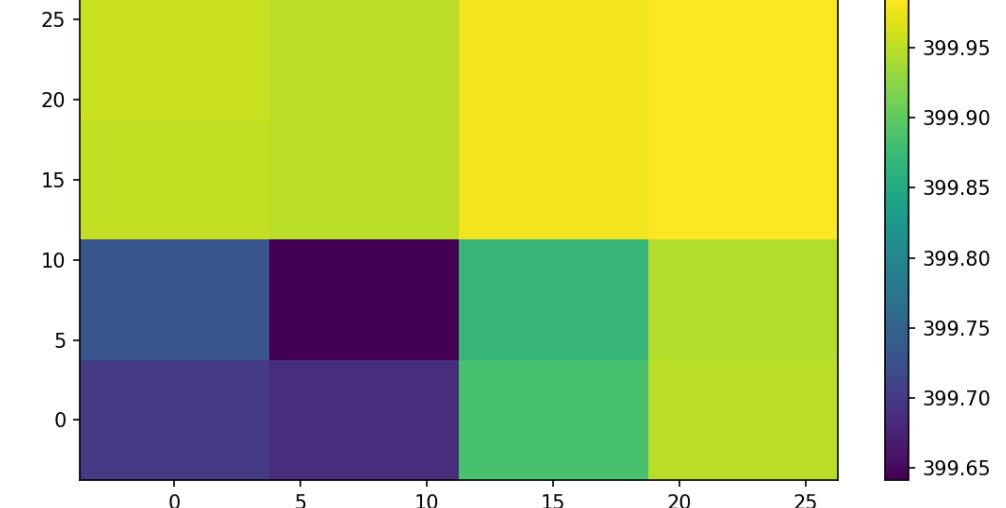
```
#  
# 3 lines of comments  
#  
4 4 4 1  
0.0 30.0  
0.0 30.0  
0.0 30.0  
ij k T_e S_e rho_e C_e K_e flag T_dyn_flag  
1 1 1 400.0 1.0E-07 1.0 0.0 0.0 0.0 1  
2 1 1 400.0 0.0E+00 1.0 0.0 0.0 0.0 0  
3 1 1 400.0 0.0E+00 1.0 0.0 0.0 0.0 0  
.....  
4 4 4 400.0 0.0E+00 1.0 0.0 0.0 0.0 0
```

```
#  
# 3 lines of comments  
#  
4 4 4 1  
0.0 30.0  
0.0 30.0  
0.0 30.0  
ij k T_e S_e rho_e C_e K_e flag T_dyn_flag  
1 1 1 400.0 1.0E-07 1.0 0.0 0.0 0.0 1  
2 1 1 400.0 0.0E+00 1.0 0.0 0.0 0.0 0  
3 1 1 400.0 0.0E+00 1.0 0.0 0.0 0.0 0  
.....  
4 4 4 400.0 0.0E+00 1.0 0.0 0.0 0.0 0
```

$z = 0$, step = 1



$z = 0$, step = 1 (a source is present)

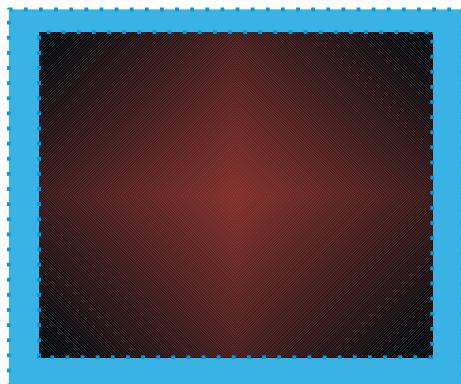


C(T) and K(T) values are provided through 'Te-dependent_e-parameters.txt' file.

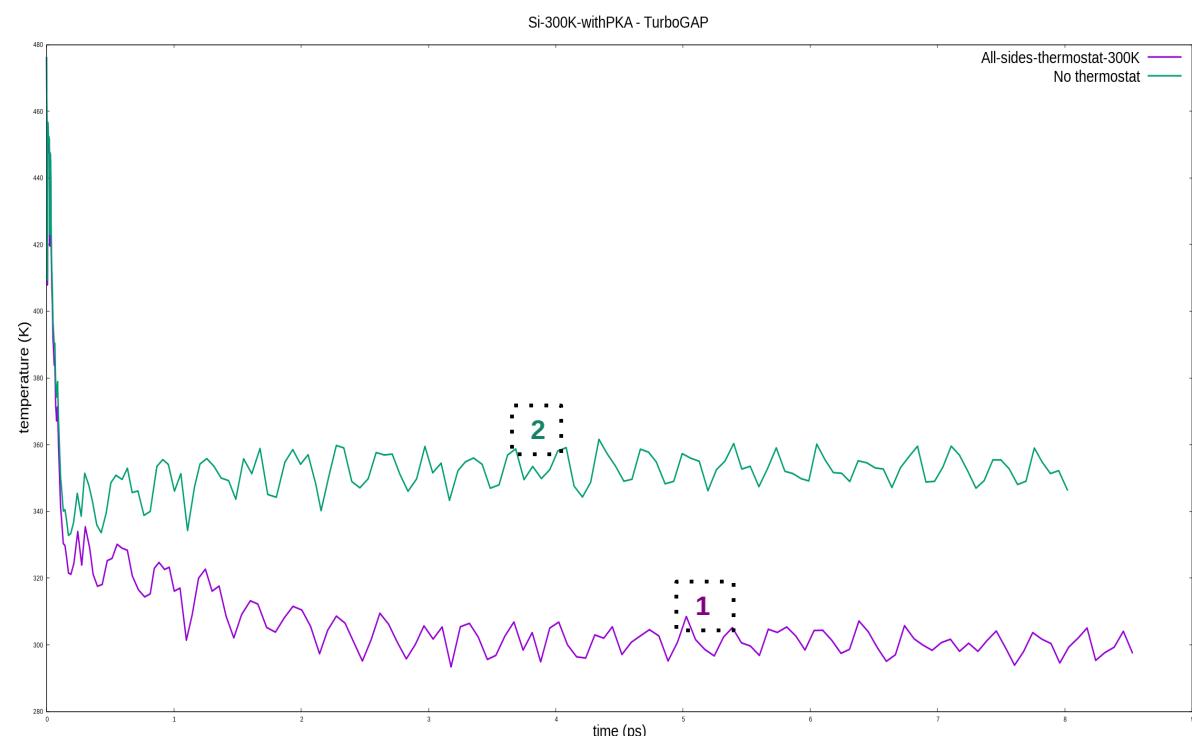
Use "make_group" command to select groups of atoms with different physical processes active

PKA simulation with/without thermostats on sides

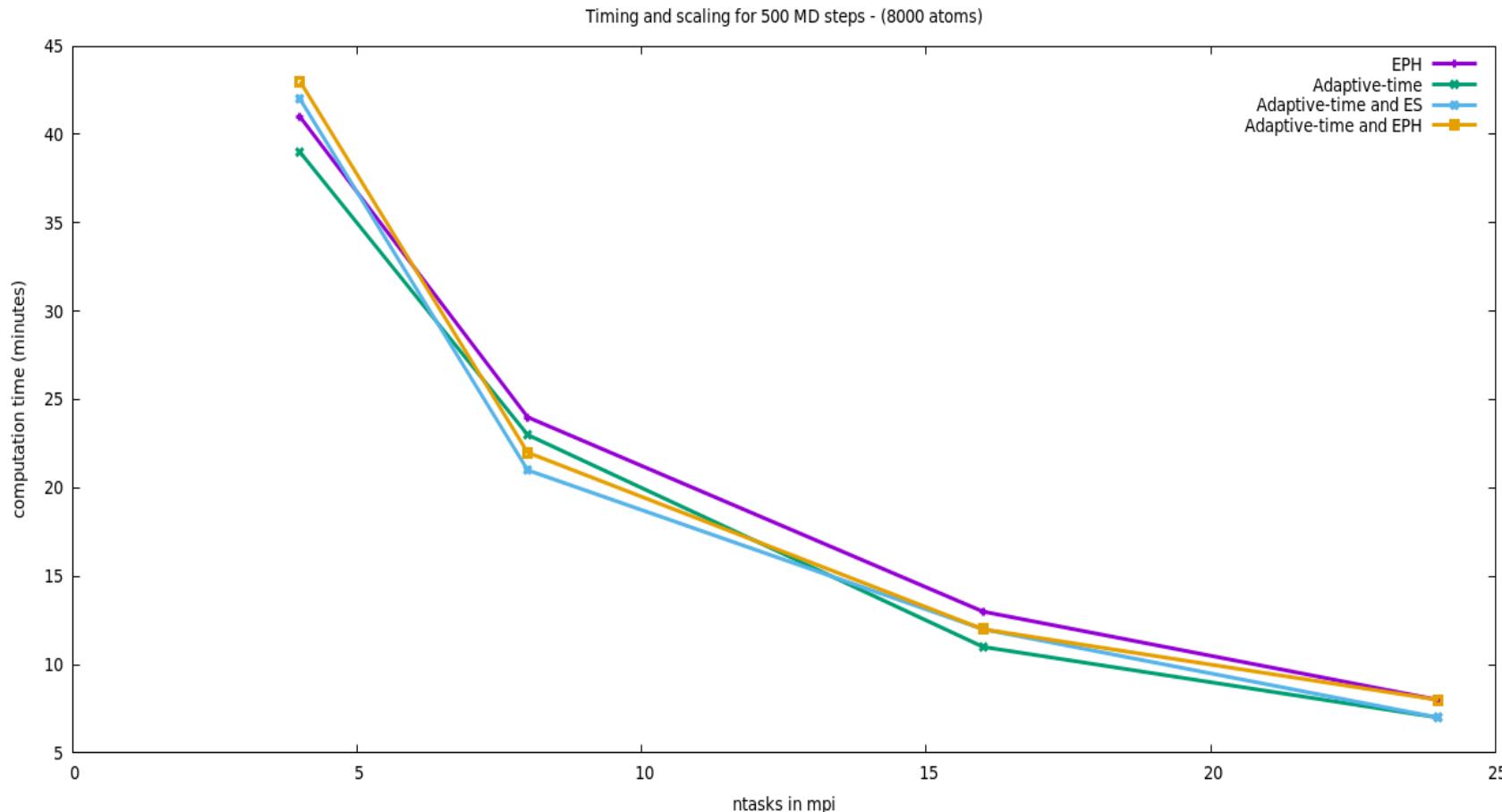
```
make_group = inside block xlow xhigh ylow yhigh zlow zhigh  
make_group = sides subtract all inside  
  
thermostat_groupID = sides  
  
optimize_groupID = all  
  
adapt_time_groupID = all  
  
eel_groupID = all
```



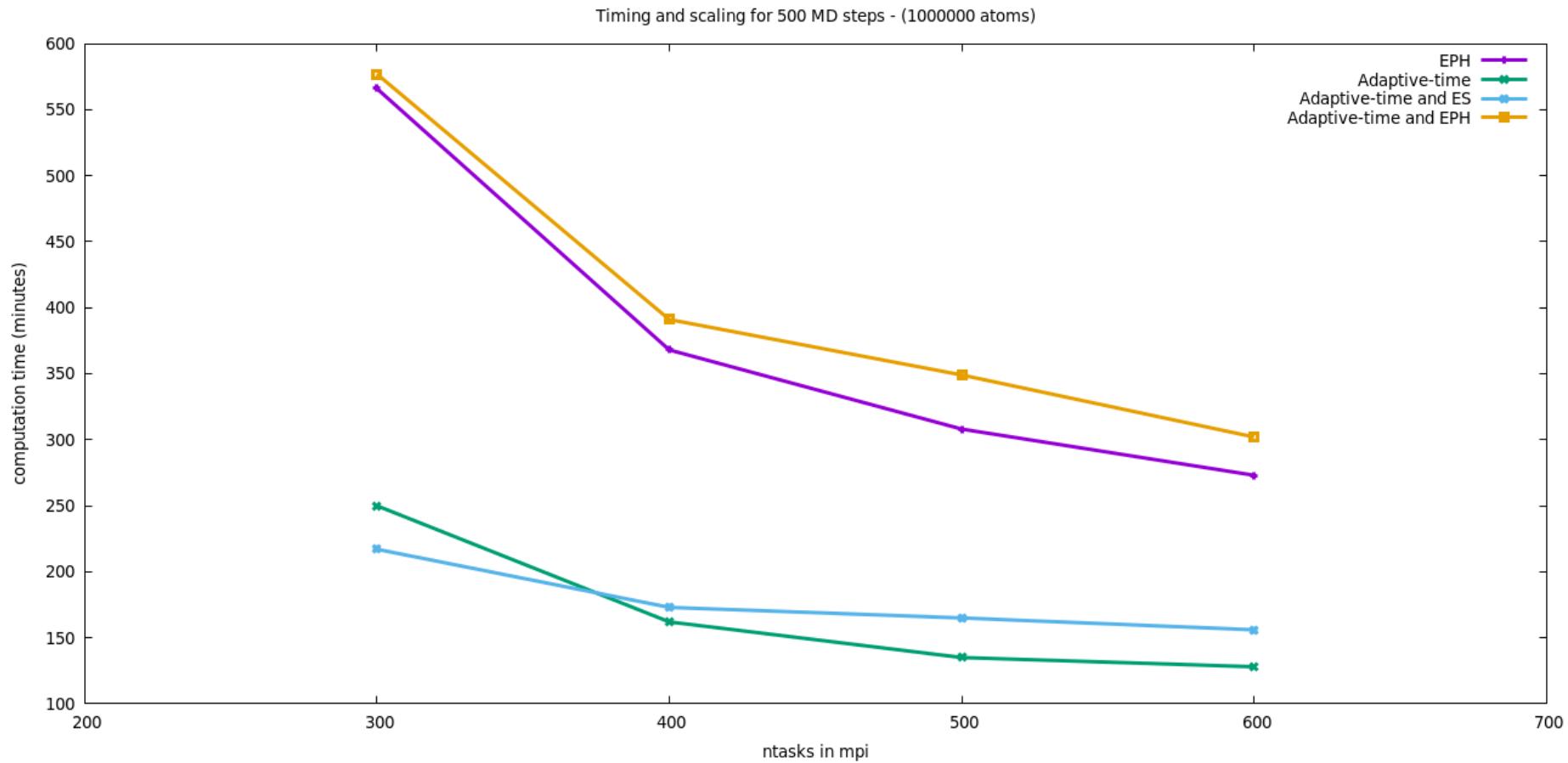
```
optimize_groupID = all  
  
adapt_time_groupID = all  
  
eel_groupID = all
```



Timing and scaling with OpenMPI parallelization of the Implementations



Timing and scaling with OpenMPI parallelization of the Implementations



At any time, due several factors computation times in a HPC cluster can have ~ 20 % spread

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

1. Adaptive time

```
adaptive_time = .true.  
adapt_time_groupID = all  
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. When switched ON, the process acts on the atoms designated by the group ID mentioned through the input keyword *adapt_time_groupID*. In order to apply this algorithm only after a certain number of time steps, the particular value should 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. Similarly, the maximum amount of kinetic energy that can be transferred in a time step is provided through the input keyword *adapt_emax*. A time step is chosen such that the distance that may be moved by any of the atoms is less than the maximum allowed displacement and the energy transfer is less than the maximum allowed value. If the allowed maximum displacement prompts for a time step that breaks the maximum energy transfer criterion, then the time step based on the latter condition is chosen. This implies that the smallest time step is always chosen. So, a reasonable value of at least one of the two criteria must be provided to do the simulation using a reasonable time step.

Default values:

```
adaptive_time = .false.; adapt_tstep_interval = 1; adapt_tmin = 0.001; adapt_tmax = 1.0;  
adapt_xmax = 0.01; adapt_emax = 10.0; adapt_time_groupID = all.
```

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

```
electronic_stopping = .true.  
eel_groupID = all  
eel_cut = 1.0  
eel_freq_out = 10
```

```

estop_filename = 'stopping-data-file'
eel_md_last_step = 0
eel_md_prev_time = 0.0
eel_E_prev_time = 0.0

```

The execution of this part can be switched ON by writing *electronic_stopping* = *.true.*. The process acts on the atoms designated by the group ID mentioned through the input keyword *eel_groupID*. When it is executed the supposed electronic energy losses of the atoms are subtracted from the atoms as if frictional forces are acting on them depending on their velocities. It is based on the electronic 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 the forces on the atom is reduced due to friction. The values of electronic energy loss with the progress of the simulation time are printed to an output file at certain interval of time steps. This frequency of output energy data can be provided through the input keyword *eel_freq_out*. The output file from this calculation will contain the per-time step, cumulative electronic energy losses and the kinetic energy and temperature of the atoms in the particular group with time in a file called ElectronicEnergyLoss.txt. When a simulation has to be restarted from a previous one then the *eel_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 *eel_md_prev_time* and *eel_E_prev_time* are then set to the MD time and the net cumulative electronic energy loss values corresponding to the last step of the previous run. The output quantities in the file ElectronicEnergyLoss.txt are updated by addition to the values from the previous run. Note, the thermo.log file from previous run needs to be save separately as it will overwritten by the next run of the code. The input 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) are provided as follows

4th line – $E_1 \quad (dE/dx)_1(\text{element 1}) \quad (dE/dx)_1(\text{element 2}) \quad (dE/dx)_1(\text{element 3}) \quad \dots$

5th line – $E_2 \quad (dE/dx)_2(\text{element 1}) \quad (dE/dx)_2(\text{element 2}) \quad (dE/dx)_2(\text{element 3}) \quad \dots$

$\dots \quad \dots \quad \dots \quad \dots \quad \dots$

*N*th line – $E_N \quad (dE/dx)_N(\text{element 1}) \quad (dE/dx)_N(\text{element 2}) \quad (dE/dx)_N(\text{element 3}) \quad \dots$

Default values:

```
electronic_stopping = .false.; eel_cut = 1.0; eel_freq_out = 1; estop_filename = 'NULL';
eel_groupID = all; eel_md_last_step = 0; eel_E_prev_time = 0.0; eel_md_prev_time = 0.0.
```

3. Non-adiabatic processes

```
nonadiabatic_processes = .true.
eph_groupID = all
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'
eph_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 = .false.*, then none of the following lines of input will be used. When this process is switched ON it will act on the atoms designated by the group ID mentioned through the input keyword *eph_groupID*. 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. The per-time step friction energy, random energy and the cumulative net energy (due to both friction and random energy exchanges)

and the electronic temperature, along with the temperature, kinetic and potential energies of the atoms in the particular group are provided in the output file eph-EnergySharingData.txt. 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 output quantities in the file 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 (because this file is related and common to other general parts of the TurboGAP code). The name of the file for getting this mesh parameters output is provided through input keyword *eph_Toutfile*. The data for electron-ion coupling parameter obtained from TD-DFT simulations is provided in a text file using the input keyword *eph_betafile*. The input keyword *eph_Tinfile* is used to provide the text file name where several of the input parameters are provided. 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:

eph_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’ (see the list of all default values below). 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_fdm_option* are turned off, then the *eph_fdm_option* should as well be turned off.

Note that some calculations can only be performed by providing the input data through the text file in input keyword *eph_Tinfile*. When there is any external source term and / or the electronic heat capacity and thermal conductivity are known to be temperature dependent, this file should be used. In the file, if any ‘flag’ is set as 1, then external source term ‘S_e’ will be considered. Similarly, if ‘T_dyn_flag’ is set as 1, then the electronic heat capacity per volume ‘C_e’ and electronic thermal conductivity ‘K_e’ are considered to be temperature dependent. Then the data for C_e(T_e) and K_e(T_e) with increasing temperatures must be provided in a text file named as “Te-dependent_e-parameters.txt”. The format of these data must be as follows:

Default values:

```

nonadiabatic_processes = .false.; eph_fdm_option = 1; eph_friction_option = 1;
eph_random_option = 1; eph_md_last_step = 0; eph_freq_Tout = 1; eph_freq_mesh_Tout = 1;
eph_fdm_steps = 1; eph_gsx = 1; eph_gsy = 1; eph_gsz = 1; model_eph = 1; eph_rho_e = 1.0;
eph_C_e = 1.0; eph_kappa_e = 1.0; eph_Ti_e = 300.0; eph_box_limits = -100.0, 100.0, -100.0,
100.0, -100.0, 100.0; eph_E_prev_time = 0.0; eph_md_prev_time = 0.0; eph_Tinfile = 'NULL';
eph_Toutfile = 'NULL'; eph_betafile = 'NULL'; eph_groupID = all.

```

Input keyword **make_group**

In TurboGAP one or more atoms can be combined into groups on which different physical processes can be made to act while doing an MD simulation. For example, a thermostat can be applied to the sides of a cubic box when another process can be active on rest of the atoms inside the side walls. There are several ways by which different groups of atoms can be made. The *make_group* keyword must be used to designate one or more atoms into a specific group. In general any input keyword of the program can be used only once in the input script. However, as more than one group may need to be defined for modeling specific setups, the *make_group* keyword can be used more than once in the input script. The general format for using the *make_group* command is as follows:

```
make_group = group_ID    group_Style    Number_specific_to_group_style    Entries_specific_to_group_style
```

So, there are four kinds of input data that are required in order to define a group of atom(s). Each of these are explained in the following.

group_ID – it is the ID (or name) of the present group that is being defined. The atoms in the group will be identified by this name. For any subsequent operation on this group of atoms this group ID has to be referred. The name can be a combination of characters and numbers up to length 16.

group_Style – it is one of various styles (presently seven) by which a group of atoms can be defined. Presently available group styles are *block*, *add*, *subtract*, *id*, *sphere*, *atomtype*, *dynamic*. The group style *all* is already defined for all atoms in the simulation box and cannot be re-defined. Each of these group styles are explained below which will help to understand the third and fourth kinds of input data (*Number_specific_to_group_style* and *Entries_specific_to_group_style*) specific to the style.

Note a group of all atoms with group style *all* and group ID *all* is defined by default. At present, a maximum of additional 16 groups can be made.

Group style **block**

This style is used to define a block of atoms. The block is built with the lower and upper limits of values along the x-, y- and z-axes (i.e., xlow, xhigh, ylow, yhigh, etc.) specified in the input. Here *Number_specific_to_group_style* is the number of values that will be provided to define block (or blocks together). One block is defined by 6 limits, so when a block style group is being made this number should be equal to a multiple of 6. The *Entries_specific_to_group_style* are real numbers denoting the limits of the block (or blocks together). Examples:

1. `make_group = blk1 block 6 xlow xhigh ylow yhigh zlow zhigh`

This will make a group named blk1 containing the atoms whose coordinates are within the limits (including the limiting values).

2. `make_group = blk2 block 12`

`xlow1 xhigh1 ylow1 yhigh1 zlow1 zhigh1`

`xlow2 xhigh2 ylow2 yhigh2 zlow2 zhigh2`

This will make a group of atoms whose coordinates are within any of the two blocks defined by the limits.

3. `make_group = blk3 block 36`

`xlow1 xhigh1 ylow1 yhigh1 zlow1 zhigh1`

`xlow2 xhigh2 ylow2 yhigh2 zlow2 zhigh2`

xlow6 xhigh6 ylow6 yhigh6 zlow6 zhight6

This makes a single group of atoms named as blk3 where six block are defined together (they can be all the sides of the simulation box, etc.)

Group style add

This style is used to add the atoms in two or more groups to the group that is presently being defined. So, necessarily the `Number_specific_to_group_style` should be at least equal to 2, followed those many `Entries_specific_to_group_style`, which are now just the IDs of the groups whose atoms should be added into the present group. The groups to be considered for adding should be already defined before performing this operation. Example:

1. `make_group = blk123 add 3 blk1 blk2 blk3`

This will make group blk123 contain all the atoms that are in groups blk1, blk2 and blk3.

Group style id

This style is used to make a group of atoms with the mentioned atom IDs. The `Number_specific_to_group_style` is the number of IDs that will be provided and the `Entries_specific_to_group_style` are the IDs of the atoms. Example:

1. `make_group = grp1 id 5 21 33 212 300 100`

This will make a group named grp1 with five atoms whose IDs are as given.

Group style subtract

This style is used to form a group of the atoms which is only in one group and none of the other groups. Here again, the `Number_specific_to_group_style` should be at least equal to 2, followed those many `Entries_specific_to_group_style`, which are the IDs of the groups. The present group will be made with those atoms which are in the group mentioned first and in none of the subsequent groups. Example:

1. `make_group = sub1 subtract 2 all blk3`

This makes a group named sub1 containing all atoms in the simulation box which are not in the group blk3.

2. `make_group = sub subtract 4 all blk1 blk2 blk3`

This makes a group named sub where those atoms only are present which are not in any of the groups blk1 blk2 blk3.

Group style sphere

This style is used to define a group of atoms which fall within a sphere of a given radius. The `Number_specific_to_group_style` here is 4 and the `Entries_specific_to_group_style` are first the radius of the sphere and then followed by the x, y and z coordinates of the centre of the sphere. Example:

1. `make_group = sph1 sphere 4 3.2 4.2 5.0 4.2`

This makes a group sph1 for the atoms which are within a sphere of radius 3.2 Å and centred at (4.2, 5.0, 4.2).

Group style atomtype

This style is used to group atoms by their types. The `Number_specific_to_group_style` should be at least equal to 1 and can be maximum equal to the number of species of atoms in the simulation box. The `Entries_specific_to_group_style` are just the symbols of the elements or species. This may be required only when the simulation system consists of many types of atoms. Example:

1. `make_group = grp2 atomtype 1 H`

This makes a group named grp2 of all the H atoms in the system, provided there are multiple species in the system and H is one of them.

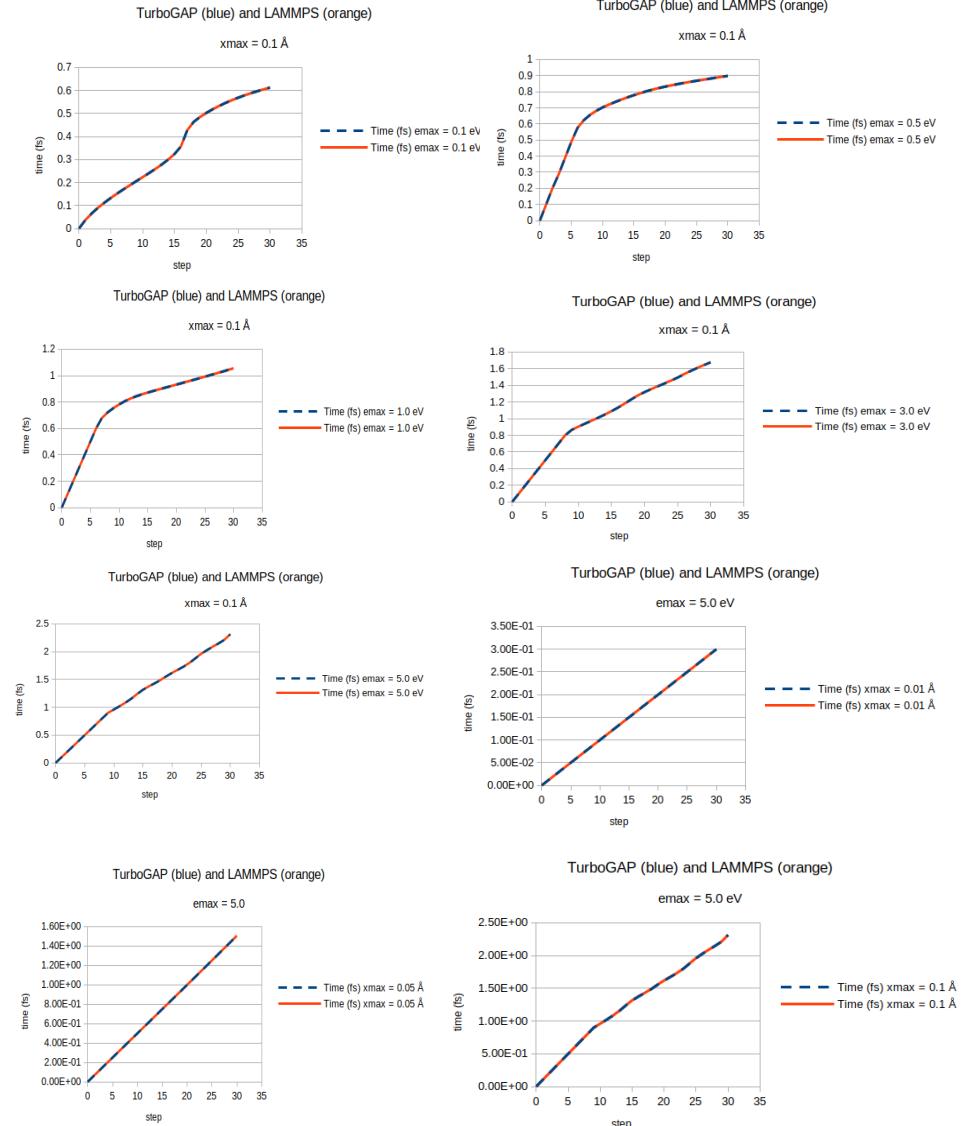
Group style *dynamic*

This style is used to designate an already defined group as dynamic, meaning that the atoms in this group will be checked if they still belong to the geometrical region that was originally defined for them as the MD time has evolved. This check is performed at an interval of MD steps provided through the `Number_specific_to_group_style`. The update of the specified group is performed at the specified interval after the inter-atomic forces have been computed. Note that defining groups as dynamic is computation expensive and doing too frequent an update of the dynamic groups will also increase computation times. Example:

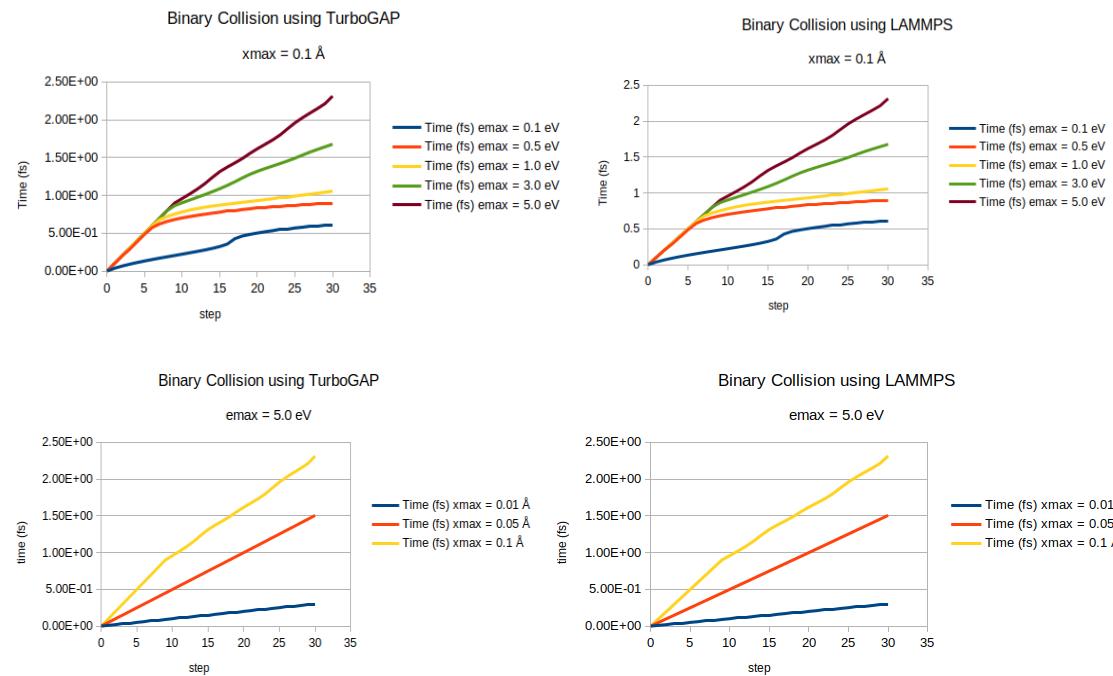
1. `make_group = blk2 dynamic 1000`
2. `make_group = sph1 dynamic 100`

In example 1, the group blk2 of style *block* defined earlier will be checked for update of its constituent atoms, which means it will be remade, after every 1000 MD time-steps. In example 2, the group sph1 of style *sphere* defined earlier will undergo the same update after every 100 steps.

Binary Collision Simulations using TurboGAP and LAMMPS and sensitivity of time step to xmax and emax criterion

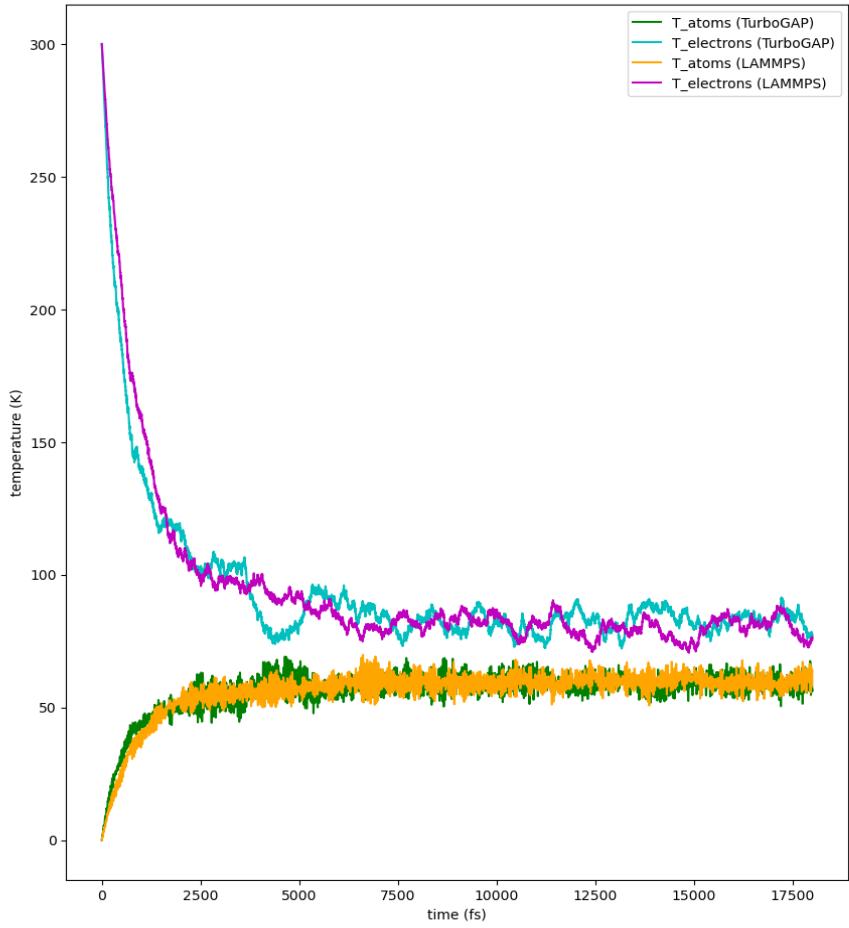


Two Si atoms
at $r_1 = (10.0, 10.0, 10.0)$ and $r_2 = (13.0, 10.0, 10.0)$
with $v_{1x} = 1.0 \text{ \AA/fs}$, $v_{1y} = 0.0$, $v_{1z} = 0.0$ and $v_{2x} = -1.0 \text{ \AA/fs}$, $v_{2y} = 0.0$, $v_{2z} = 0.0$
using Si (not stiffened) GAP potential.



Full EPH model in TurboGAP and LAMMPS

216 Si atoms in a box of dimensions 16.29 Å on each side.
Using Si-GAP potential.

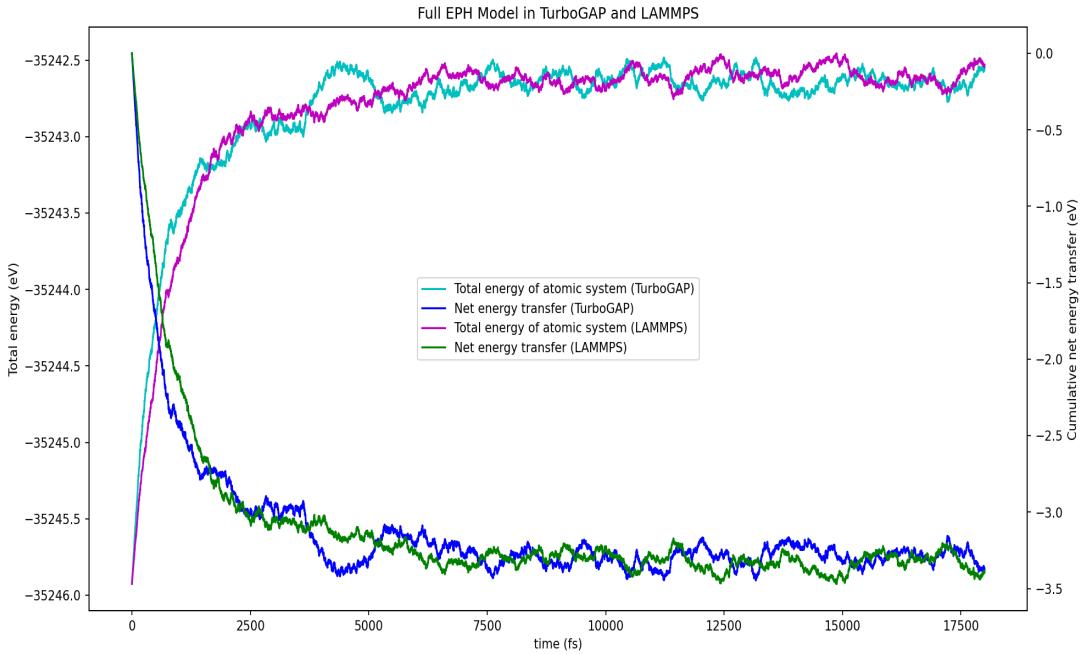


T_a = 0.0 K, T_e = 300.0 K
Coupling parameter – constant case
T. Jarrin et al. Phys. Rev. B 104, 195203

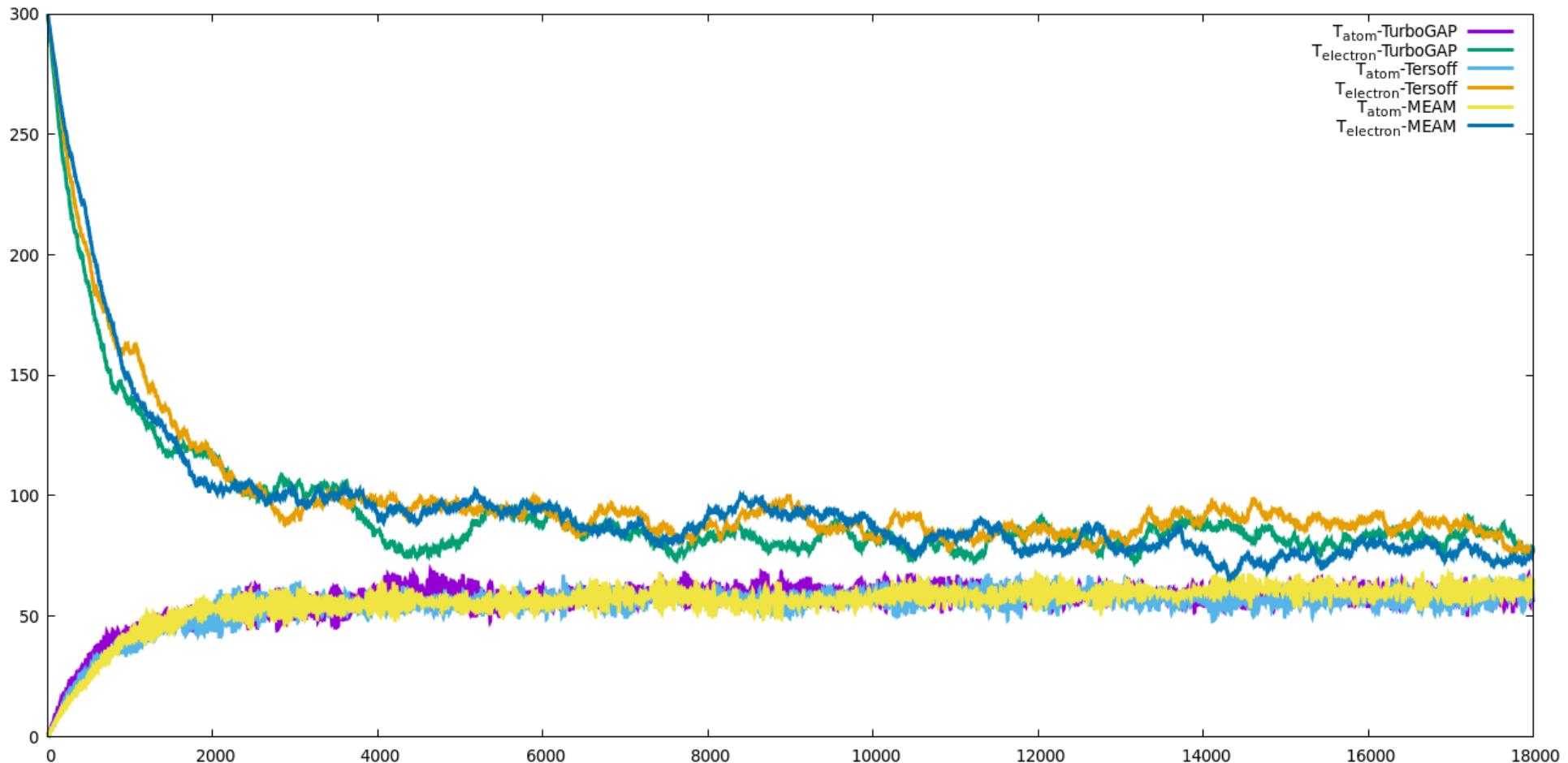
$$C_e = 3.5 \times 10^{-6} \text{ eV/K/Å}^3$$

$$K_e = 0.1248 \text{ eV/K/Å/ps}$$

Electron system size is same as the atomic system size.



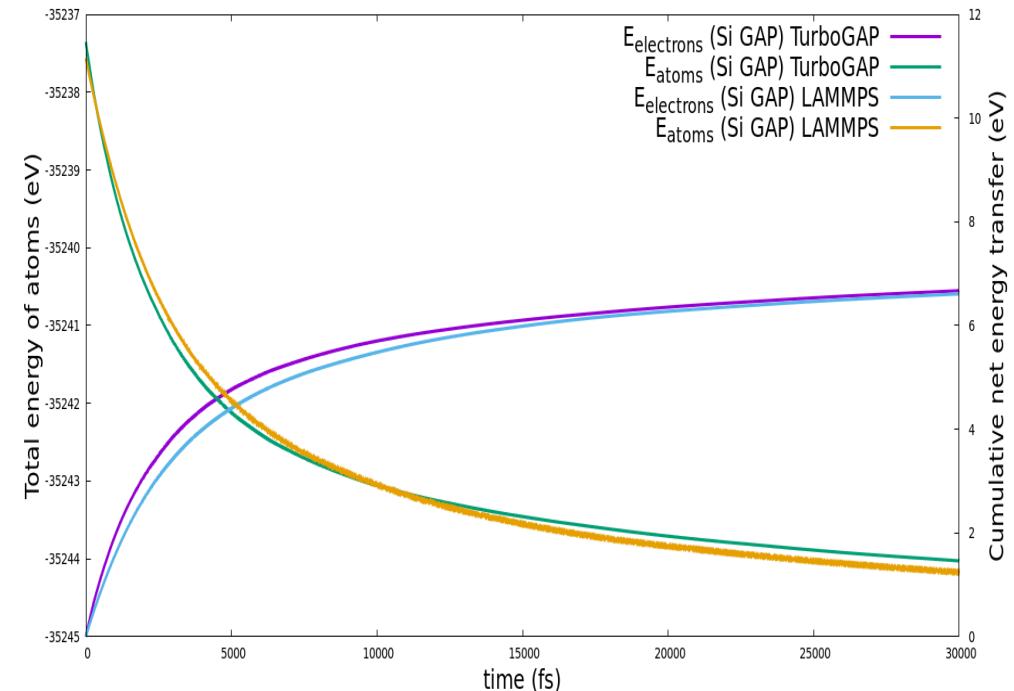
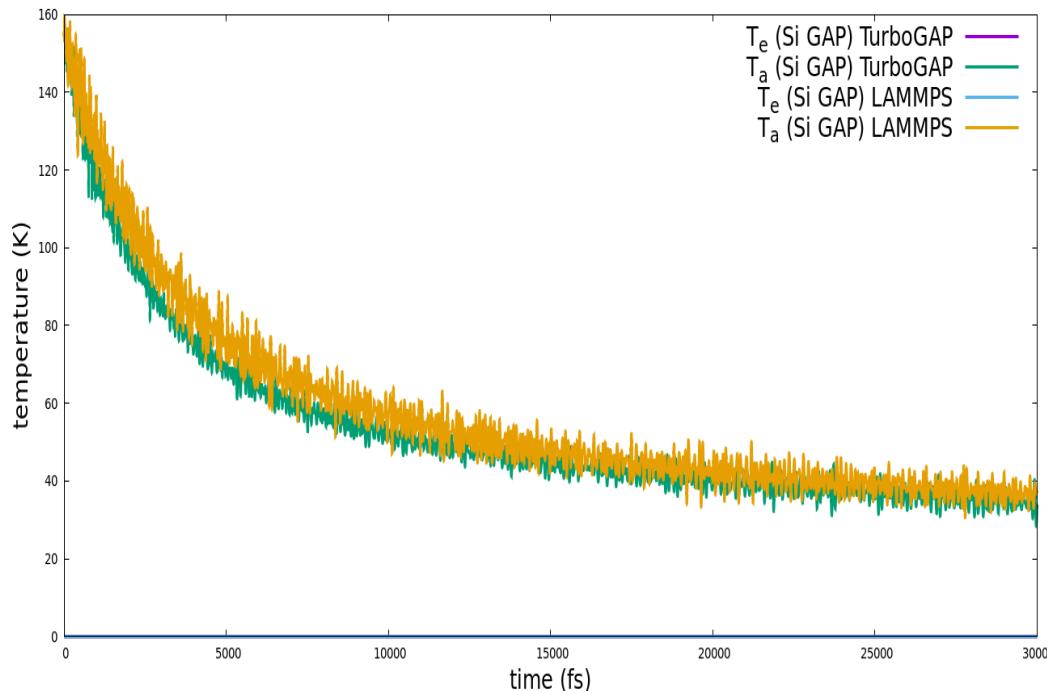
Same system and simulations using Si GAP with TurboGAP and using Tersoff, MEAM potentials with LAMMPS

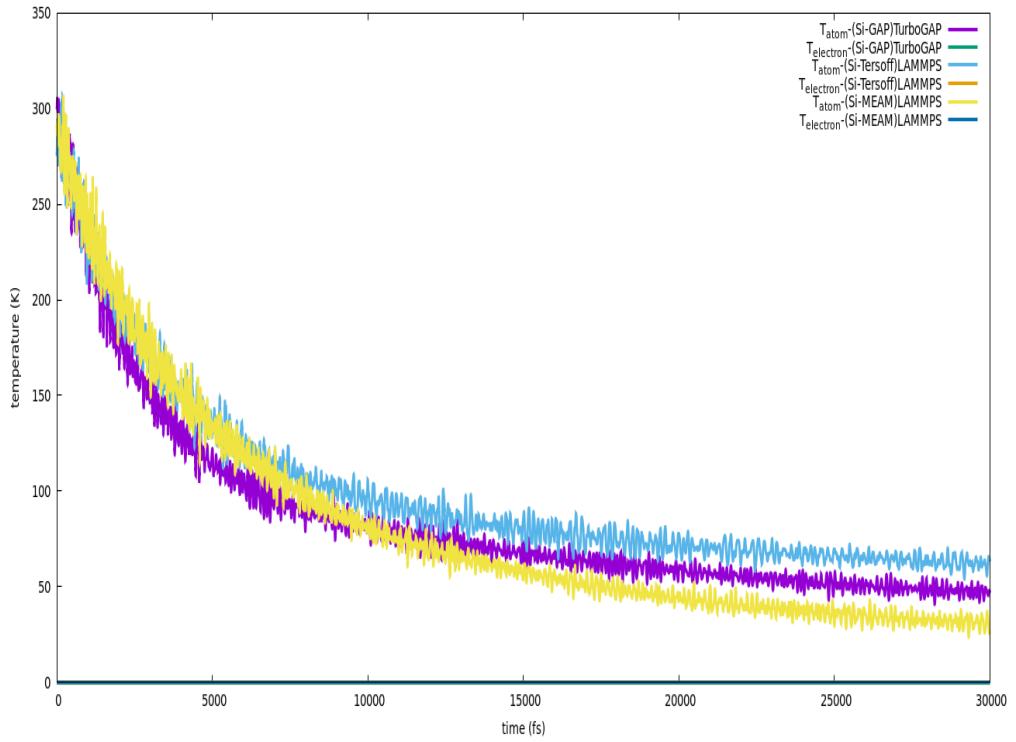


EPH model (with only friction and fixed $T_e = 0.0$ K) in TurboGAP and LAMMPS

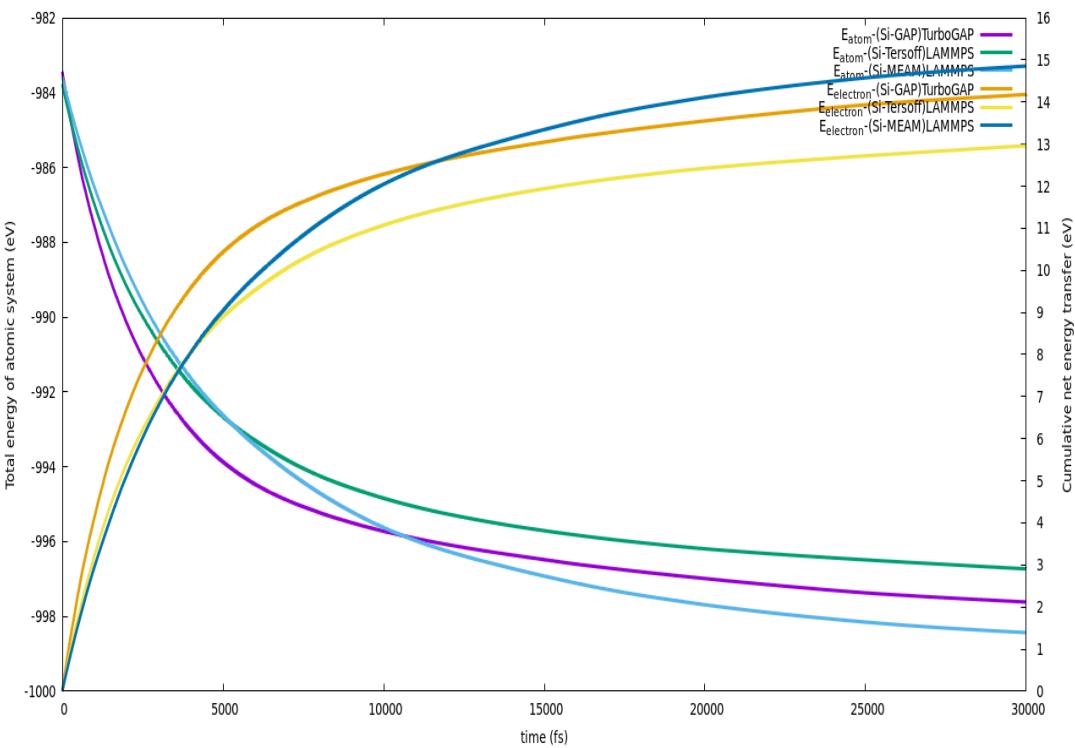
216 Si atoms in a box of dimensions 16.29 \AA on each side.
Using Si-GAP potential.

$T_a = 154 \text{ K}$, constant $T_e = 0.0 \text{ K}$
Coupling parameter – constant case of T. Jarrin et al. Phys. Rev. B 104, 195203





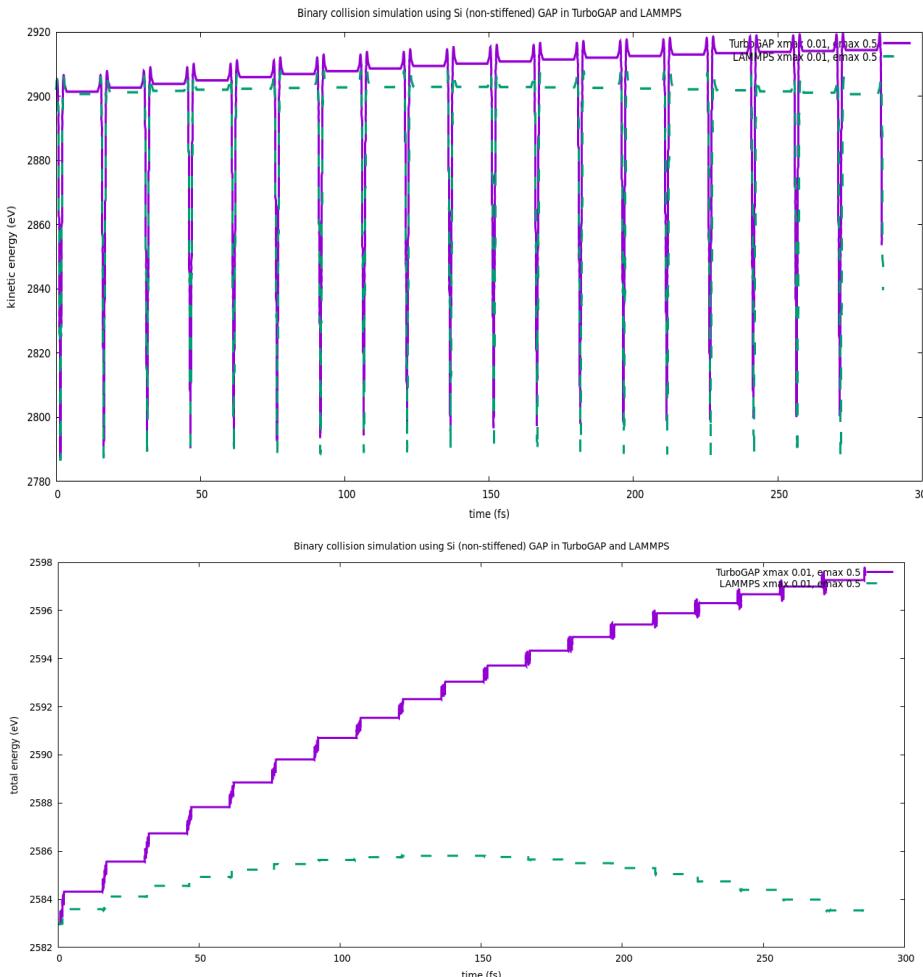
EPH model (with only friction and fixed $T_e = 0.0 \text{ K}$) using Si-GAP in TurboGAP and MEAM, Tersoff potentials in LAMMPS



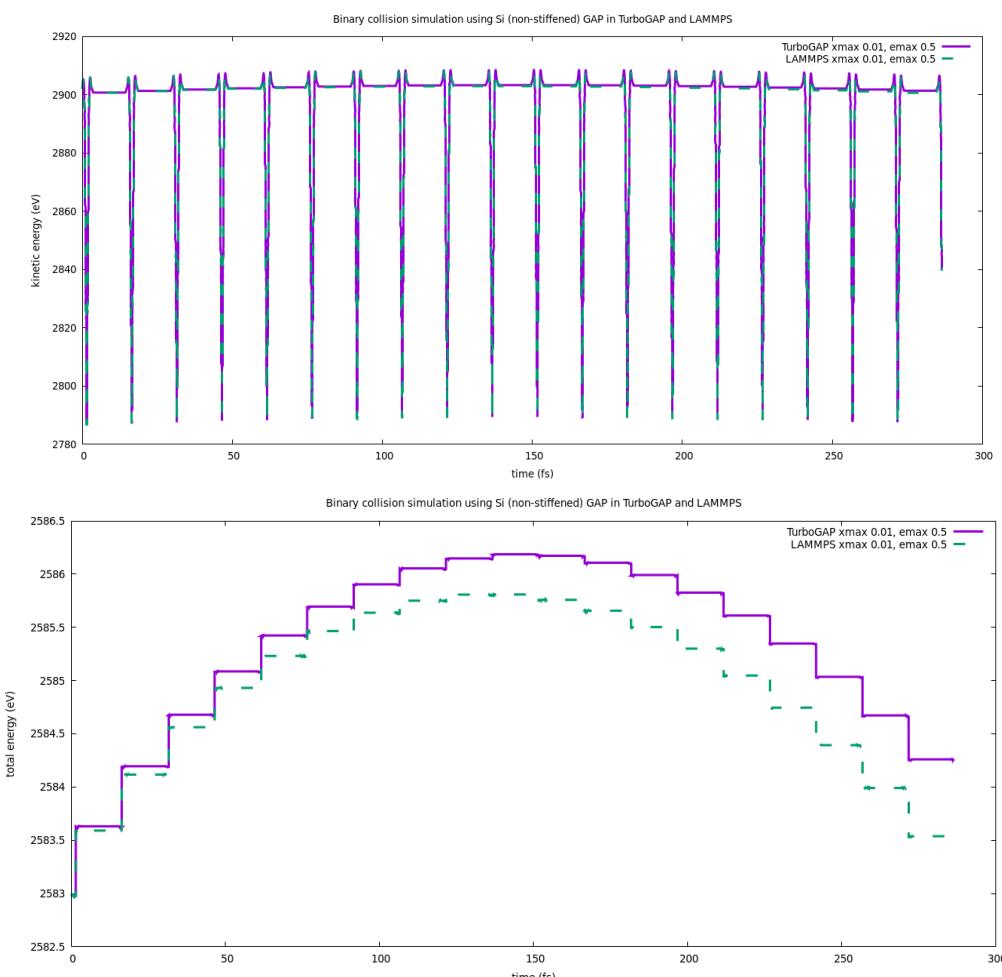
The original time-integration was suitable only for constant time-step simulations. The illustrations shown till here were using the original method before correction.

A small correction is done in this method to handle the variable time-step simulations, within the present scopes.

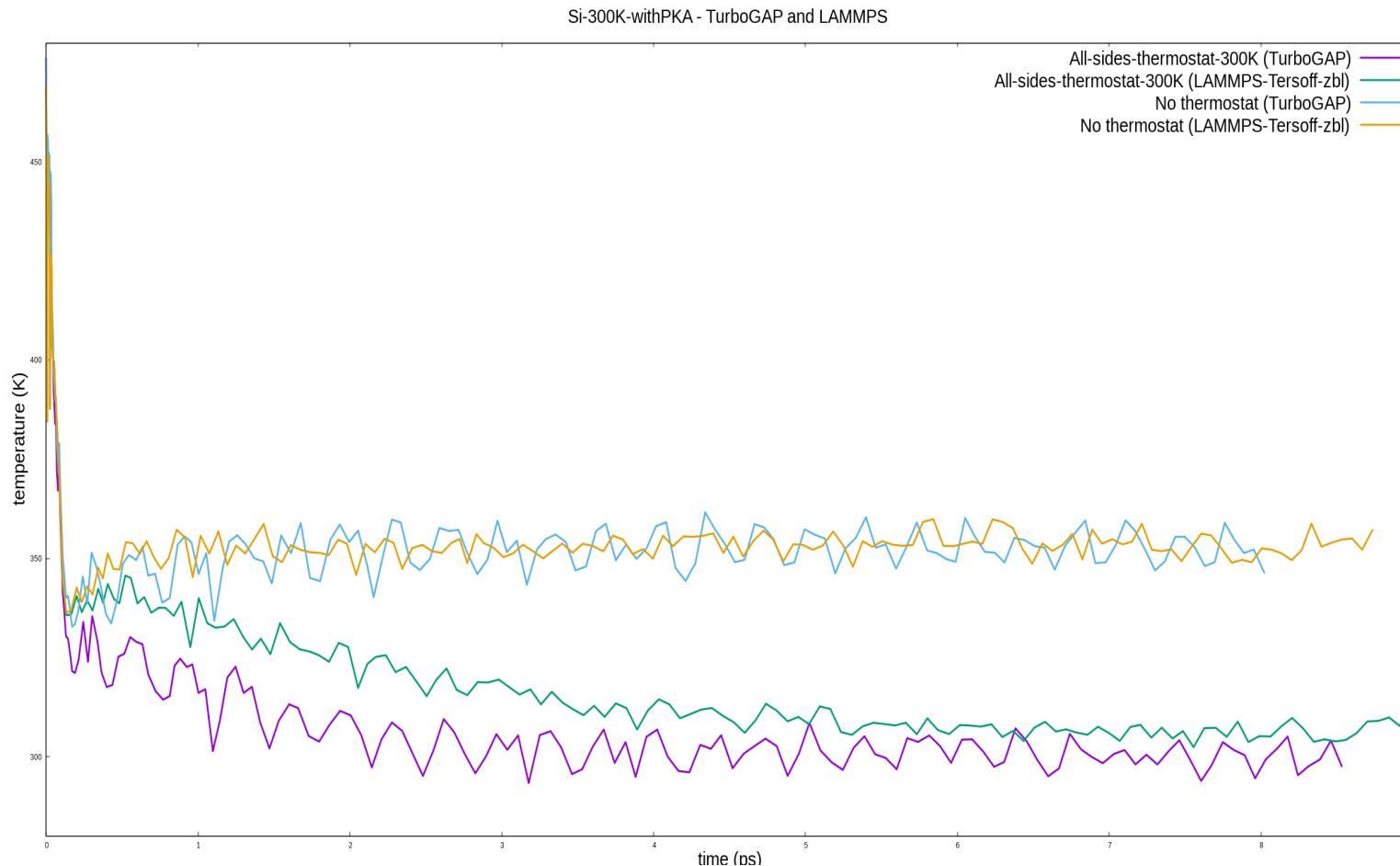
Previous



Present (corrected)



PKA simulation with/without 300 K thermostats on sides



Acknowledgements

Prof. Andrea Sand, Prof. Artur Tamm and Dr. Miguel A. Caro for good discussions.

Dr. Ali Hamedani for providing feedback on adaptive time from high energy cascade simulations.

The computational resources provided by the Aalto Science-IT project.